ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

INVESTIGATING CONJUGATE HEAT TRANSFER IN A SQUARE CYLINDER VIA LATTICE BOLTZMANN METHOD

M.Sc. THESIS

Aanif Hussain

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Program

JULY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

INVESTIGATING CONJUGATE HEAT TRANSFER IN A SQUARE CYLINDER VIA LATTICE BOLTZMANN METHOD

M.Sc. THESIS

Aanif Hussain
(511211101)

Department of Aeronautical and Astronautical Engineering

Aeronautical and Astronautical Engineering Program

Thesis Advisor: Prof. Dr. Bayram ÇELİK

JULY 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

LATTICE BOLTZMANN YAKLAŞIMIYLA KARE SİLİNDİRDE BİRLEŞİK ISI TRANSFERİNİN İNCELENMESİ

YÜKSEK LİSANS TEZİ

Aanif Hussain
(511211101)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bayram ÇELİK

TEMMUZ 2024

Aanif Hussain, an M.Sc. student of ITU Graduate School with Student ID 511211101 successfully defended the thesis entitled "INVESTIGATING CONJUGATE HEAT TRANSFER IN A SQUARE CYLINDER VIA LATTICE BOLTZMANN METHOD", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Bayram ÇELİK** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Mehmet ŞAHİN** Istanbul Technical University	.....
	Asst. Prof. Dr. Altuğ Melik BAŞOL Özyeğin University	.....
		.....

Date of Submission : **21 May 2024**
Date of Defense : **5 July 2024**

To my family,

FOREWORD

I am privileged to express my heartfelt gratitude to my esteemed advisor, Prof. Dr. Bayram Celik, whose guidance has been instrumental throughout my thesis journey. Prof. Dr. Celik's unwavering support and patience have enriched our weekly discussions, which spanned technical matters, career aspirations and moments of shared laughter. His mentorship extends beyond academia, shaping my personal growth and commitment to lifelong learning.

I extend my deepest gratitude to my family: my mother Dilafrose Kauser and my father Mohammad Hussain. They have wholeheartedly embraced and encouraged my academic pursuits. My parents have bestowed me with the greatest gifts of education, upbringing and moral guidance nurturing my intellectual curiosity and shaping the person I am today. My sisters, Beenish and Mehvish, with whom I share extraordinary connections based on mutual understanding and shared experiences. I am deeply thankful for the profound influence they have had in shaping my academic journey and personal growth. My grandmas, Mehr un Nisa and Hajira, who have been a source of unconditional love throughout my life.

I also extend my gratitude to my teachers at the Faculty of Aeronautics and Astronautics especially Prof. Dr. Mehmet Sahin and Prof. Dr. Ahmet Ziya Sahin. Their expertise and dedication to academia have been invaluable in expanding my understanding of aerospace engineering. I am thankful for the knowledge and insights they have imparted during their lectures, which have enriched my educational experience and inspired my research pursuits.

Finally, I express my deepest gratitude to Almighty for without His guidance and blessings, none of this would have been possible. I am humbled by His mercy in every aspect of my life. May His guidance continue to illuminate my path and grant me strength as I embark on new endeavors.

July 2024 Aanif Hussain

TABLE OF CONTENTS

ABBREVIATIONS

GPU	**:** Graphics Processing Unit
CPU	**:** Central Processing Unit
CUDA	**:** Compute Unified Device Architecture
DRAM	**:** Dynamic Random Access Memory
SM	**:** Streaming Multiprocessors
SIMT	**:** Single Instruction Multiple Thread
AMD	**:** Advanced Micro Devices
RTX	**:** Ray Tracing Texel eXtreme
CHT	**:** Conjugate Heat Transfer
LBM	**:** Lattice Boltzmann Method
PDE	**:** Partial Differential Equation
BTE	**:** Boltzmann Transport Equation
LGA	**:** Lattice Gas Automata
PDF	**:** Probability Distribution Function
BGK	**:** Bhatnagar-Gross-Krook
SRT	**:** Single Relaxation Time
MRT	**:** Multiple Relaxation Time
CFD	**:** Computational Fluid Dynamics
LBE	**:** Lattice Boltzmann Equation
NSE	**:** Navier Stokes Equation
NaN	**:** Not a Number
NEBB	**:** Non Equilibrium Bounce Back

SYMBOLS

F	**:** External force
f, g	**:** Distribution functions
c	**:** Lattice velocity
r	**:** Position vector
t	**:** Time
m	**:** Mass
K.E	**:** Kinetic Energy
L	**:** Length
B	**:** Height
P, p	**:** Pressure
K	**:** Boltzmann's constant
R	**:** Universal gas constant
T	**:** Temperature
Γ	**:** Time period
u, v	**:** Velocity components
e	**:** Internal energy
$\mathbf{f^{eq}}$	**:** Equilibrium distribution function
Ω	**:** Collision operator
ω	**:** Collsion frequency
τ	**:** Relaxation time
$\mathbf{w_i}$	**:** Weighting factor
Δt, Δx, Δy	**:** Step size in time, x and y direction respectively
ν	**:** Kinematic viscosity
ρ	**:** Fluid density
α	**:** Thermal diffusivity
$\mathbf{c_s}$	**:** Speed of sound
Re	**:** Reynolds number
M	**:** Mach number
D	**:** Square length
Pr	**:** Prandtl number
$\mathbf{k_r}$	**:** Conductivity ratio
Nu	**:** Nusselt number
$\mathbf{C_p}$	**:** Coefficient of pressure
h	**:** Convective heat coefficient
k	**:** Thermal conductivity coefficient
Bi	**:** Biot number

LIST OF TABLES

Page

LIST OF FIGURES

INVESTIGATING CONJUGATE HEAT TRANSFER IN A SQUARE CYLINDER VIA LATTICE BOLTZMANN METHOD

SUMMARY

Conjugate heat transfer (CHT) is an approach that involves combination of heat transfer in fluids and solids. Unlike general heat transfer phenomena, conjugate heat transfer involves simultaneous analysis of fluid flow and thermal behavior. In practical applications, the interaction between fluid flow and heat transfer is common, necessitating conjugate approach for conducting analysis. Conventional heat transfer analyses with constant thermal properties often do not consider the influence of fluid flow on the temperature distribution. Conjugate heat transfer studies become imperative when the effects of fluid motion on temperature variations cannot be neglected. Conjugate heat transfer finds applications across various fields that include aerospace engineering, automotive design, biomedical engineering and industrial processes. Examples include thermal management in high-speed aircraft engines, cooling systems for electronic devices and optimizing heat dissipation in manufacturing processes etc. Beyond traditional engineering disciplines, conjugate heat transfer is also relevant in biomedical sciences (e.g., thermal modeling for drug delivery systems), geophysical studies (e.g., modeling heat transfer in earth sciences), and chemical engineering (e.g., optimizing reactor designs).

Presently, numerical simulations using the Lattice Boltzmann Method (LBM) offer a powerful and effective alternative to classical CFD methods for studying conjugate heat transfer. This is because LBM is inherently parallelly programmable and does not require a conjugate boundary condition for solving CHT problems. Advancements in computational techniques particularly utilizing the parallel processing capabilities of Graphics Processing Units (GPUs) have revolutionized the numerical solvers that are parallelly programmable. High-performance computing centers enable researchers to tackle complex problems efficiently by harnessing the computational power of GPUs. This is particularly advantageous in problems such as conjugate heat transfer that involves intricate thermal-fluid dynamics. Since, LBM is inherently adaptable to run parallel computations, it is ideal for simulating CHT problems. Therefore, LBM with its GPU-based parallel computing capabilities not only enables research and studies in this field with unprecedented accuracy and computational efficiency but also eliminates the need for conjugate boundary conditions, thereby reducing computational costs significantly.

In this thesis, an in-house developed Lattice Boltzmann solver is employed to simulate conjugate heat transfer. Non-isothermal viscous flow around a 2-D bluff body i.e. a thermally conductive square cylinder confined in a duct is the primary focus of analysis. The LBM solver is versatile and can be easily adapted to various other geometric configurations and can also be extended to solve multi-phase flows by

incorporating relevant phase equations in the numerical solver. The computational setup of the LBM-CHT solver includes a flow solver and a heat solver integrated together in a single framework of a CUDA file (.cu). The solver is implemented using CUDA C++ programming model that is built to be executed on NVIDIA GPUs. Chapter 1 introduces the basic GPU programming principles that were used to develop the solver. Chapter 2 covers the theoretical aspect of Lattice Boltzmann method giving insights into the early developments, schemes used for relaxation models and different lattice arrangements in 1-D, 2-D and 3-D problems. Chapter 3 discusses the developments in the field of conjugate heat transfer and its numerical implementation using LBM. Numerical stability and boundary conditions are also discussed. Finally, we start our numerical study in chapter 5. We begin with the validation of our flow and heat solvers. The geometrical definition of our problem setup, boundary conditions, flow and thermophysical properties used for CHT investigation are discussed. The whole workflow of the numerical solver is provided in a flowchart followed by mesh dependency test. Post-processed results given as contours of flow field and temperature distribution are illustrated through a series of figures that are generated with .vtk extension and visualised in PARAVIEW. Each figure is unique to user-defined input for Prandtl number, solid to fluid conductivity ratio and Reynolds number. Chapter 5 is dedicated to discussion and analysing the key findings. The fluid flow in this study pertains to unsteady time periodic regime that commences at a critical threshold of $Re \geq 50$ for the square cylinder in the channel with blockage ratio $H/D = 4$. The periodic nature of flow affects the temperature distribution and heat transfer in the system. To summarise and parameterise all the findings of this study, time-averaged Biot number of the cylinder is calculated to judge its performance. Finally, the concluding remarks about this study are highlighted in chapter 6.

This study contributes to the understanding of conjugate heat transfer phenomena through computational modeling and numerical simulations. The findings provide valuable insights into the effect of fluid flow on heat transfer and temperature distribution. This can help in paving the way for optimized design and performance of thermal systems.

LATTICE BOLTZMANN YAKLAŞIMIYLA KARE SİLİNDİRDE BİRLEŞİK ISI TRANSFERİNİN İNCELENMESİ

ÖZET

Konjuge ısı transferi (CHT), hem katı hem de sıvıdan oluşan bir bölgede ısı transferini analiz etmek için kullanılan bir yaklaşımdır. Bu yöntemde, ısı transferi sıvıda konveksiyon ve katıda iletim ile yönetilir. Isı değiştiriciler, yanma odaları, mikro-yakıt hücreleri ve çip üzerinde laboratuvar cihazları gibi pratik sistemler, termal bileşenlerin tasarımı, verimliliği ve optimizasyonu için konjuge yaklaşımı kullanır. Yastık plakalı, kanat ve boru, düz borulu ısı değiştiriciler, aero-motor kanatlarının soğutulması, yanma odası kaplamaları ve elektrikli bileşenler gibi uygulamalar CHT kullanımını içerir. Sıvı-katı ara yüzeyinde ısı transferine yönelik deneysel çalışmalar mevcut olsa da, ölçekleme zorlukları, hassas gerçek zamanlı ölçüm zorlukları ve yüksek maliyetler nedeniyle genellikle daha küçük geometrilerle sınırlıdır. Hesaplamalı Akışkanlar Dinamiği (CFD) ve diğer sayısal simülasyon yöntemleri, CHT çalışmalarını yürütmek için güçlü bir alternatif sunar. Sonlu fark ve sonlu hacim yöntemleri, CHT problemlerini doğru bir şekilde çözer ve pratik uygulamalarda yaygın olarak kullanılır. Ancak, bu yöntemler büyük ölçekli geometriler ve düzensiz geometriler, Newton olmayan akışkan özellikleri gibi karmaşık sıvı-katı ara yüzleri ile başa çıkarken kaynak ve zaman açısından maliyetli olabilir. Bir CHT probleminde, sıvı-katı ara yüzeyinde ısı transferini doğru bir şekilde modellemek esastır. Sabit duvar sıcaklığı veya sabit ısı akısı gibi geleneksel sınır koşullarının kullanılması yetersizdir çünkü bu, katıdaki ısı iletim rolünü göz ardı eder.

Sıvı ve katı arasındaki ara yüzeye konjuge sınır denir ve ısı transfer mekanizmasını modellemek için iki yönlü bir bağlama koşulu gerektirir. Bu, yerel ısı akısını sağlamak ve sürekli sıcaklık dağılımını sağlamak için ara yüzeyde enerji korunumu ve termodinamik yasalarını kullanarak elde edilebilir. Sıvı-katı ara yüzeyindeki sınır koşulu, Dirichlet ve Neumann sınır koşullarının bir kombinasyonunu içeren geleneksel olmayan bir yaklaşımla formüle edilmiştir. Bir CHT problemini verimli bir şekilde çözmek için, istenen sonuçları makul bir hesaplama süresi içinde elde etmek, pratik uygulama için gereklidir. Sıvı ve katı denklemlerini monolitik bir yaklaşımla çözmek, zaman alıcı bir çözümleyici uygulamasına sahiptir ve hesaplama açısından maliyetlidir, ancak güçlü bir bağlama ve küresel bir çözüm sunar. Öte yandan, ayrılmış yaklaşım, sıvı ve katı için çözümleyicileri ayırarak çözüm sürecini basitleştirir, ancak yinelemeli bağlama ve daha uzun hesaplama süresi pahasına gelir. Bu CFD yaklaşımlarının her ikisi de konjuge ısı transferi problemlerini çözmek için popülerdir, ancak genel olarak, uygulama, hesaplama kaynakları ve zaman açısından artırılmış yetenekler gerektirirler. Bu nedenle, alternatif bir sayısal teknik olan Lattice Boltzmann yöntemi, son on yıllarda termo-akışkan problemlerini çözmek için güçlü bir hesaplama aracı haline gelmiştir.

Lattice Boltzmann yöntemi (LBM), makroskopik durum değişkenlerini değerlendirmek için sahte-parçacık dağılım fonksiyonunu kullanan parçacık tabanlı bir sayısal çözümleyicidir. İlk olarak 1973 yılında lattice gaz otomataları (LGA) olarak tanıtılan LBM, akışkan akışlarını simüle etmek için bir hesaplama tekniği olarak geliştirilmiştir. Araştırmacılar, çok fazlı akışlar, manyeto-hidrodinamik, gözenekli medya akışı, türbülanslı akışlar ve konjuge ısı transferi gibi çeşitli fiziksel fenomenlerde LBM'nin uygulamalarını göstermiştir. Birçok çalışma, farklı sınır koşullarını ele almak ve LBM simülasyonlarının doğruluğunu ve verimliliğini artırmak için daha yüksek dereceli sınır düzenleri geliştirmeye odaklanmıştır. Konjuge ısı transferi bağlamında, LBM, ısı transfer mekanizmasını yönetmek için sıvı-katı ara yüzeyinde açık bir sınır koşulu gerektirmez, bu da uygulanmasını kolaylaştırır. Ayrıca, çarpışma ve akış süreçlerinin yapısı nedeniyle LBM, paralel programlamaya doğal olarak uygundur ve GPU'lar üzerinde olağanüstü hesaplama hızı sağlar.

LBM'de parçacık dağılım fonksiyonu, uzay ve zamanda bir parçacık koleksiyon birimini temsil eder. Dağılım fonksiyonları, ilgi alanındaki belirli konumlarda, lattice siteleri olarak bilinen yerlerde yerleştirilir. Genel olarak, Lattice Boltzmann yöntemi, birim bir kartesyen ızgara kullanılarak uygulanır, ancak yapılandırılmamış ızgaraların son uygulamaları umut verici sonuçlar göstermiştir. Konjuge ısı transferi problemleri için, dağılım fonksiyonu, doğası gereği sıvı-katı ara yüzeyinde ısı akısı ve sıcaklık dağılımının sürekliliğini sağlar. Mezomik bir miktar olarak, sıvı ve katı parçacıklar arasındaki etkileşimleri hesaba katar, sıvı-katı ara yüzeyini izleme gereksinimini ortadan kaldırır. Bu ayırt edici özellik, LBM'yi CHT problemlerini çözmek için ideal hale getirir. İlk olarak LBM'nin, sıvı-katı ara yüzeyinde açık bir işlem gerektirmeden ve düşük Reynolds sayısı akış rejiminde düz ara yüzeylerde iyi uygulanarak, CHT problemlerini CFD çözümleyicilerden daha iyi çözebileceği gösterilmiştir. LBM'deki ilerlemeler, daha fazla CHT problemine uygulamalarını genişletmiştir. Örneğin, araştırmacılar konjuge ısı transferi problemlerini keyfi ara yüzeyler için modellemiş ve eğri ara yüzeylerle başa çıkmak için bir gömülü sınır şeması önermişlerdir. LBM alanındaki devam eden araştırmalar göz önüne alındığında, çeşitli akışkan akışı ve ısı transferi problemlerinde uygulamalarını keşfetmeye devam eden birçok çalışma bulunmaktadır.

Bu çalışmanın amacı, düşük Reynolds sayılarında meydana gelen zaman periyodik akışlar için bir kanalda sıkışmış katı kare bir silindirin ısı transferi özelliklerini araştırmaktır. GPU'da çalışabilen doğrulanmış bir içsel lattice Boltzmann çözümleyicisi, CHT simülasyonlarını gerçekleştirmek için kullanılmaktadır. Çözümleyici, akışkanlarda sıkıştırılamaz viskoz akış ve konveksiyon denklemlerini ve katı bölgede iletim denklemini sayısal simülasyon için uygular. Reynolds sayıları, $Re \geq 50$ olduğunda, kanaldaki ısı ve kütle transferleri akış fiziğinden etkilenir. Katıdan sıvıya termofiziksel özellikleri (k_r), Prandtl sayısı (Pr) ve Reynolds sayısını (Re) sistematik olarak değiştirerek parametrik bir çalışma yaparız. Bu şekilde, genel ısı transfer mekanizmasını, zaman ortalamalı sıcaklık değişimlerini ve sistemin genel performansını belirleyebiliriz. Çalışma, $Re = 50, 100, 150$, ve 200; $k_r = 0.71, 1, 5$, ve 10; ve $Pr = 0.71, 1, 5$, ve 10 için gerçekleştirilmiştir. Silindir-kanal sistemindeki ısı transferi ve sıcaklık dağılımını incelemek için farklı çalışma koşulları altında kararsız kütle ve ısı transferleri araştırılmıştır.

1. Bölüm, çözümleyiciyi geliştirmek için kullanılan temel GPU programlama ilkelerini tanıtır. 2. Bölüm, Lattice Boltzmann yönteminin teorik yönlerini ele alır ve erken gelişmeler, rahatlama modelleri için kullanılan şemalar ve 1-D, 2-D ve 3-D problemlerindeki farklı ızgara düzenlemeleri hakkında bilgi verir. 3. Bölüm, konjuge ısı transferi alanındaki gelişmeleri ve LBM kullanarak sayısal uygulamasını tartışır. Sayısal stabilite ve sınır koşulları da tartışılmıştır. Son olarak, sayısal çalışmamıza 5. bölümde başlıyoruz. Akış ve ısı çözücülerimizin doğrulanmasıyla başlıyoruz. Problem kurulumumuzun geometrik tanımı, sınır koşulları, CHT araştırması için kullanılan akış ve termofiziksel özellikler tartışılmaktadır. Sayısal çözümleyicinin tüm iş akışı, bir akış şemasında sağlanmış ve ardından ağ bağımlılığı testi yapılmıştır. Akış alanı ve sıcaklık dağılımının konturları olarak verilen işlenmiş sonuçlar, .vtk uzantılı olarak üretilmiş ve PARAVIEW'de görselleştirilmiş bir dizi şekil aracılığıyla gösterilmiştir. Her şekil, kullanıcı tarafından tanımlanan Prandtl sayısı, katıdan sıvıya iletkenlik oranı ve Reynolds sayısı girişi için benzersizdir. 5. Bölüm, tartışma ve ana bulguların analizine adanmıştır. Bu çalışmadaki akışkan akışı, $Re \geq 50$ kritik eşiğinde başlayan zaman periyodik rejime aittir ve blokaj oranı $H/D = 4$ olan kanaldaki kare silindir için geçerlidir. Akışın periyodik doğası, sistemdeki sıcaklık dağılımını ve ısı transferini etkiler. Bu çalışmanın tüm bulgularını özetlemek ve parametrelemek için silindirin zaman ortalamalı Biot numarası hesaplanmış ve performansı değerlendirilmiştir. Son olarak, bu çalışma hakkındaki sonuçlar 6. bölümde vurgulanmıştır.

Bu çalışma, hesaplamalı modelleme ve sayısal simülasyonlar yoluyla konjuge ısı transferi fenomeninin anlaşılmasına katkıda bulunmaktadır. Bulgular, akışkan akışının ısı transferi ve sıcaklık dağılımı üzerindeki etkisine dair değerli bilgiler sunmaktadır. Bu, termal sistemlerin optimize edilmiş tasarımı ve performansı için yol açabilir.

1. OVERVIEW OF GPU PROGRAMMING

In this chapter, we introduce the fundamental concepts of GPU architecture, delve into the structure of CUDA, explore the implementation and utilization of kernel functions, and examine the principles of thread algebra essential for efficient parallel computing.

1.1 Introduction

Graphics Processor Unit, GPU has evolved into a robust source for accelerating computations in addition to its original utilization for producing high-definition 3D graphics. In recent years, GPUs have been associated with CPUs to perform heterogeneous computations [4]. A multi-core and multi-threaded processor with a large memory bandwidth, GPU has emerged as a horsepower of data processing specializing in highly parallel compute-intensive tasks [3]. From handy desktop and laptop systems to clusters and super-computers, GPUs are employed in broad variety of disciplines that involve technical computing. Figure 1.1 shows the evolution of the number of floating-point operations per seconds carried out by the successive versions of GPUs and CPUs. Due to more transistors dedicated to processing data in GPUs than flow control and data caching, they outperform CPUs in highly parallel computational tasks [4].

CUDA, an abbreviation for Compute Unified Device Architecture, is amongst the powerful programming interfaces used for launching parallel compute engines (kernels) in NVIDIA GPUs to solve computational problems. CUDA is accessed through CUDA libraries and compiler directives that are offered in widely-used programming languages including Python, C++, C and FORTRAN as illustrated in Figure 1.2. CUDA programming model executes parallel computations by adding a set of extension codes to the numerical program. This environment is heterogeneous that is compiled on CPUs and complemented by GPUs. Beginning with CUDA 6 version and higher, a Unified Memory was introduced by NVIDIA to help the user access both

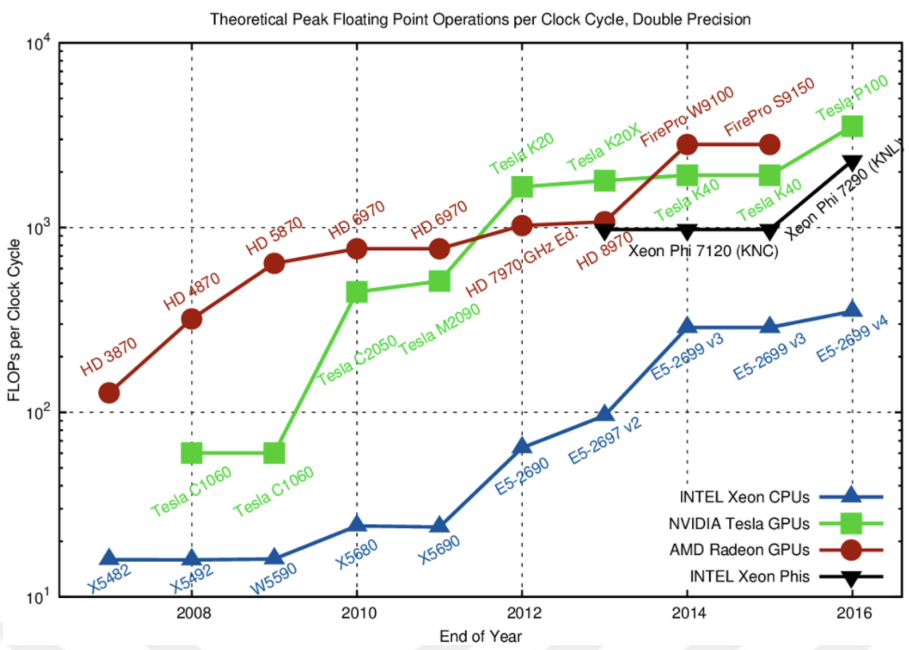


Figure 1.1 : Time history of floating point operations per second for GPU and CPU [3].

CPU and GPU memory by employing a single pointer. The data seamlessly transitions between the Host (CPU) and the Device (GPU) automatically. [4]. For this thesis study, CUDA C++ enabled GPU programming was utilised for carrying out computations to reap the benefits of parallelization [11].

1.2 CUDA Structure and Device Architecture

A CUDA program includes sequential code that executes on the CPU and is enhanced by parallel code which runs on GPU. The code which is executed on host (CPU) is programmed in standard C++ and the code which is executed on device (GPU)

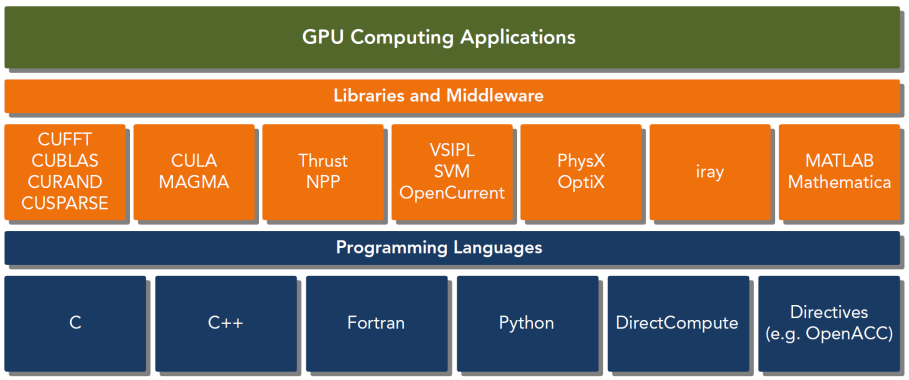


Figure 1.2 : Accessibility to CUDA platform [4].

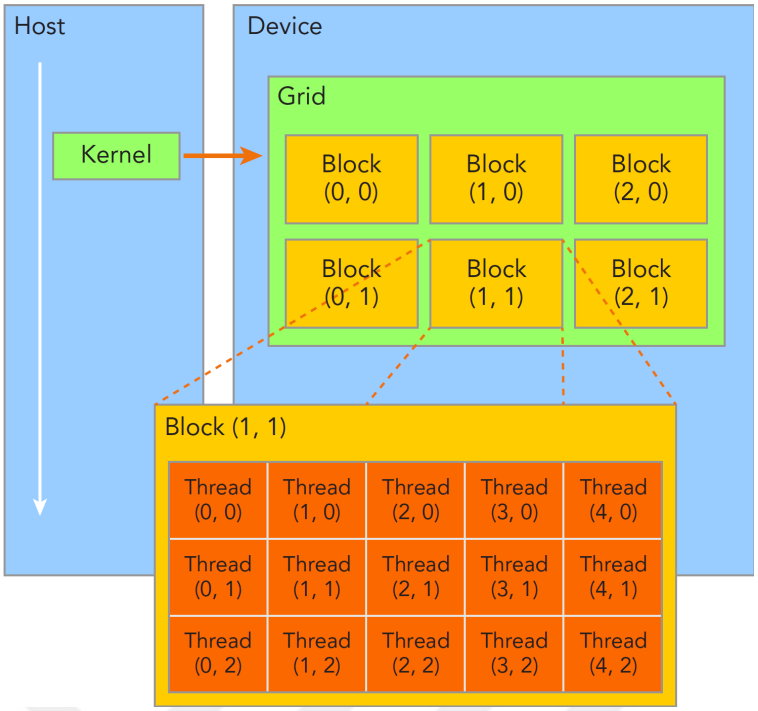


Figure 1.3 : CUDA layer model [3].

is programmed in CUDA C++ [3]. The host code operates independently whereas the device code, once executed, switches back the control to the host. A kernel is a primary component in programming with CUDA. It is a block of code that runs on the GPU (device). It is declared using the `__global__` declaration specifier. There are three distinct layers of control in a CUDA program to support the parallelization of an algorithm, as shown in Figure 1.3. The most basic and finest component is called a thread. The second layer is called a block. It is comprised of threads bundled together and has a maximum limit of 1024 threads per block. Finally, a grid of blocks when executed together is referred to as a kernel [12]. The host and device explicitly handle their own memory known as host memory and device memory (DRAM) respectively. The memory is allocated to GPU variables using the CUDA command `cudaMalloc(variable_name)`. The data from host to device is copied using `cudaMemcpyHostToDevice([options])` and from device to host using `cudaMemcpyDeviceToHost([options])`. The re-allocation command `cudaFree(variable_name)` frees the variable memory from the device [5]. The general flow of commands that take place in a CUDA program is shown in Figure 1.4.

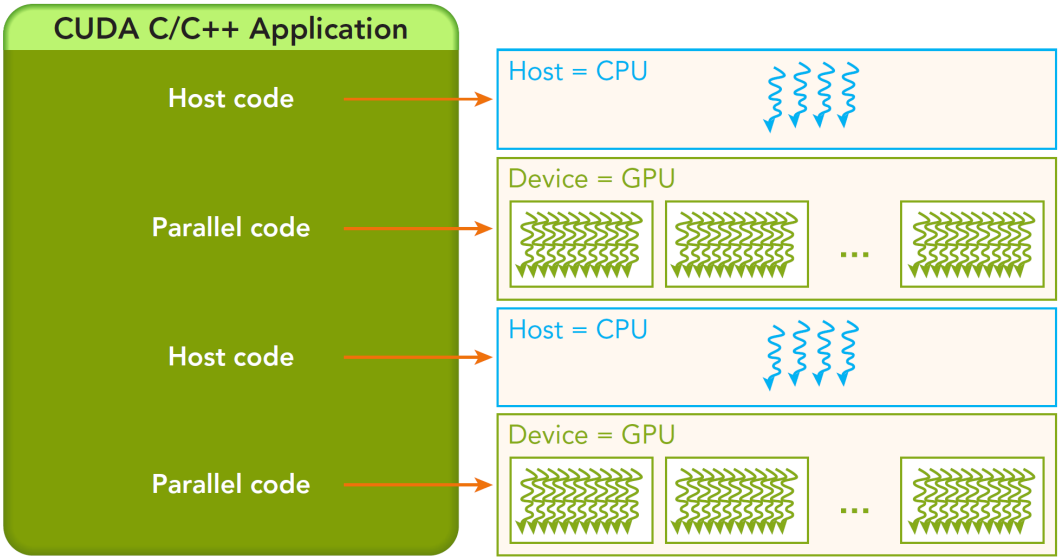


Figure 1.4 : CUDA program structure [4].

The architecture of Graphics Processing Unit is centered on a cluster of Streaming Multiprocessors (SM). Each SM supports parallel execution of threads. Thus, a single GPU has multiple SMs that can execute thousands of threads concurrently. Launching a kernel triggers a number of thread blocks to be divided amongst various SMs or it may occur on a single SM depending on the availability of resources. In CUDA, a wrap is a group of 32 threads that perform the same action at the same instant. Each SM has an assigned partition of 32-thread wraps. This framework employed by CUDA is called Single Instruction Multiple Thread (SIMT) [4]. The key components of an SM are given below and illustrated in Figure 1.5.

- CUDA cores
- L1 cache or shared memory
- Register file
- Load or store units
- Special function units
- Warp scheduler

A block of threads in CUDA operates on a single SM. Until the execution of the kernel, the thread block will remain scheduled on that particular SM. However, multiple thread

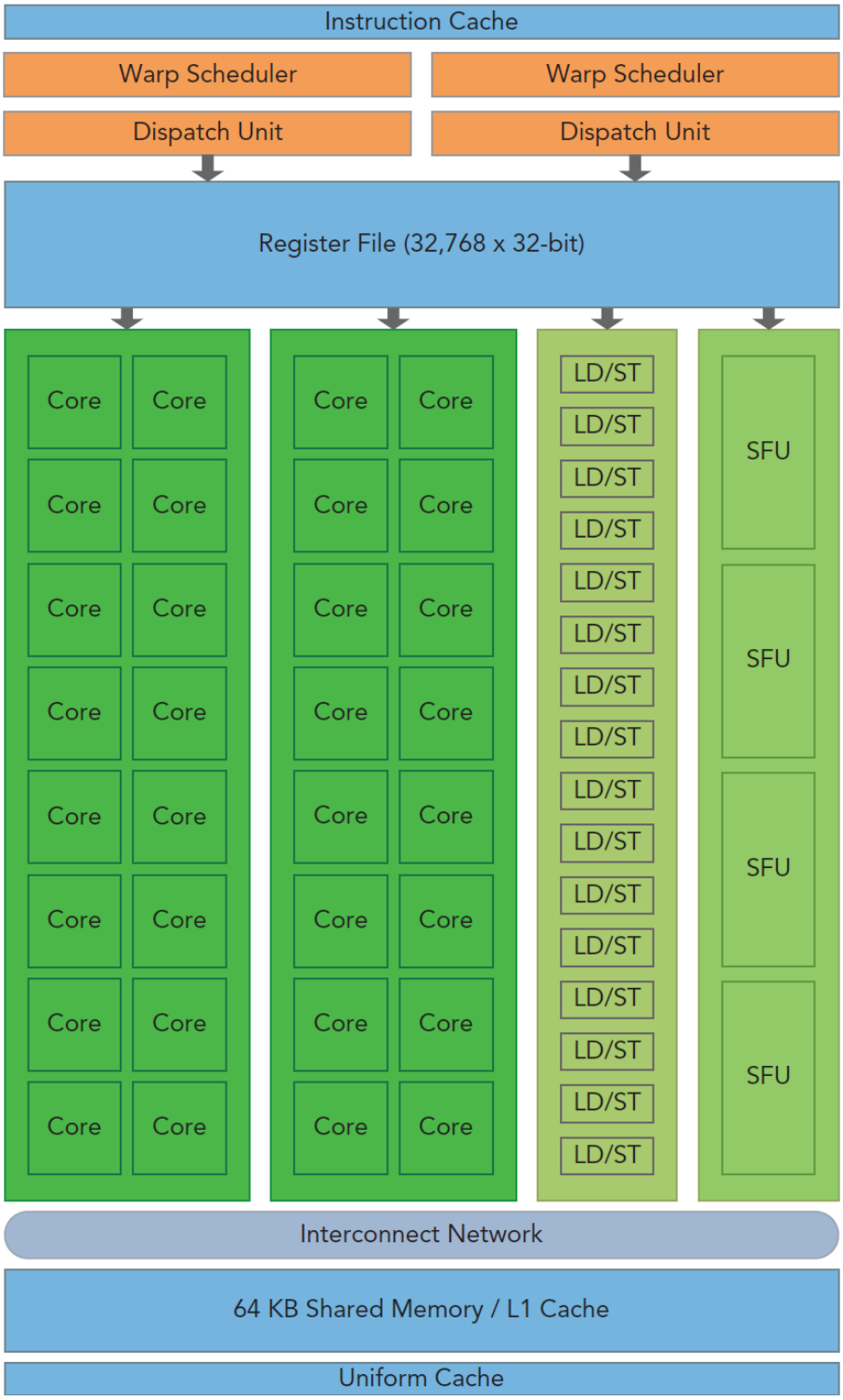


Figure 1.5 : GPU memory architecture [4].

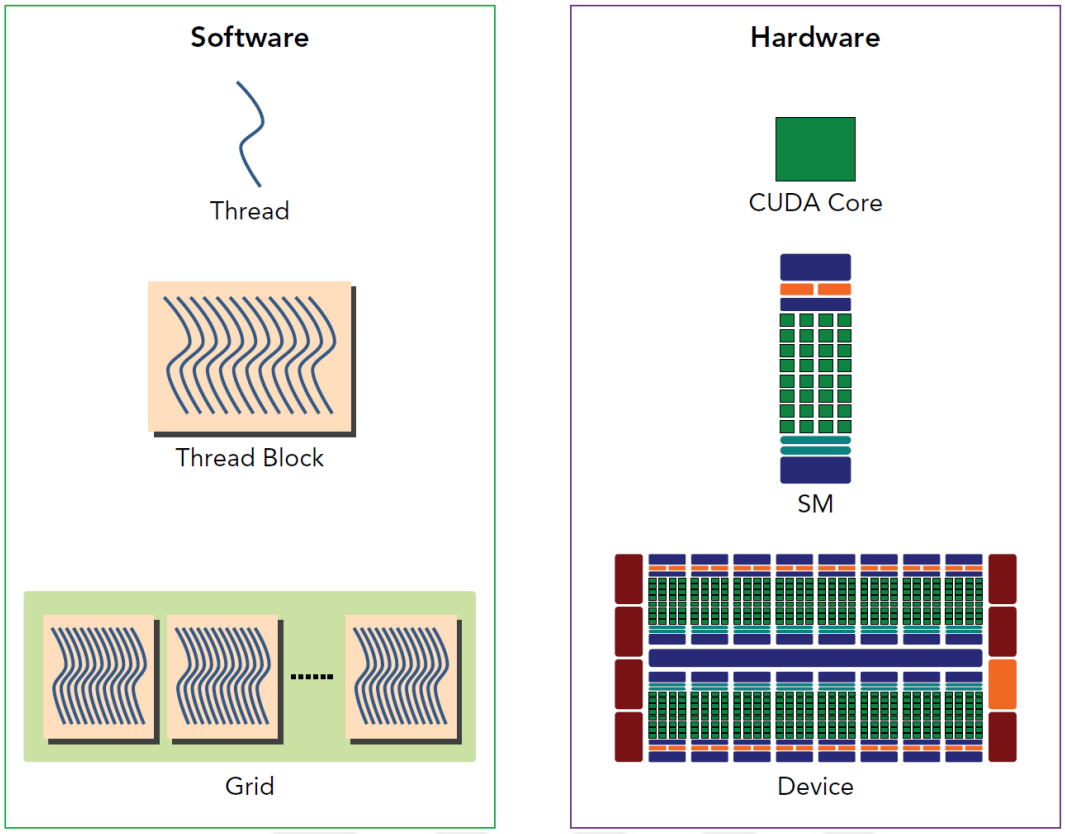


Figure 1.6 : GPU (device) software and hardware structure [4].

blocks can also be associated with an SM. The software and hardware illustration of the CUDA key components is shown in Figure 1.6. Only the shared memory and registers make it possible for the threads in a block to intercommunicate. The former is divided among multiple thread blocks in a SM whereas the latter is partitioned among the threads [5]. It is important to note that all the threads are not executed together in a block. Threads work at different paces inside a thread block.

1.3 Kernel Functions

As previously introduced, a kernel function is a piece of code written in CUDA C/C++ that runs on the GPU and is executed by many parallel threads.

1.3.1 Launching a kernel

The code that runs on device (GPU) is implemented via kernel function. It defines the procedure of computation that will take place for a single thread and its accessible data.

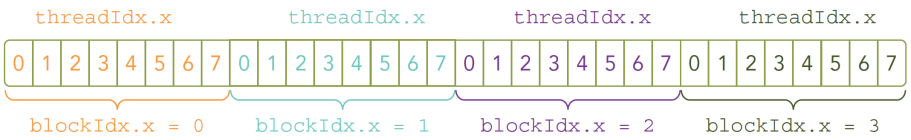


Figure 1.7 : Layout of blocks and threads [5]

When a kernel is invoked, the CUDA threads parallelly perform the computations. In CUDA, a kernel call is made from the host as shown:

```
kernel_name <<<grid, block>>>(argument list);
```

The identification numbers for threads in a block and for blocks in a grid is pre-assigned and can be determined by thread algebra which is discussed in section 1.4. The identification of blocks and threads is similar to numbering elements in a large matrix. This helps in configuring:

- The total thread count required for a kernel.
- The dimensions of the blocks to be used inside a kernel.

It is important to note that only the threads within a block can communicate with each other and threads from different blocks are isolated [4]. Therefore, it is important to organize the threads in a suitable manner. As an example, if 32 data elements are needed for carrying out a computation, 4 blocks of 8 threads each can be deployed for a kernel as shown:

```
kernel_name <<<4, 8>>>(argument list);
```

The layout of threads for this kernel call is shown in Figure 1.7. Since the data is stored linearly in the global device memory, built-in variables `blockIdx.x` and `threadIdx.x` are used to:

- Give the thread a unique global identity.
- Establish a mapping between threads and data elements.

Based on the requirement of data to be processed, a kernel can be launched in suitable ways such as 32 elements in one block or 32 blocks with one element each as shown:

```
kernel_name <<<1, 32>>>(argument list);
            OR
kernel_name <<<32, 1>>>(argument list);
```

Once a kernel is invoked, the control switches back immediately to the host irrespective of whether the kernel was completely processed on all threads [5]. To force the host to wait for kernel completion, the following function should be used after every kernel call:

```
cudaDeviceSynchronize(void);
```

1.3.2 Writing a kernel

To declare a kernel in the host code, the `__global__` declaration specifier followed by kernel name is used. The kernel function must always have a void return type.

```
__global__ void kernel_name(arguments);
```

A CUDA kernel function is written in a way that the loop for executing the iterations is not needed [3]. For example, adding the elements of two vectors each of size 32 parallelly on GPU, the kernel function is declared as:

```
__global__ void sumArrays(float *B, float *A, float *C) {

int j = threadIdx.x;

C[j] = A[j] + B[j];

}
```

While the host C++ program for the same example would use a loop to iterate for each element of the vectors as shown:

```
void sumArrays(float *B, float *A, float *C, const int M) {
for (int j = 0; j < M; i++)
C[j] = A[j] + B[j];
}
```

To invoke the kernel from the host and perform the computation on the device without returning to the host immediately, one should launch the kernel as:

```
sumArrays<<<1,35>>>(float *B, float *A, float *C);
cudaDeviceSynchronize(void);
```

1.3.3 Function types in CUDA

The summary of different function types used in CUDA C++ is provided as:

- __global__
  - It is implemented on the device (GPU).
  - It can be invoked from both host and devices with compute capability of 3.
  - Only void return type.
- __device__
  - It is implemented on the device (GPU).
  - It can be invoked from the device only.
- __host__
  - It is implemented on the host (CPU).
  - It can be invoked from host only.

Despite the usefulness, CUDA kernels have certain restrictions that should be kept in mind while using them:

- They are likely to exhibit a non-synchronous behavior.
- They have access only to the memory available on device.
- They do not support function pointers.
- They do not endorse a variable number of arguments.
- They do not back static variables.

1.4 Thread Algebra

A crucial part of CUDA programming is knowing how to organize the threads. Each thread is unique and can be distinguished from others due to two identification coordinates which are:

- Index of block in a grid.
- Index of thread in a block.

The indexes are in-built and accessed by variables `blockIdx` and `threadIdx` in a CUDA program [3]–[5]. Depending on the dimension of the block, the block indexes in y, x and z are defined by:

```
blockIdx.y
blockIdx.x
blockIdx.z
```

Similarly, indices of thread in y, x and z directions are given as :

```
threadIdx.y
threadIdx.x
threadIdx.z
```

The grid and block dimensions are organized in three dimensions and are accessed by variables `gridDim` and `blockDim`. For most of the matrix operations including our 2-D conjugate heat transfer problem, the general approach is to employ a 2D grid with 2D blocks as shown in Figure 1.8. For these types of operations, two primary indices are needed:

- Block and local thread index
- Global thread index

A thread can be identified systematically in two steps. Firstly, it belongs inside a certain block that has a specific block identification number which can be accessed by `blockIdx.x` and `blockIdx.y` along x and y axis respectively. Inside the block, it

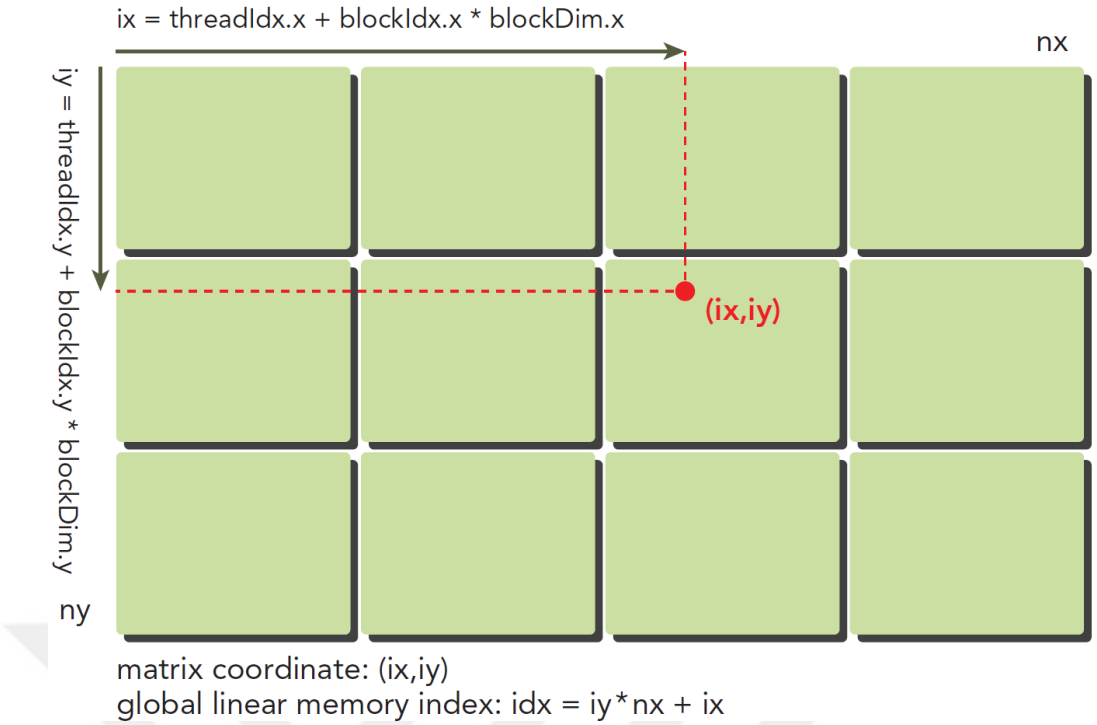


Figure 1.8 : Local and global thread index [4].

has a specific thread identification number which is accessed by `threadIdx.x` and `threadIdx.y` in x and y direction. The local indices of a thread inside block of threads are computed as:

```
ix = blockIdx.x * blockDim.x+threadIdx.x

iy = blockIdx.y * blockDim.y +threadIdx.y
```

After calculating local indices, the global identity number (index) of threads in a grid can be calculated as:

```
idx = iy * nx + ix
```

NOTE: The simulations for this thesis study were conducted using a system equipped with an AMD Ryzen 7 CPU(5800H) @3.201 GHz featuring 8 cores and 16 logical processors, along with an NVIDIA GeForce RTX 3050 GPU and CUDA V12.0 Toolkit.

2. LATTICE BOLTZMANN METHOD

In this chapter, we introduce LBM and discuss its suitability for parallel programming ands Conjugate Heat Transfer (CHT). We explore the origins of LBM from kinetic theory, examine the distribution function, and delve into the Boltzmann transport equation and its modeling for LBM. Additionally, we cover relaxation time models and various lattice arrangements essential for accurate simulations.

2.1 Introduction

Lattice Boltzmann method (LBM) has become an efficient numerical technique for simulating heat transfer and fluid flow problems. It is a particle based numerical solver and works at mesoscopic scale where a unit of particle collection is represented by a pseudo-particle distribution function [1,13]. LBM can be programmed parallelly owing to streaming and collision processes that influence the inter-particle interactions at mesoscopic scale [1,13]. The parallelized code utilizes the power of Graphics Processing Units (GPUs) to accelerate computations [14]. The fundamental working principle of LBM relies on Boltzmann transport equation that describes fluid behaviour using a statistical approach. Instead of dealing with each particle, a unit of particle collection in space and time is represented by a probability distribution function f. The distribution functions are located at discrete locations in space called lattice sites [1,6]. These lattice sites form a uniform grid on the domain of interest. The macroscopic state variables are obtained based on the values of distribution functions by computing the statistical moments of f [6]. This bridge between the macroscopic and mesoscopic scale relies on a primary understanding of Kinetic Theory. To comprehend and analyse LBM, the theoretical aspect of kinetic theory is crucial for information which will be discussed in next subsection. Introduced by Hardy et al. [15] as Lattice Gas Automata (LGA), LBM has evolved over time with several modifications and developments. From handling different boundary conditions like

adiabatic walls, no slip walls, zero pressure gradient etc. [16,17] to employing higher order and complex boundary conditions [18], LBM has made significant progress in solving problems such as turbulent flows [19], multi-phase flows [20], porous media flow [21], magneto-hydrodynamics [22] and conjugate heat transfer [23]–[28]. It has become popular due to its intrinsic advantages over conventional CFD solvers like finite difference and finite volume methods. Firstly, LBM does not require additional boundary treatments to model the transfer of heat and flux balance at fluid-solid interfaces making its implementation straightforward for such problems. Secondly, conventional solvers face challenges like time scaling, difficulty in handling large scale geometries etc.

In this thesis study, we employ LBM for modeling and solving conjugate heat transfer (CHT) for a two dimensional square shaped cylinder enclosed in a channel/duct. CHT is a coupled heat transfer process that occurs in a heterogeneous domain composed of solid and fluid regions [29]. Using LBM, the local thermodynamics and flux conservation are naturally satisfied at the solid-fluid interfaces [1,13] in CHT problems. Therefore, there is no need to trace the solid-fluid interfaces which eliminates the need for a conjugate boundary condition [30]. These inherent benefits of LBM make it ideal for solving CHT problems.

2.2 Kinetic Theory

Matter is known to be comprised of particles like atoms or molecules. They form the building blocks of nearly all the entities that exist in the universe that have a finite mass. Although, atoms/molecules are further composed of sub-atomic particles, the latter are not stable to exist independently and they combine to form atoms/molecules. These particles are always in random motion colliding with each other. For investigating the dynamics of particles, they are assumed to be infinitesimally small rigid spherical bodies whose motion is governed by classical laws such as Newton's laws, mass conservation, momentum and energy equations. As an example, if a molecule in space, $\mathbf{r}$ moving with a velocity, $\mathbf{c}$ at some time t is acted upon by an external force $\mathbf{F}$ as shown in Figure 2.1. From Newton's second law, we can write,

$$\mathbf{F} = \frac{d(m\mathbf{c})}{dt} \tag{2.1}$$

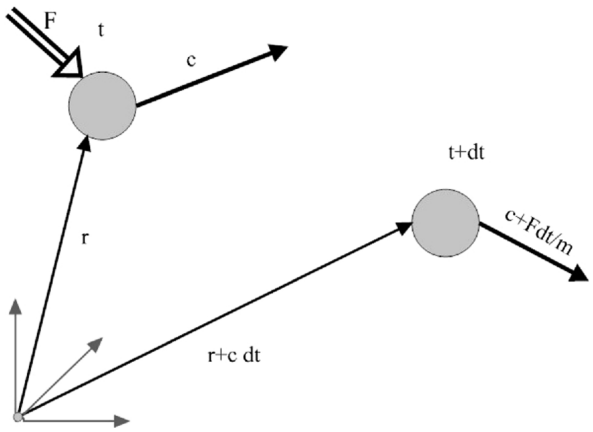


Figure 2.1 : A moving particle under external force [1]

If mass of particle is assumed constant, equation (2.1) becomes

$$\mathbf{F} = m\frac{d(\mathbf{c})}{dt} \tag{2.2}$$

The position and velocity of the particle after time $t + dt$ will change as follows:

$$\mathbf{r'} = \mathbf{r} + \mathbf{c}dt \tag{2.3}$$

$$\mathbf{c'} = \mathbf{c} + \mathbf{F}\frac{dt}{m} \tag{2.4}$$

Due to the absence of an external force $\mathbf{F}$ (see equation 2.4), the particle will move with same velocity $\mathbf{c}$. This implies that the particle velocity changes due to an external factor. For a system of particles, this increase in velocity of particles increases the total internal energy of system. At macroscopic scale, this increase of kinetic energy ($K.E$) of particles is manifested as an increase in pressure and temperature of the system [1,6]. Kinetic theory relates the pressure and temperature of a system to its kinetic energy using fundamental physical and gas laws. This can be derived assuming an ideal gas model. Consider a molecule in a container bombarding its walls as shown in Fig 2.2. Pressure is the measure of the force exerted per unit area of the wall. Assuming elastic collision between the wall and molecule, the magnitude of the impulse of force will be given as:

$$F\Delta t = mc_x - (-mc_x) = 2mc_x \tag{2.5}$$

Where $\Delta t$is the duration of collision. The duration of collision Δt will be equal to the time taken between two consecutive hits on the same wall.

$$\Delta t = \frac{2L}{c_x} \tag{2.6}$$

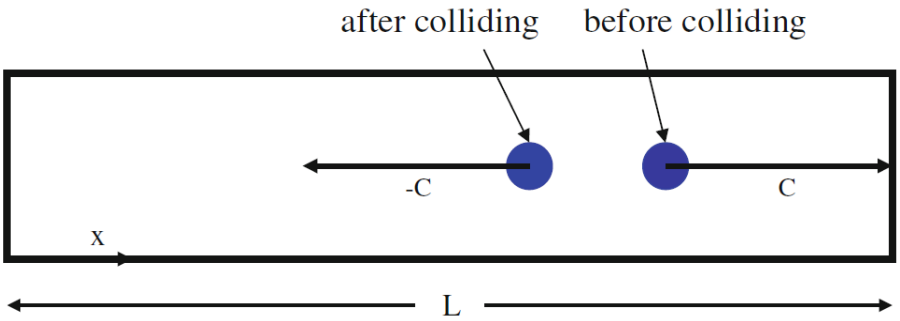


Figure 2.2 : Elastic collision in a particle system [1].

Using equation (2.6) in (2.5) yields the magnitude of force as:

$$F = \frac{mc_x^2}{L} \tag{2.7}$$

For N molecules in a cubic volume of side L, this result can be written in general form. The molecules will have speed, $c^2 = c_x^2 + c_y^2 + c_z^2$ where c_z, c_x and c_y are the components of particle's velocity in z, x, and y directions respectively. Equation (2.7) then becomes:

$$F = N\frac{mc^2}{3L} \tag{2.8}$$

If the area of wall is A, then pressure exerted on wall can be calculated as:

$$P = \frac{Nmc^2}{3LA} = \frac{Nmc^2}{3V} \tag{2.9}$$

Equation (2.9) can be rearranged as:

$$P = \frac{mc^2}{2}.\frac{2}{3}.\frac{N}{V} \tag{2.10}$$

where $\frac{mc^2}{2}$ is the measure of kinetic energy of a molecule. Let $n = N/V$ be number of molecules per unit volume. Then,

$$P = \frac{2}{3}nK.E \tag{2.11}$$

Temperature and pressure of gas are related by an equation of state:

$$PV = mRT \tag{2.12}$$

where m is total gaseous moles and R is the gas constant. Using equation (2.12) in (2.11), the kinetic energy of particle and temperature are related as:

$$K.E = \frac{3}{2}KT$$

where K is called Boltzmann's constant and is equal to the ratio of gas constant to Avogadro's constant ($K = R/N_A$). Thus, from equations (2.11) and (2.13), it can be seen that both pressure and temperature are actually different scales of kinetic energy of particles. Thus, kinetic theory attempts to reconcile the microscopic kinetic energy of particles with the macroscopic state variables of pressure and temperature. However, dealing with a large number of particles and formulating governing equations for every particle individually is challenging and requires huge computational resources. Instead, using a parameter to characterize the behavior of a bulk of particles will solve this complication.

2.2.1 Distribution function

Maxwell (1831-1879) introduced the concept of distribution function arguing that knowing the information about instantaneous velocity and location of the particles is not significant. Rather, the probability of particles in a specific location having a certain range of velocity at an instant of time is more crucial to depict the system's general behaviour. Maxwell-Boltzmann distribution function calculates the probability distribution of particles as a function of absolute temperature [31] as:

$$f(c) = 4\pi \left(\frac{m}{2\pi KT}\right)^{\frac{3}{2}} c^2 e^{-\frac{mc^2}{2KT}} \tag{2.14}$$

where T is absolute temperature of system, K is Boltzmann's constant and m is mass of particles. This function increases from zero (corresponding to particles in rest) to a maximum value (corresponding to particles moving at most probable speed) as a quadratic curve and then decreases exponentially as function of c as shown in Figure 2.3. Boltzmann generalised Maxwell's distribution function to possible energy states of system. He argued that equation (2.14) describes the distribution of an ideal gas when it has reached equilibrium. Boltzmann proved that distribution at a particular energy state (E) is proportional to $e^{-\frac{E}{KT}}$ [32] and is given by Boltzmann distribution as:

$$f(E) = Ae^{-}\frac{E}{KT} \tag{2.15}$$

The Boltzmann distribution function is expressed as:

$$f(c) = \left(\frac{m}{2\pi KT}\right)^{\frac{3}{2}} e^{-\frac{mc^2}{2KT}} \tag{2.16}$$

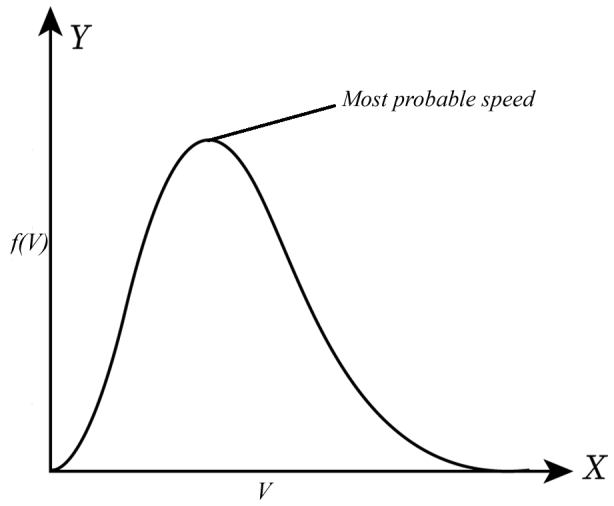


Figure 2.3 : Maxwell's distribution curve.

Boltzmann also showed that distribution function in equation (2.16) can be derived from Maxwell's distribution given in equation (2.14) from the integration of the latter function over a sphere's outer surface to account for all the possible velocity states of particles [1,31,32]. This laid the foundation for Boltzmann transport equation that is fundamental in the development of LBM.

2.2.2 Boltzmann transport equation

Boltzmann transport equation, BTE describes the particle dynamics of a system using a statistical approach by employing probability distribution function (PDF). Instead of solving for every particle in the system, distribution function (f) provides information about a fraction of particles moving with some velocity in space and time. From Figure 2.4, when an external force **F** acts on a system of particles, the position vector **r** and the velocity **c** will change. It is shown in equations (2.3) and (2.4). As a result of the external force, the PDF f of particles also changes. This differential change is calculated by BTE by employing a non-linear integral operator Ω that is a function of f. It is known as the collision operator and accounts for inter-particle collisions [13]. Mathematically, it is given as:

$$f(\mathbf{r},\mathbf{c},t) - f(\mathbf{r}+\mathbf{c}dt,\mathbf{c}+\mathbf{F}dt,t+dt) = -\Omega(f)\mathbf{drdc}dt \tag{2.17}$$

The differential form of f can be written in terms of **r**, **c** and t as:

$$df = \frac{\partial f}{\partial \mathbf{r}} \cdot \mathbf{dr} + \frac{\partial f}{\partial \mathbf{c}} \cdot \mathbf{dc} + \frac{\partial f}{\partial t} dt \tag{2.18}$$

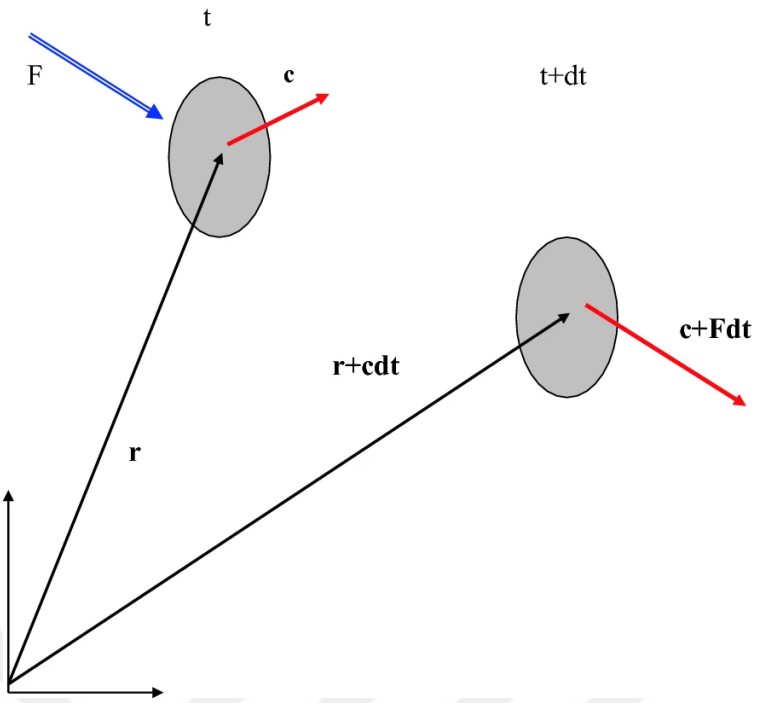


Figure 2.4 : Change in position and velocity under external force [1].

Time derivative of f can be calculated as:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{c}} \cdot \frac{d\mathbf{c}}{dt} + \frac{\partial f}{\partial \mathbf{r}} \cdot \frac{d\mathbf{r}}{dt} \tag{2.19}$$

Since $\mathbf{c} = \frac{d\mathbf{r}}{dt}$ and $\frac{\mathbf{F}}{m} = \frac{d\mathbf{c}}{dt}$ (see equations 2.3 and 2.4), equation (2.19) becomes:

$$\frac{df}{dt} = \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{c}} + \mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} + \frac{\partial f}{\partial t} \tag{2.20}$$

Now, multiplying equation (2.17) by $\frac{1}{dt\mathbf{drdc}}$, we get:

$$f(\mathbf{r} + \mathbf{c}dt, \mathbf{c} + \mathbf{F}dt, t + dt) - f(\mathbf{r}, \mathbf{c}, t) = \Omega(f)dt \tag{2.21}$$

Applying the limit on a small time instant dt, we get:

$$lt_{dt \to 0} - \frac{f(\mathbf{r}, \mathbf{c}, t) - f(\mathbf{r} + \mathbf{c}dt, \mathbf{c} + \mathbf{F}dt, t + dt)}{dt} = \Omega(f) \tag{2.22}$$

$$\implies \frac{df}{dt} = \Omega(f) \tag{2.23}$$

Since, equations (2.20) and (2.23) have the same term on the left side of equality sign, the terms on the right side of equality are equated to derive the partial differential form of BTE as:

$$\frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{c}} + \mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} + \frac{\partial f}{\partial t} = \Omega(f) \tag{2.24}$$

This equation is fundamental for understanding LBM. In the absence of an external force, equation (2.24) becomes:

$$\mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} + \frac{\partial f}{\partial t} = \Omega(f) \tag{2.25}$$

The final equation (2.25) is a first-order partial differential equation of BTE. Unlike Navier-Stokes equation which has many state variables, the refined BTE has two mesoscopic variables f and Ω. Macroscopic variables of state are derived from statistical moments of f. For application in fluid dynamics, macroscopic state variables of density (ρ), velocity ($\mathbf{V} = u\hat{i} + v\hat{j}$) and system's internal energy (e) are calculated using following:

$$\rho = m \int f \mathbf{dc} \tag{2.26}$$

$$\rho \mathbf{V} = m \int \mathbf{c} f \mathbf{dc} \tag{2.27}$$

$$\rho e = \frac{1}{2} m \int (c^2 - u^2) f \mathbf{dc} \tag{2.28}$$

The three equations (2.26), (2.27), and (2.28) are equivalent forms of mass conservation, momentum equation and energy equation in LBM respectively [33].

2.3 Relaxation Time Models

Solving equation (2.25) analytically is challenging as collision operator (Ω) is a complex function. Ω is a function of f and a complex integral [34]. Thus, for numerically solving the PDE, computational scientists have to rely on approximation models of Ω. For accurate and robust modeling of Ω, various models have been put forth. It began with a single relaxation time model (SRT) given in the 1980s followed by multi-relaxation time (MRT) models which developed in the mid 1990s. Since the 2000s, various other models like the entropic LB model [35], the cumulant LB model [36], and hybrid models [37] have emerged to improve the accuracy, convergence and stability of LBM solvers especially for capturing fluid behavior under non-ideal conditions. However, the SRT model pioneered by Bhatnagar, Gross and Krook (BGK) remains the most widely used approximation scheme due to its computational efficiency, linear form and reliable accuracy [38]. We will discuss the BGK model

followed by an introduction to the multi-relaxation time model to get an insight about the modeling of the collision operator.

2.3.1 Single relaxation time model

The model was proposed by L. Bhatnagar, P. Gross, and Max Krook. The BGK approximation models the collision operator Ω as:

$$\Omega = -\omega(f - f^{eq}) = -\frac{1}{\tau}(f - f^{eq}) \tag{2.29}$$

where ω is the frequency of collision, and its inverse τ is a constant relaxation time and f^{eq} is the distribution function at equilibrium. Introducing the BGK scheme for Ω in the force-free BTE (equation 2.25), we get,

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} = \frac{1}{\tau}(f^{eq} - f) \tag{2.30}$$

Discretising the above equation in space ($\mathbf{r}$) and time (t) along a specific velocity direction i, we get

$$f_i(\mathbf{r} + \mathbf{c}_i\Delta t, t + \Delta t) = \frac{\Delta t}{\tau}[f_i^{eq}(\mathbf{r},t) - f_i(\mathbf{r},t)] + f_i(\mathbf{r},t) \tag{2.31}$$

The discretised expression in equation (2.31) is called as lattice Boltzmann equation (LBE) and acts as an equivalent of the Navier-Stokes equations for solving fluid dynamics problems. For understanding the physical relevance of equation (2.31), it is considered as a combination of two mesoscopic processes. The processes of collision and streaming occur successively on the distribution function f and they govern the core of any LBM simulation. Collision updates the distribution functions f_i to account for the interaction between particles and are relaxed towards a state of equilibrium f_i^{eq}. The relaxation time is set by τ. The updated distribution functions f_i are calculated using the BGK approximation:

$$f_i(\mathbf{r},t) = \frac{1}{\tau} f_i^{eq}(\mathbf{r},t) + (1 - \frac{\Delta t}{\tau}) f_i(\mathbf{r},t) \tag{2.32}$$

Subsequently, the updated distribution functions f_i are propagated to neighboring nodes (lattice sites) depending on the lattice velocity $\mathbf{c}_i$. This process is called streaming and is mathematically given as:

$$f_i(\mathbf{r} + \mathbf{c}_i\Delta t, t + \Delta t) = f_i(\mathbf{r},t)$$

Figure 2.5 illustrates processes of collision and streaming. The distribution functions at the central node (shown in black color) and adjacent nodes (shown in grey color) are updated. The discrete components of f_i propagate to the adjacent lattice sites based on their corresponding lattice velocities, $\mathbf{c}_i$. This occurs at all the lattice sites in the computational domain simultaneously and continuously.

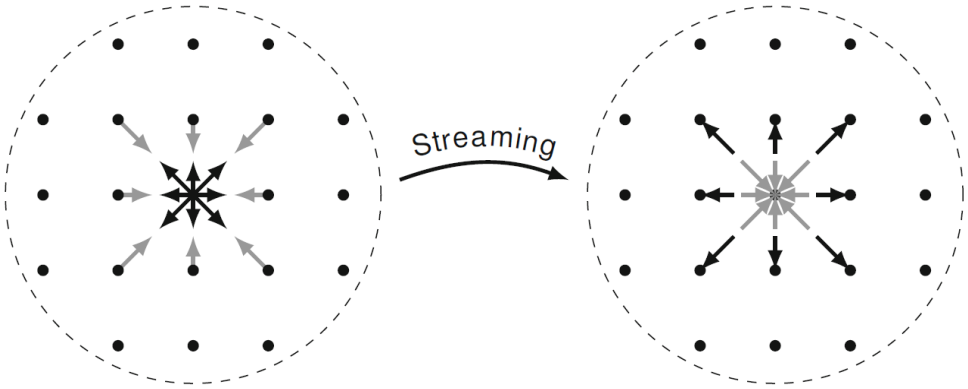


Figure 2.5 : Collision and Streaming [6].

2.3.2 Multi-relaxation time model

Multi-relaxation time model introduces different relaxation times for various modes of the distribution function. This offers flexibility and stability to LBM in handling a wider range of flow conditions especially at high Reynolds numbers. Unlike BGK-SRT model which uses a linear model for the collision operator, MRT employs a collision matrix [39]. The LBE is formulated as:

$$f_i(\mathbf{r}+\mathbf{c}_i\Delta t, t+\Delta t) - f_i(\mathbf{r},t) = -[Q](f_i(\mathbf{r},t) - f_i^{eq}(\mathbf{r},t)) \tag{2.34}$$

Transforming equation (2.34) to momentum space, we get:

$$f_i(\mathbf{r}+\mathbf{c}\Delta t, t+\Delta t) - f_i(\mathbf{r},t) = -[M]^{-1}[S](m_i(\mathbf{r},t) - m_i^{eq}(\mathbf{r},t)) \tag{2.35}$$

$[S]$ denotes the matrix for relaxation time, $[M]$ is the matrix of transformation and (m_i) is a vector where each of its element is a moment of the discrete components of f_i. $[S]$ is a diagonal matrix and $[M]$ is unique to each lattice arrangement. Lattice arrangements or models are discussed in the forthcoming subsection 2.4. For most two-dimensional problems in fluid flow and CHT, the D2Q9 lattice arrangement is the most popular and

widely used model [38]. Its transformation matrix $[M]$ is obtained [39] as:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & -1 & -1 & 1 \\ 0 & -2 & 0 & 0 & 2 & 0 & -1 & -1 & 1 \\ 0 & 0 & 1 & 1 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & -2 & -2 & 0 & 2 & 1 & -1 & -1 \\ 0 & 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \end{bmatrix}$$

The moment vector (m) is obtained from transformation matrix of (f) as:

$$(m) = [M](f) \tag{2.37}$$

$[S]$ is a 9 by 9 diagonal matrix with diagonal elements as:

$$[S] = \text{diag}\left(1, 1.4, 1.4, 1, 1.2, 1, 1.2, \frac{2}{1+6k}, \frac{2}{1+6k}\right) \tag{2.38}$$

where k depends on the physical property of the domain that is under consideration. For a fluid flow problem and a heat transfer problem, k is taken equal to the viscosity of the fluid (ν) and the thermal diffusivity (α) of the medium respectively [36]. Noticeably, the SRT model can be deduced from the MRT model when all diagonal elements of $[S]$ are taken equal to $\frac{1}{\tau}$ which is a constant relaxation time.

2.4 Lattice Modelling

For LBM application, a proper lattice arrangement should be chosen for an accurate description of the problem. Based on whether the problem is 1-D, 2-D or 3-D, different lattice arrangements have been put forth. For each dimension, a discrete set of velocities are chosen based on the specific requirement of the problem to be simulated. These models are typically chosen to satisfy certain criteria like isotropy, Galilean relativity or invariance and the ability to recover momentum equations (Navier-Stokes) in the continuum limit. Lattice models or arrangements are depicted in the form DnQm where n is the dimension of the domain and m is the velocity model. Forthcoming subsections describe the various lattice models/arrangements used for one-dimensional, two-dimensional and three-dimensional problems.

2.4.1 1-D lattice arrangements

Typically, three models are widely popular for one-dimensional problems. These are D1Q2, D1Q3 and D1Q5 lattice models. They are shown in Figure 2.6.

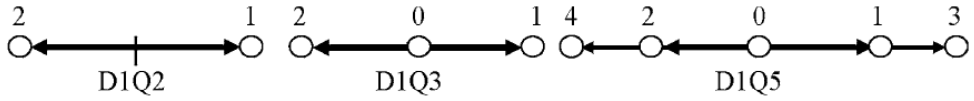


Figure 2.6 : One-dimensional lattice models [1].

D1Q2 has two lattice velocities each towards the left and the right direction. D1Q3 has an additional zero velocity while the higher order D1Q5 model has an additional zero and double magnitude velocities in the left and the right direction.Each of the velocity direction has an associated weighting factor. The lattice velocities $\mathbf{c}_i$ and corresponding weighting factors w_i are given in Table 2.1.

Table 2.1 : Velocity vectors and weighting factors for 1-D models [1].

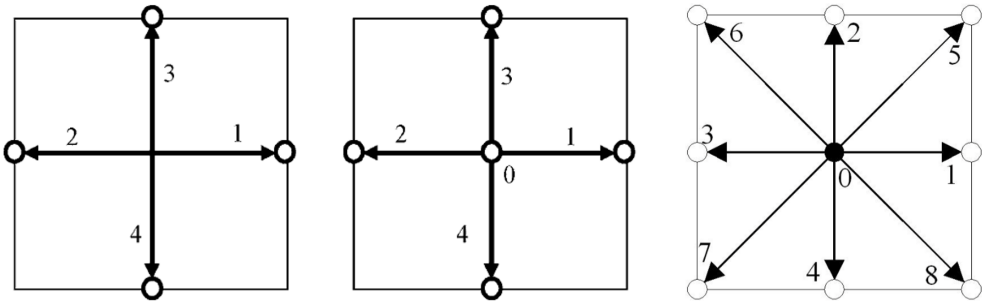
Direction	D1Q2		D1Q3		D1Q5	
i	$\mathbf{c}_i$	w_i	$\mathbf{c}_i$	w_i	$\mathbf{c}_i$	w_i
0	-	-	0	4/6	0	6/12
1	1	1/2	1	1/6	1	2/12
2	-1	1/2	-1	1/6	-1	2/12
3	-	-	-	-	2	1/12
4	-	-	-	-	-2	1/12

2.4.2 2-D lattice arrangements

In two-dimensional domains, D2Q4 and D2Q5 lattice models are used for problems involving convection and conduction of heat. D2Q9 is the popular lattice arrangement in 2-D due its advantage of conserving isotropy in all directions. This makes it ideal in 2-D problems for fluid flows and conjugate heat transfer. We will employ the D2Q9 lattice arrangement in our study of conjugate heat transfer.

Figure 2.7 shows the different 2-D lattice arrangements. The components of lattice velocity $\mathbf{c}_i$ along x-y directions and corresponding weighting factors w_i for 2-D lattice arrangements are tabulated Table 2.2. Higher order lattice models like D2Q15 offer

deeper refinement of flow behavior but come with additional computational costs in terms of memory and time.



D2Q4 D2Q5 D2Q9

Figure 2.7 : Two-dimensional lattice models [1,7].

Table 2.2 : Velocity vectors and weighting factors for 2-D models [1].

Direction	D2Q4		D2Q5		D2Q9	
i	$\mathbf{c}_i$	w_i	$\mathbf{c}_i$	w_i	$\mathbf{c}_i$	w_i
0	-	-	(0, 0)	2/6	(0, 0)	4/9
1	(1, 0)	1/4	(1, 0)	1/6	(1, 0)	1/9
2	(0, 1)	1/4	(0, 1)	1/6	(0, 1)	1/9
3	(-1, 0)	1/4	(-1, 0)	1/6	(-1, 0)	1/9
4	(0, -1)	1/4	(0, -1)	1/6	(0, -1)	1/9
5	-	-	-	-	(1, 1)	1/36
6	-	-	-	-	(-1, 1)	1/36
7	-	-	-	-	(-1, -1)	1/36
8	-	-	-	-	(1, -1)	1/36

2.4.3 3-D lattice arrangements

In three dimensional domains, D3Q15 and D3Q19 are the most employed lattice models. Due to the nature of 3-D problems, low order lattice arrangements are typically employed for initial evaluation.They are shown in Figure 2.8. The components of lattice velocities along the x-y-z directions and the corresponding weighting factors have been tabulated in Table 2.3. Rare higher order models such as D3Q27 are used when enhanced precision and resolution are prioritised over computational cost and time constraints.

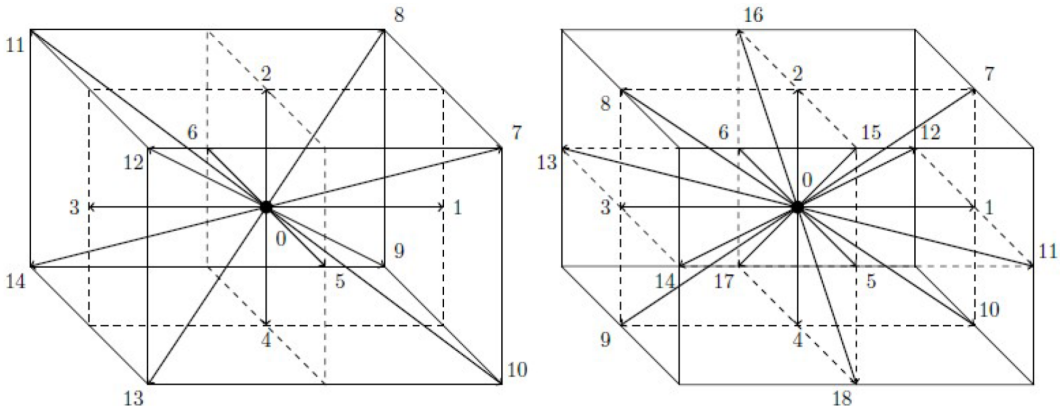


Figure 2.8 : Three-dimensional lattice models [2].

Table 2.3 : Velocity vectors and weighting factors for 3-D models [1,2].

Direction	D3Q15		D3Q19	
i	$\mathbf{c}_i$	w_i	$\mathbf{c}_i$	w_i
0	(0, 0, 0)	16/72	(0, 0, 0)	12/36
1	(1, 0, 0)	8/72	(1, 0, 0)	2/36
2	(0, 1, 0)	8/72	(-1, 0, 0)	2/36
3	(-1, 0, 0)	8/72	(0, 1, 0)	2/36
4	(0, -1, 0)	8/72	(0, -1, 0)	2/36
5	(0, 0, 1)	8/72	(0, 0, 1)	2/36
6	(0, 0, -1)	8/72	(0, 0, -1)	2/36
7	(1, 1, 1)	1/72	(1, 1, 0)	1/36
8	(1, 1, -1)	1/72	(1, -1, 0)	1/36
9	(1, -1, -1)	1/72	(1, 0, 1)	1/36
10	(1, -1, 1)	1/72	(1, 0, -1)	1/36
11	(-1, 1, -1)	1/72	(-1, 1, 0)	1/36
12	(-1, 1, 1)	1/72	(-1, -1, 0)	1/36
13	(-1, -1, 1)	1/72	(-1, 0, 1)	1/36
14	(-1, -1, -1)	1/72	(-1, 0, -1)	1/36
15	-	-	(0, 1, 1)	1/36
16	-	-	(0, 1, -1)	1/36
17	-	-	(0, -1, 1)	1/36
18	-	-	(0, -1, -1)	1/36

3. CONJUGATE HEAT TRANSFER

In this chapter, we introduce the concept of conjugate heat transfer (CHT) and discuss the numerical method for CHT using the Lattice Boltzmann Method (LBM). We also cover the implementation of stability and boundary conditions, which are crucial for accurate and reliable simulations.

3.1 Introduction

Conjugate heat transfer (CHT) is an approach involving the combined transfer of heat that occurs in a heterogeneous domain such a system composed of fluid and solid sub-domains [29]. Heat transfer in the fluid is governed by convection and in the solid by conduction. Practically, a number of real-life systems such as heat exchangers [40]–[42], combustion chambers [43], aero-engine blades [44], electrical equipment [45], micro and nano-systems like micro-fuel cells etc. [46] come under the application of CHT amongst a wide range of fields [47]. In CHT, heat transfer in the solid and the fluid occurs across the solid-fluid interface. Conventional CFD solvers need an accurate boundary condition to describe and quantify the governing mechanism for heat transfer at such boundaries. Employing traditional boundary conditions such as constant heat flux and constant wall temperature are not adequate in describing the overall transfer of heat in a system as it neglects the heat transfer by conduction within the solid region [48]. Therefore, a combination of Dirichlet and Neumann-like boundary condition is formulated to address the heat flux balance and the continuous distribution of temperature across the fluid-solid interface [49]. These boundaries are known as conjugate boundaries and the boundary conditions used are called conjugate boundary conditions [50]. Conjugate boundary condition uphold the law of energy conservation and local thermodynamics at the interface [29]. However, for large-scale systems, complex interface boundaries etc., CFD solvers become inefficient due to challenges like iterative coupling and proper time scaling

between solid and fluid sub-domains. The monolithic approach provides a strong coupling and a global solution at the expense of a high computational cost and on the other hand, the partitioned approach provides flexibility and reduces computational effort at the expense of a high computational time [30].

The application of lattice Boltzmann method to CHT problems has become popular in recent times. LBM solves the conjugate heat transfer problems efficiently without the need for a conjugate boundary condition at the fluid-solid interface. This potential of LBM was first shown by Wang et al. [47]. They showed that LBM can solve CHT problems more efficiently without needing a conjugate boundary condition. This applies well to interface geometries that are straight and in the low Reynolds number flow regime [29,47]. Further developments were made by Li et al. [51] who modeled conjugate heat problems for arbitrary interfaces and Hu et al. [52] who proposed an immersed boundary-like scheme to deal with curved interfaces. In this study, we will be dealing with straight interfaces and boundaries which allow us to fully take advantage of LBM for parallel computing [29]. The advancements in the field of parallel computing have accelerated time efficiency which makes LBM ideal for large-scale problems as well.

3.2 Numerical Method for Conjugate Heat Transfer via LBM

In LBM, conjugate heat transfer is modeled using two discretised Boltzmann transport equations separately. BTE (see equation 2.31) has to be modeled independently for viscous flow and transfer of heat [53]. This method is popular for integrating the energy equation with LBM. It involves introducing another equation that yields a convection-conduction equation to govern heat transfer mechanism in both fluid and solid sub-domains together [29,33,47].

3.2.1 Governing equations for viscous fluid flow and convection-conduction

The distribution functions f and g are employed to represent the fluid flow and temperature distribution respectively. The two BTE equations are solved simultaneously. A single computational framework is used for running both the flow and heat solvers. The evolution of both equations for flow and transfer of heat is

concurrent. There exists one way coupling between the two equations where BTE for fluid flow is independent but BTE for heat transfer relies on the flow state variable of velocity $\mathbf{V}$ for calculating the equilibrium distribution function, g^{eq}. This coupling is essential for the fluid flow to influence the temperature distribution and transfer of heat in CHT problems. The discretised Boltzmann transport equations governing viscous flow and transfer of heat via convection-conduction are expressed using f and g respectively as:

$$f_i(\mathbf{r}+\mathbf{c}_i\Delta t, t+\Delta t) = \frac{1}{\tau_f} f_i^{eq}(\mathbf{r},t) + (1-\frac{\Delta t}{\tau_f}) f_i(\mathbf{r},t) \tag{3.1}$$

$$g_i(\mathbf{r}+\mathbf{c}_i\Delta t, t+\Delta t) = \frac{1}{\tau_g} g_i^{eq}(\mathbf{r},t) + (1-\frac{\Delta t}{\tau_g}) g_i(\mathbf{r},t) \tag{3.2}$$

The step size for time (Δt) and space ($\Delta x, \Delta y$) in LBM is generally taken as equal to 1 for simplicity and numerical stability. It allows for an easier interpretation of results and ensures the time scale of lattice corresponds directly with the physical time scale. For $\Delta t = 1$, the governing equations (3.1) and (3.2) are rewritten w.r.t collision frequency ω as:

$$f_i(\mathbf{r}+\mathbf{c}_i\Delta t, t+\Delta t) = \omega_f f_i^{eq}(\mathbf{r},t) + (1-\omega_f) f_i(\mathbf{r},t) \tag{3.3}$$

$$g_i(\mathbf{r}+\mathbf{c}_i\Delta t, t+\Delta t) = \omega_g g_i^{eq}(\mathbf{r},t) + (1-\omega_g) g_i(\mathbf{r},t) \tag{3.4}$$

The frequency of collision for fluid flow ω_f is given by the inverse relation to fluid viscosity [29,54] as:

$$\omega_f = \frac{1}{3\nu + 0.5} \tag{3.5}$$

As the D2Q9 lattice arrangement is chosen for our CHT study, macroscopic state variables for fluid flow at a lattice site are computed using density (mass) and momentum conservation by calculating zeroth and first-order moments of f [6] as:

$$\rho = \sum_{i=0}^{8} f_i \tag{3.6}$$

$$\rho\mathbf{V} = \sum_{i=0}^{8} \mathbf{c}_i f_i \tag{3.7}$$

Similarly, the frequency of collision for the temperature distribution ω_g is inversely related to the coefficient of diffusivity α of material as:

$$\omega_g = \begin{cases} \frac{1}{3\alpha_{fluid}+0.5} & \textit{in fluid region} \\ \frac{1}{3\alpha_{solid}+0.5} & \textit{in solid region} \end{cases} \tag{3.8}$$

The magnitude of temperature is computed from the zeroth moment of g_i [6,54] at a lattice site as:

$$T = \sum_{i=0}^{8} g_i \tag{3.9}$$

3.2.2 Equilibrium distribution function

It represents the theoretical distribution of particles at each lattice site under local equilibrium conditions. It stems from Maxwell-Boltzmann distribution to characterise fluid behaviour under flow and thermal equilibrium. As we know, macroscopic state variables of velocity ($\mathbf{V}$), temperature (T) and density (ρ) are computed from the distribution functions by first and second order moments of respective distribution functions. Under non-equilibrium conditions, the local distribution functions f and g tend to move towards their equilibrium values f^{eq} and g^{eq} respectively. The collision process given in equation (2.32) aims to achieve this objective by evolving the fluid towards equilibrium state locally. That is to say, equilibrium distribution function acts as a reference for the local distribution functions to evolve to a state of equilibrium distribution and the difference between them represents the non-equilibrium component of flow and heat transfer.

In principle, the equilibrium distribution function imposes mass, momentum and energy conservation laws on the distributions function to be fulfilled at a macroscopic scale. [29,47]. This is achieved by Hermite series expansion. f_{eq} upon expansion retrieves the macroscopic laws [55]. In LBM simulations, equilibrium distribution function f_i^{eq} is evaluated along lattice velocity direction i at every node location. For viscous flow, f_i^{eq} [25,54] is computed as:

$$f_i^{eq}(\mathbf{r},t) = w_i \rho(\mathbf{r},t) \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{V}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{V})^2}{2c_s^4} - \frac{\mathbf{V} \cdot \mathbf{V}}{2c_s^2}\right) \tag{3.10}$$

where w_i is the corresponding weighting factor of lattice velocity $\mathbf{c}_i$; ρ and $\mathbf{V}$ are the density and velocity of fluid at the node location respectively; and c_s is the lattice speed of sound. Similarly, for the LBE governing convection-conduction, the discrete equilibrium distribution function g_i^{eq} is calculated w.r.t T and $\mathbf{V}$ [54] as:

$$g_i^{eq}(\mathbf{r},t) = w_i T(\mathbf{r},t) \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{V}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{V})^2}{2c_s^4} - \frac{\mathbf{V} \cdot \mathbf{V}}{2c_s^2}\right) \tag{3.11}$$

In the solid region of the domain, $\mathbf{V} = 0$ for all of the lattice sites. Consequently, the LBE for heat transfer in the solid region naturally governs heat conduction without any additional considerations. The equilibrium distribution function g_i^{eq} in the solid becomes:

$$g_i^{eq}(r,t) = w_i T(r,t) \tag{3.12}$$

As earlier described in chapter 2, $\mathbf{c}_i$, w_i and c_s are unique to every lattice arrangement. The propagation speed of sound c_s is crucial in determining the maximum speed at which information propagates through the lattice grid influencing the stability and accuracy of the LBM simulations. In LBM, c_s is a constant that relates mesoscopic density and pressure through an equation of state:

$$p = c_s^2 \rho \tag{3.13}$$

For D1Q2, D2Q4 and D3Q6 lattice arrangements [1,6]:

$$c_s = \frac{1}{\sqrt{2}} \tag{3.14}$$

For D1Q3, D2Q5, D2Q9, D3Q15 and D3Q19 lattice arrangements [1,6,29]:

$$c_s = \frac{1}{\sqrt{3}} \tag{3.15}$$

3.2.3 Setting Mach and Reynolds number for numerical stability

Unlike other numerical solvers, in LBM, the computational stability of simulations is non-trivial. Only at low Mach, LB equation can recover the Navier-Stokes equation. When employing BGK approximation for Ω, the numerical solver will remain stable only if the parameters like maximum velocity component, relaxation time (τ), Reynolds number etc. do not breach the stability limits. Firstly, one needs to match the relevant dimensionless parameters between the physical scale and mesoscopic scale. Instability in LBM simulation arises usually when distribution functions and state variables grow exponentially. This happens due to the wrong choice of parameters. To stay within the stability limits, necessary conditions serve as a guide to avoid the dreaded "Not a Number" (NaN) abyss and keeping the simulations afloat. In LBM, the step size for space ($\Delta x, \Delta y$) and time (Δt) is generally set to 1 for simplicity. The following stability conditions must be fulfilled for achieving stable numerical solutions using LBM [6]:

- $f_i, f_i^{eq} > 0$ (must always initialise to non-zero values)
- $\frac{\tau}{\Delta t} > \frac{1}{2}$
- Maximum velocity magnitude $|u_{max}|$ should not exceed the speed limit in a lattice model.

  $|u_{max}| < \sqrt{\frac{2}{3}}\frac{\Delta x}{\Delta t}$ for D1Q2, D1Q3, D2Q4, and D2Q5.

  $|u_{max}| < \sqrt{\frac{1}{3}}\frac{\Delta x}{\Delta t}$ for D2Q9, D3Q15, D3Q19, and D3Q27.

  Stability behavior of velocity was illustrated by Kruger et al. [6] as shown in Figure 3.1.

The lattice sites (nodes) in x-direction along the length of cylinder (L) are given by $N = \frac{L}{\Delta x} = L$. We know that Reynolds number in physical scale is given by:

$$Re = \frac{U_\infty L}{\nu} \tag{3.16}$$

where L is the characteristic length, U_∞ is free stream velocity and ν is fluid viscosity. For mesoscopic scale, Reynolds number is computed as:

$$Re = \frac{U_L N}{\nu_L} \tag{3.17}$$

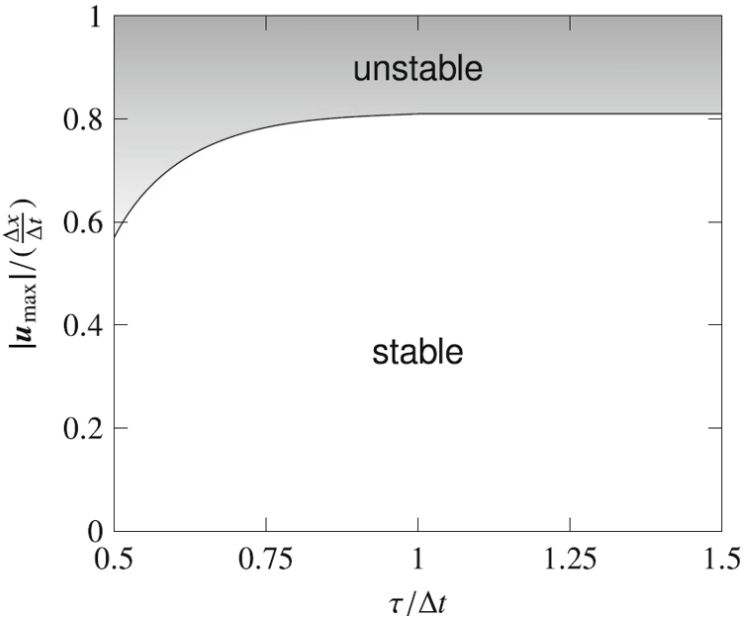


Figure 3.1 : Stable and unstable regions for 2-D and 3-D lattice models [6].

The velocity U_L in lattice scale is usually in the magnitude range of 0.01-0.2 for a stable solution. Based on Re and U_L, ν_L can be calculated.

Mach number, M in LBM should be chosen as low as possible for incompressible viscous flow. It is calculated as:

$$M = \frac{\Delta x}{N\sqrt{3}}(\frac{\Delta t}{\tau} - \frac{1}{2})Re \tag{3.18}$$

The order of truncation error in LBM is M^2 [1]. Thus, N and τ are chosen to achieve low Mach values.

3.3 Boundary Conditions in LBM

In LBM, applying boundary conditions employs a non-trivial approach unlike CFD where conditions for state variables at the boundaries are implemented directly. In LBM, instead of using state variables, unknown (incoming) distribution functions are evaluated at the boundaries. Various approaches have been put forth for accurate evaluation of unknowns at the lattice boundaries [25,56]–[60]. Researchers have tested and refined techniques to correctly describe and quantify the physical boundary condition for LBM applications. In our conjugate heat transfer simulations, two sets of boundary conditions are needed. One for the flow solver to address state variables of velocity and density (pressure) at the boundaries and the another for the heat transfer solver pertaining to temperature at the boundaries. The popular approaches to tackle the most commonly used boundary conditions in LBM are discussed ahead.

3.3.1 Boundary conditions for fluid flow

For implementing flow boundary conditions, unknown distribution function, f at the boundaries is calculated in terms of the flow state variables. The various approaches to implement flow boundary conditions are discussed in this sub-section.

3.3.1.1 Bounce back scheme

It is used for modeling a static or moving solid boundary in a flow domain. Conventional CFD methods use no-slip condition at the boundaries where fluid is at rest ($\mathbf{V} = 0$) e.g. on channel walls, at the boundary where fluid encounters an obstacle

etc. In LBM, to model no slip and free slip conditions [1,6], bounce back method is a popular approach. In no-slip condition, the parallel and perpendicular components of velocity at the wall are zero whereas in free-slip, the perpendicular component of velocity at wall is only equal to zero. Condition of free-slip is mainly used to model fluid flow in 3D problems like flow over a river bed etc. [60]. For implementing no-slip at walls, bounce back ensures that outgoing distribution functions, f_i near the solid wall is reflected back into the flow domain. There are three popular variations of implementing bounce back scheme in the literature with differing views about the accuracy level of each one [16,61]–[63].

In bounce back scheme I, the location of the solid boundary or wall is halfway from the nearest lattice sites. Figure 3.2 demonstrates that the wall lies at a distance $\Delta y/2$ from the adjacent lattice node. From the definition of bounce back, the outgoing distribution functions at the boundary, f_4, f_7 and f_8 are reversed and returned back into the fluid. The unknown distribution functions (see dotted arrows), f_2, f_5 and f_6 are equated to their opposite distribution functions as: $f_5 = f_7$, $f_2 = f_4$ and $f_6 = f_8$.

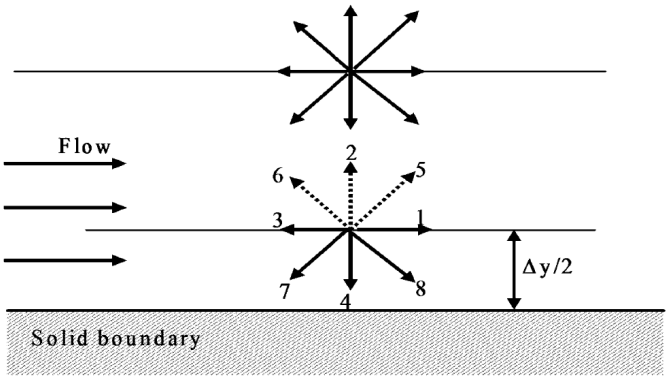


Figure 3.2 : Bounce back scheme I [1].

Bounce back scheme II places the wall or solid boundary midway between two adjacent lattice sites. As can be seen in Figure 3.3, this variation of bounce back takes place inside the wall as the domain is extended. With the same set of known and unknown distribution functions as in bounce back I, it is formulated: $f_6 = f_8$, $f_5 = f_7$ and $f_2 = f_4$. Remarkably, in this scheme, the lattice site under consideration depends on the reflected distribution functions from adjacent lattice sites in the wall.

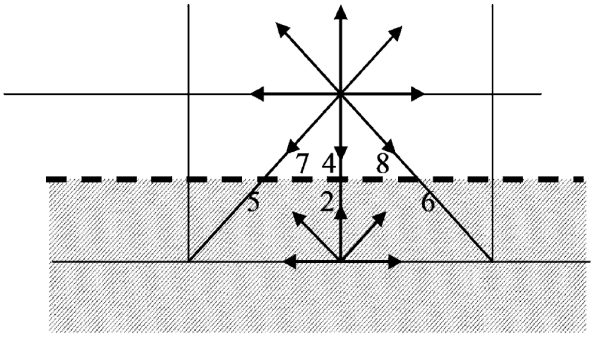


Figure 3.3 : Bounce back scheme II [1].

Bounce back scheme III takes the direct approach and overlaps the solid boundary exactly on the lattice site as shown in Figure 3.4. Similar to previous schemes, $f_2 = f_4$, $f_6 = f_8$ and $f_5 = f_7$. This scheme has simpler application in comparison to schemes I and II.

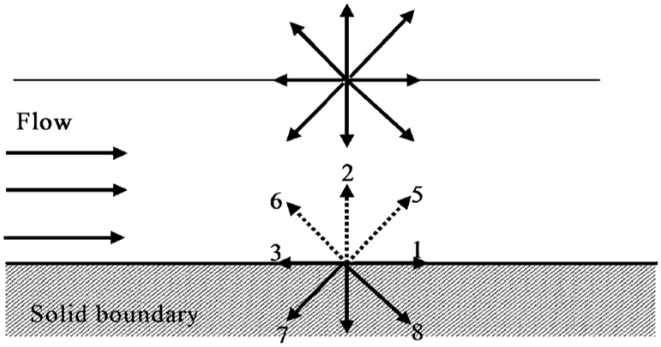


Figure 3.4 : Bounce back scheme III [1].

In order to ensure conservation of mass and momentum, bounce back schemes are implemented after the distribution functions are streamed to neighbouring lattice sites.

3.3.1.2 Velocity boundary condition

When a Dirichlet boundary condition such as constant velocity or density is specified, the unknown (incoming) distribution functions on boundary are calculated from a system of equations. This method is formally known as non-equilibrium bounce back (NEBB) method and popularly called as Zou-He method. It is independent of relaxation parameter with accuracy level of third order [56,58]. To illustrate this concept, we take a domain with north, south, east and west walls as depicted in Figure 3.5. The unknown distributions along boundaries are shown in dotted lines.

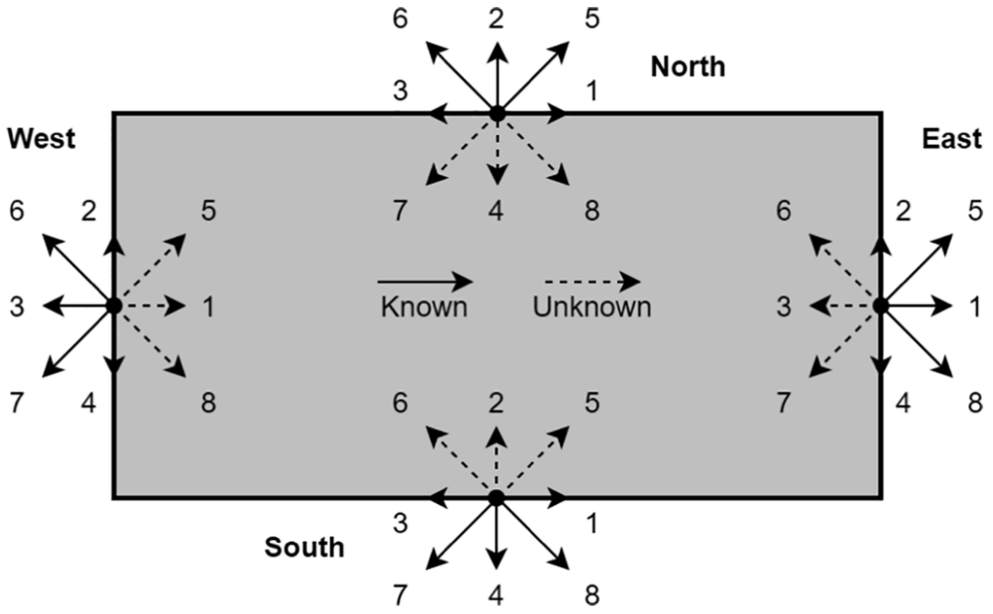


Figure 3.5 : Distribution functions at the boundaries of domain [8].

a) West boundary: At the west boundary, if velocity is specified (say $u = u_w$, $v = v_w$), we can expand equations (3.6) and (3.7) as follows:

$$\rho = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 \tag{3.19}$$

$$\rho u_w = f_1 + f_5 + f_8 - f_6 - f_3 - f_7 \tag{3.20}$$

$$\rho v_w = f_2 + f_5 + f_6 - f_7 - f_4 - f_8 \tag{3.21}$$

The number of unknowns at the west boundary i.e. ρ, f_1, f_5, and f_8 exceeds the total number of equations. The NEBB method provides values for the unknowns by enforcing an additional constraint. The non-equilibrium component of distribution functions normal to the wall should satisfy:

$$f_1^{eq} - f_1 = f_3^{eq} - f_3 \tag{3.22}$$

where f_1^{eq} and f_3^{eq} can be calculated from equation (3.10). With 4 equations and 4 unknowns, the system of equations is solved and yields:

$$\rho_w = \frac{f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)}{1 - u_w} \tag{3.23}$$

$$f_1 = f_3 + \frac{2}{3}\rho_w u_w \tag{3.24}$$

$$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_w u_w + \frac{1}{2}\rho_w v_w \tag{3.25}$$

$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_w u_w - \frac{1}{2}\rho_w v_w \tag{3.26}$$

In case density at the boundary is specified, we follow the same procedure and solve for velocity at the wall by back substitution. On the rest of the boundaries of the domain, the unknown quantities can be calculated using the same formulation.

b) North Boundary:

$$\rho_n = \frac{f_0 + f_1 + f_3 + 2(f_2 + f_6 + f_5)}{1 + v_n} \tag{3.27}$$

$$f_4 = f_2 - \frac{2}{3}\rho_n v_n \tag{3.28}$$

$$f_7 = f_5 + \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_n v_n - \frac{1}{2}\rho_n u_n \tag{3.29}$$

$$f_8 = f_6 - \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_n v_n + \frac{1}{2}\rho_n u_n \tag{3.30}$$

c) East Boundary:

$$\rho_e = \frac{f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)}{1 + u_e} \tag{3.31}$$

$$f_3 = f_1 - \frac{2}{3}\rho_e u_e \tag{3.32}$$

$$f_7 = f_5 + \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_e u_e - \frac{1}{2}\rho_e v_e \tag{3.33}$$

$$f_6 = f_8 - \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_e u_e + \frac{1}{2}\rho_e v_e \tag{3.34}$$

d) South Boundary:

$$\rho_s = \frac{f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)}{1 - v_s} \tag{3.35}$$

$$f_2 = f_4 + \frac{2}{3}\rho_s v_s \tag{3.36}$$

$$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_s v_s + \frac{1}{2}\rho_s u_s \tag{3.37}$$

$$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho_s v_s - \frac{1}{2}\rho_s u_s \tag{3.38}$$

3.3.1.3 Open boundary condition

Open boundaries ensure that flow at the outlet remains uniform i.e. zero gradient in the flow direction. Enforcing constant density or velocity at the outlet can cause pressure waves to reflect back into the domain increasing computational time to reach the desired solution. This is an inherent feature to LBM [6] which cannot be avoided.

Various advanced strategies have been put forth and the research is ongoing because there is still no single approach to tackle open-boundaries. The general method for tackling open boundaries is [1] as follows:

$$f_{i,n} = 2f_{i,n-1} - f_{i,n-2} \tag{3.39}$$

This scheme is used when domain length in the direction of flow velocity is sufficient to dampen the flow variations.

3.3.2 Boundary conditions for heat transfer

For implementing boundary conditions in heat solver, unknown distribution function, g at the boundaries is calculated in terms of the temperature . The various approaches to implement temperature boundary conditions are discussed in this sub-section.

3.3.2.1 Adiabatic wall

For adiabatic condition on a boundary, the temperature gradient T perpendicular to the wall is taken as zero. For example, on the north and south walls of the domain (see Figure 3.5), $\frac{dT}{dy} = 0$ whereas on the west and east walls, $\frac{dT}{dx} = 0$. The unknown (incoming) distribution functions, for instance on the north boundary having coordinates (x, N_y), are computed from first order extrapolation [25] as:

$$g_i(x, N_y) = g_i(x, N_y - 1) \tag{3.40}$$

where $i = 2$, 5 and 6. Similarly, the unknown distribution functions can be calculated on the other boundaries.

3.3.2.2 Temperature boundary condition

For a Dirichlet boundary condition such as constant temperature T_w specified on the wall, a flux balance at the boundary is implemented [1,25]. For instance, at the west boundary, we calculate the unknown distribution functions g_1, g_5, and g_8 as:

$$g_1 = g_1^{eq} + g_3^{eq} - g_3 \tag{3.41}$$

$$g_5 = g_5^{eq} + g_7^{eq} - g_7 \tag{3.42}$$

$$g_8 = g_8^{eq} + g_6^{eq} - g_6 \tag{3.43}$$

g_3, g_7 and g_6 are known and their equilibrium distribution functions are calculated using equation (3.11). In case the fluid velocity is zero at the wall, the above equations reduce to:

$$g_1 = w_1 T_w + w_3 T_w - g_3 \tag{3.44}$$

$$g_5 = w_5 T_w + w_7 T_w - g_7 \tag{3.45}$$

$$g_8 = w_8 T_w + w_6 T_w - g_6 \tag{3.46}$$

3.3.3 Additional considerations

Here, we delve into some aspects of corner treatment and the fluid-solid interface pertinent to conjugate heat transfer. These considerations are essential for ensuring accurate numerical simulations.

3.3.3.1 Corner treatment

The solution of the flow and heat transfer solvers should be smooth at the boundary corners. Although, corners account for a few lattice sites, their significance is important. Each corner needs to be modeled properly so that it does not pollute the numerical solution. Some approaches to deal with corners were put forth by Maier et al. [59] and Zou and He [58] for 2D cases. At the corner lattice sites, conservation of heat flux, mass and momentum should be ensured. Zou et al. [58] has shown that the unknown distribution functions at corners can be derived by enforcing a constant density ρ. As an example, the unknown distribution functions at the south-west corner are calculated as follows:

$$f_1 = f_3 \tag{3.47}$$

$$f_2 = f_4 \tag{3.48}$$

$$f_5 = f_7 \tag{3.49}$$

$$f_6 = f_8 = \frac{1}{2}(-(f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f7) + \rho) \tag{3.50}$$

For the heat transfer case, it is assumed that corners do not contribute any energy to the wall nodes [59]. For instance, at the south-east corner, unknown distributions g_1,

g_2 and g_5 are balanced using the bounce back scheme. The distribution couplet g_6 and g_8 are assumed constant by averaging their values from the previous time step as:

$$g_{6,t} = g_{8,t} = \frac{1}{2}(g_{6,t-1} + g_{8,t-1}) \tag{3.51}$$

3.3.3.2 Fluid-solid interface

The interface between the fluid and solid body does not need any special treatment in LBM. The values of ρ, $\mathbf{V}$, and T in the fluid and only T in the solid body are initialised and tracked for each iteration. This is sufficient for governing the heat transfer mechanism at the fluid-solid interface [47].

4. NUMERICAL STUDY AND POST-PROCESSING

In this chapter, we start our numerical study of conjugate heat transfer (CHT). First, we validate our flow and heat solver independently. Then, we introduce the problem definition of our CHT problem. The post-processing of results is conducted, providing the outcomes from 64 runs, which were performed considering 4 Reynolds numbers, 4 solid to fluid conductivity ratios, and 4 Prandtl numbers. In Lattice Boltzmann Method (LBM) studies, non-dimensional units are often used to simplify simulations and make results more generalizable. This approach allows researchers to compare results across different scales and conditions without converting to physical units.

4.1 Validation

To ensure the authenticity and credibility of our numerical solver, we validate the flow solver and heat solver results independently. Due to the unavailability of a CHT study for time periodic flows, it is essential that the validation process is carried out with reference studies for flow field and heat transfer separately.

4.1.1 Flow solver validation

To validate the flow solver, we rely on numerical benchmarks provided in reference studies. One notable benchmark is evaluating the distribution of pressure on solid cylinder's surface. This serves as a reliable reference for numerical validation of the flow solver. L Zhou et al. [9] have previously investigated different cases of flow control mechanisms for a 2-D square cylinder. They have provided valuable insights and data about the distribution of pressure for various cases of flow control on the surface of square cylinder. The case of a square cylinder without no flow control is of particular interest and similar to our problem. While leveraging Zhou's results as a benchmark, necessary modifications in our problem setup are made to ensure that we replicate the same conditions for conducting the simulation. We modify parameters

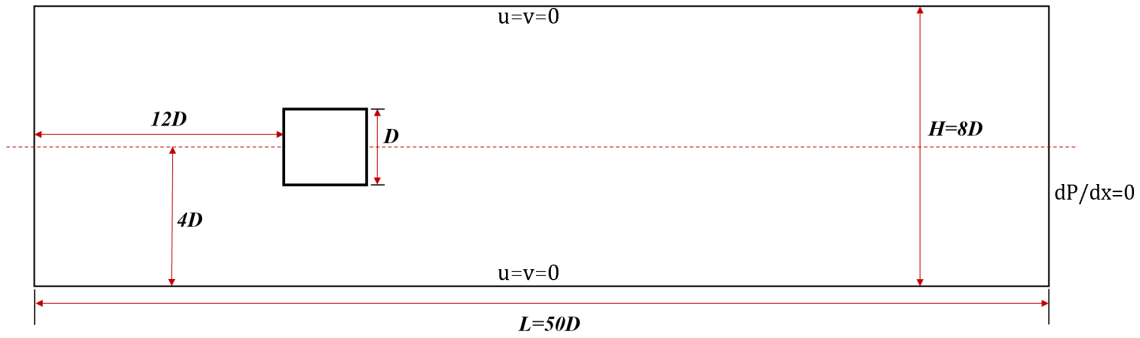


Figure 4.1 : Geometric details of Zhou's setup [9].

such as Re and H/D (blockage ratio) to match the conditions for validation as shown in Figure 4.1. The blockage ratio and Re are taken as 8 and 250 respectively with a fully developed flow entering the channel with $u_{max} = 0.1$. This ensures that our system is similar to Zhou's 2D system. Validation of flow solver result is illustrated in Figure 4.2. The simulation results of our flow solver aligns with the data from the reference study and deviates by a maximum error of 1.89%.

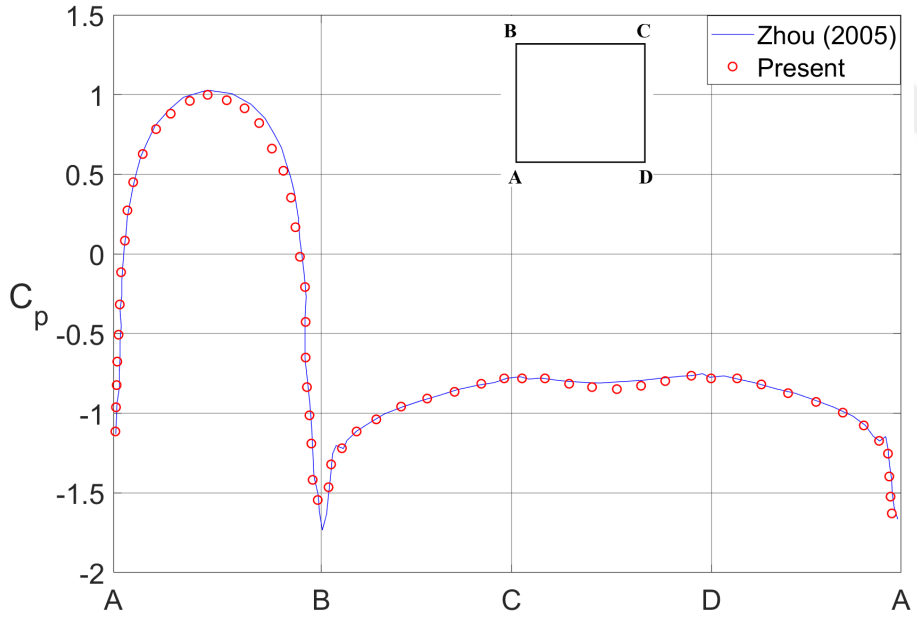


Figure 4.2 : Validation of pressure coefficient for square cylinder.

4.1.2 Heat solver validation

For validation of the heat solver results, we assess their accuracy with the reference study by Eswaran et al. [10]. They have conducted extensive and elaborate analyses

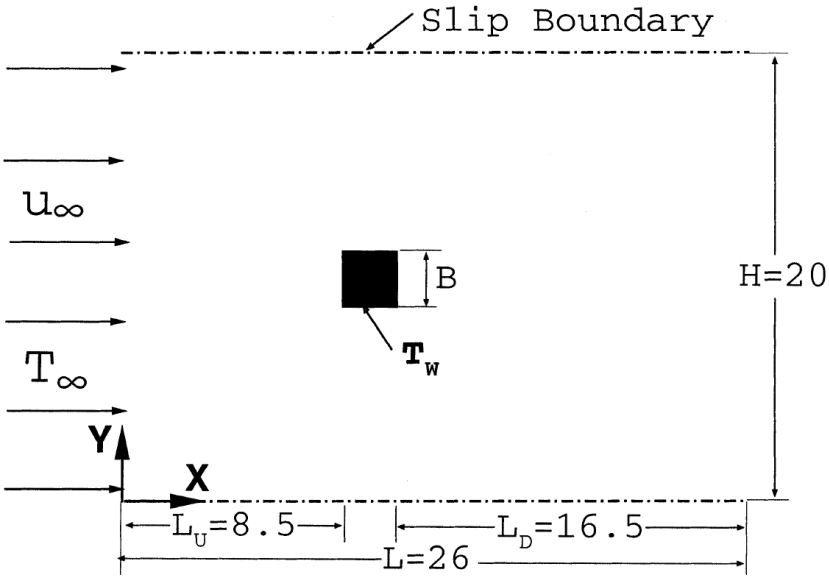


Figure 4.3 : Geometric details of Eswaran's setup [10].

for the transfer of heat for a two-dimensional square cylinder in steady and unsteady laminar (time periodic) flow regimes. The cylinder at a fixed temperature was confined in flow to observe the transfer of heat at the interfaces. The geometric details of setup are shown in Figure 4.3. A uniform velocity profile of $u_\infty = 0.1$ enters at a temperature, $T_\infty = 0$ while the cylinder is at a fixed temperature, $T_w = 1$.To quantify the transfer of heat taking place on solid-fluid interfaces, Eswaran et al. [10] calculated the Nusselt number, (Nu) on the interfaces. It is a dimensionless number defined as ratio of the transfer of heat via convection to the transfer of heat via conduction in the fluid. Following their prescribed formulation, we calculate the nusselt numbers on square faces using our LBM solver. For steady laminar flow, Nu over the cylinder's face is calculated as a surface-average of the local nusselt numbers Nu_L. For time periodic flow, Nu is also time averaged over a period of vortex shedding [64] as shown:

$$Nu = \frac{1}{\Gamma}\int_0^\Gamma (Nu)dt \tag{4.1}$$

We compute the nusselt numbers for the square cylinder's leading edge(front face), trailing edge (rear face), bottom and top face together, and average of all the faces. To validate our results accurately, Re are chosen discretely between the range 10-160, and $Pr = 0.7$ for replicating the conditions of the reference study. Our results show close similarity with the reference study as shown in Figure 4.4 and acceptable error

margins. The maximum errors are 1.1% for the front face, 1.41% for the rear face, 1.79% for the top and bottom faces, and 1.2% for average of all the faces.

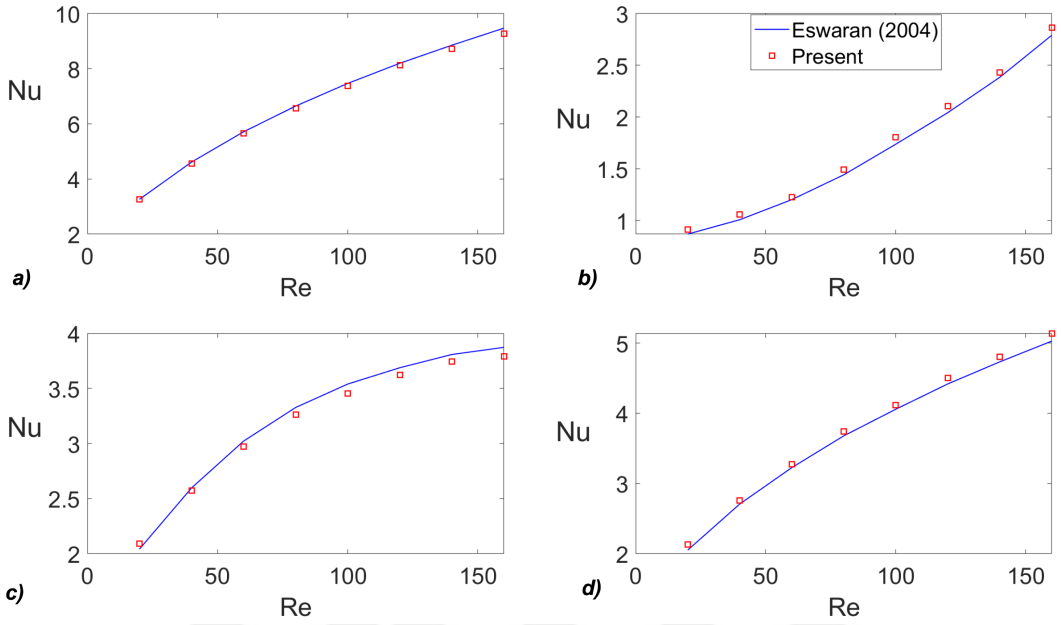


Figure 4.4 : Validation of Nusselt number for: a) front face b) rear face b) bottom and top faces d) average of all the faces of the square cylinder with varying *Re*.

4.2 Problem Definition

For the study of conjugate heat transfer (CHT) using LBM, we have adapted a popular benchmark problem in CFD i.e. a cylinder embedded in flow and enclosed in a channel. The details about the geometric sizing of the system are illustrated in Figure 4.5. A two-dimensional cylinder of a square shape of dimension D is enclosed inside a channel. The channel's dimensions are proportional to the cylinder's dimension by a linear scaling relationship. The channel's length (L) and height (H) are defined as $20D$

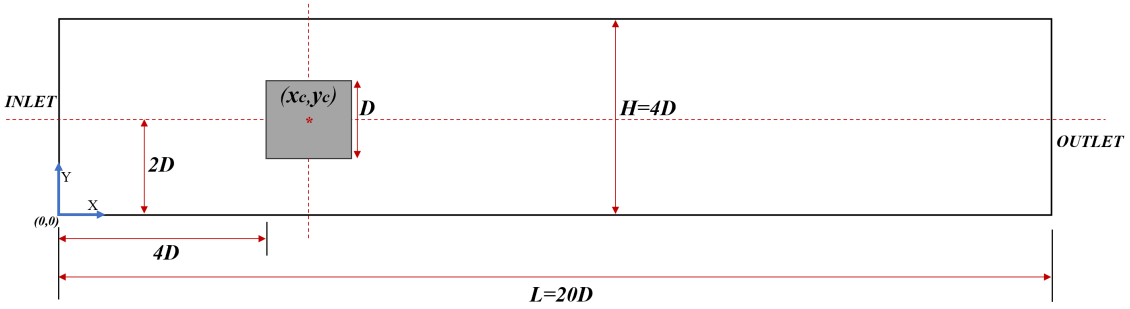


Figure 4.5 : Geometrical configuration of the problem.

and $4D$ respectively. The location of the square cylinder is symmetric w.r.t channel walls. The origin of the computational domain is located at the south-west corner of the channel. The coordinates of the cylindrical center with respect to the origin is located at: $(x_c = \frac{9D}{2}, y_c = 2D)$. The blockage ratio, H/D is equal to 4. An incompressible viscous fluid of $\rho = 1$ enters the channel with a uniform velocity and a relatively high temperature. The solid cylinder embedded in the channel is thermally conductive. As a result of flow over the cylinder, transfer of heat and temperature distribution occurs in the system. We want to investigate the heat conduction inside the square cylinder in addition to the convection of heat in the fluid which makes this study a CHT problem.

- Boundary conditions for flow:
  - The walls of the channel are modeled as no-slip surfaces ($u = v = 0$).
  - Uniform velocity at the inlet ($u = 0.1, v = 0$)
  - The channel outlet has constant density, $\rho = 0.1$.
  - The solid-fluid interfaces are modeled as no-slip walls ($u = v = 0$).
- Initial and boundary conditions for heat transfer:
  - Initial cylinder temperature, $T = 0$.
  - The walls of the channel are modeled as adiabatic along y-axis ($\frac{dT}{dy} = 0$).
  - Constant temperature at the inlet ($T = 1.0$).
  - The outlet is adiabatic in x-direction ($\frac{dT}{dx} = 0$).
- Flow and thermophysical properties:
  - Reynolds number, *Re*: 50, 100, 150 and 200.
  - Prandtl number, *Pr*: 0.71, 1, 5 and 10.
  - Solid to fluid conductivity ratio, k_r: 1, 5, 10 and 50.

The numerical algorithm employed in our LBM solver for solving the conjugate heat transfer problem is illustrated as a flowchart in Figure 4.6. The flowchart depicts

the execution of sequential steps of the dual solver till desired simulation time. The flow and heat transfer solvers are integrated into a single CUDA file for streamlined execution. The flow solver calculates the macroscopic density and velocity of fluid while the heat solver calculates the temperature.

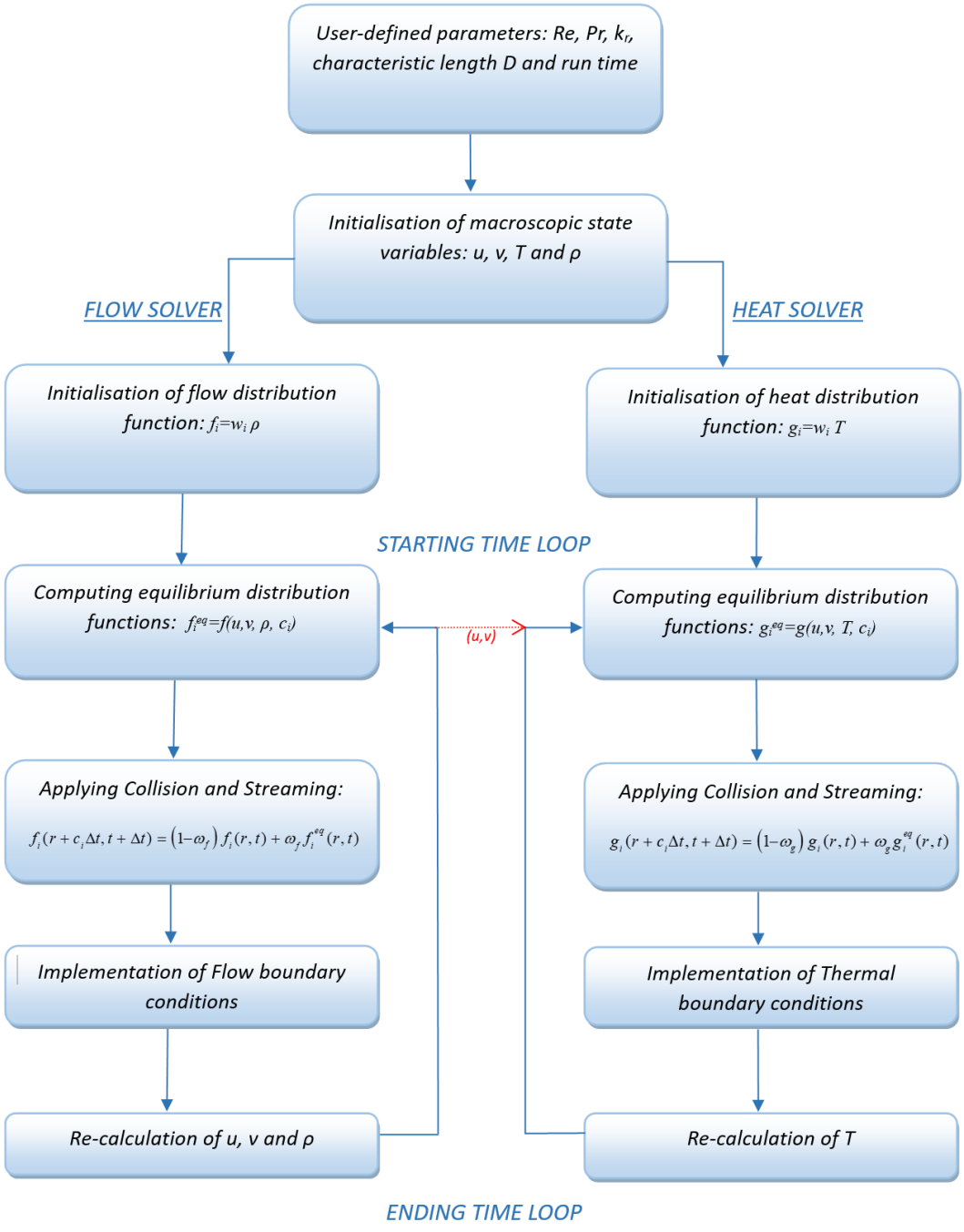


Figure 4.6 : Numerical algorithm of LBM for conjugate heat transfer.

4.3 Mesh Study

Mesh study in CFD and CHT simulations refers to the process of systematically varying the mesh or grid resolution to assess its impact on simulation results. It is a crucial step to check the accuracy of simulations and a test for reliability of results alongside validation. It allows us to evaluate the sensitivity of the simulation results with the change in mesh resolution and ensures that the simulations are accurate. In this study, conducting a mesh study is important as it directly influences credibility and authenticity of the results of our LBM solver. The mesh resolution directly affects the spatial discretisation of the domain. It determines the amount of finite nodes (lattice sites) that form the computational domain. Generally, a finer mesh captures more intricate flow features and temperature gradients but it comes with an increase in computational cost and time. Conversely, a coarser mesh reduces computational cost and time but it may lead to a loss of detail and accuracy in the results. Therefore, the mesh size at which the results become independent is chosen for accuracy as well as efficiency.

To systematically investigate the influence of changing the mesh resolution on the results of the simulation, we adopted a structured approach. The study of mesh resolution was carried out for $Re = 150$, $k_r = 1$ and $Pr = 1$. The size of the cartesian mesh is determined by the number of lattice sites per unit characteristic length (D) of the cylinder. The domain's height and length are linearly scaled. The length is taken as $20D$ and the height as $4D$ as was already discussed in the problem definition. We initialise the grid resolution from a coarser mesh where the number of lattice sites per unit characteristic length D is 64. Subsequently, the mesh resolution is increased by doubling the nodes per unit D for each mesh resolution. This progressive refinement continues till the mesh resolution becomes finer. The mesh study is illustrated in Figure 4.7 and the summary of results for each mesh resolution is shown in Table 4.1. For each mesh resolution, the temperature distribution along the faces of the square cylinder is plotted as shown in figure. We observe a decreasing trend in the local temperature values with increase in the mesh resolution. The local temperature values

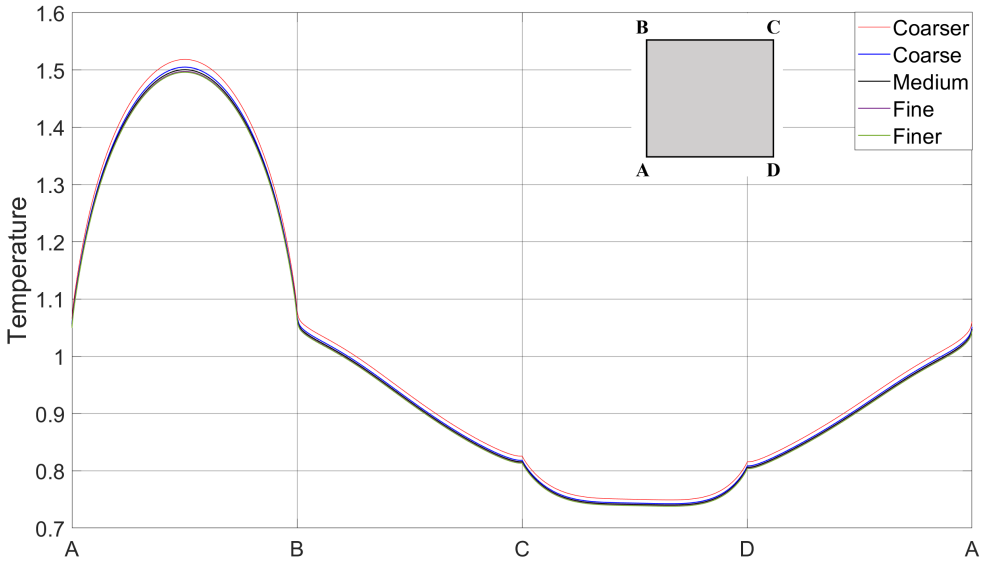


Figure 4.7 : Variation of local temperature over the cylinder for different mesh resolutions

gradually converge for $D \geq 512$ indicating independence. Based on the outcomes of the mesh study, the mesh resolution of 512 lattice sites per unit D is chosen as the optimal mesh size for achieving the desired balance between computational efficiency and result accuracy. This optimal mesh resolution ensures that our simulations produce reliable and consistent temperature distributions at a limited computational cost. By conducting this mesh study and selecting an appropriate mesh resolution, the credibility and robustness of our computational model are improved. This facilitates accurate prediction of the thermal behavior of our conjugate system which is the subject of discussion in the forthcoming sections and chapters.

Table 4.1 : Summary of mesh study.

Mesh size description	No. of lattice sites per D	Total lattice sites	% Deviation
Coarser	64	1280×256	1.07×10^{-1}
Coarse	128	2560×512	2.01×10^{-3}
Medium	256	5120×1024	6.70×10^{-5}
Fine	512	10240×2048	7.71×10^{-7}
Finer	1024	20480×4096	8.71×10^{-7}

4.4 Flow Solver Results

The post-processed results of velocity and vorticity are obtained in the form of

instantaneous contours.We discuss each of them followed by illustration of time periodic behaviour of flow in this section.

4.4.1 Velocity contours

The evolution of flow behavior with varying Reynolds number (*Re*) is clearly illustrated in the velocity contours shown in Figure 4.8. The figure presents four instantaneous contours, each corresponding to a different Reynolds number, demonstrating how the flow characteristics change with *Re*. To understand the significance of the limiting (critical) value of *Re* on the flow, it is important to observe that the onset of unsteady periodic flow velocity occurs when $Re \geq 50$, marking the beginning of the unsteady periodic flow regime. For Reynolds numbers exceeding this critical threshold ($Re \geq 50$), we observe the onset of unsteady flow phenomena characterized by oscillations downstream of the cylinder within the channel, as depicted in Figure 4.8 (a). These oscillations are indicative of the beginning of the vortex shedding phenomenon, where vortices are alternately shed from the top and bottom of the trailing edge of the cylinder into the adjacent fluid medium.

As the Reynolds number (*Re*) increases further, Figure 4.8 (b), (c), and (d) show a significant intensification in the vortex shedding phenomenon. This increase is due to the higher momentum of the incoming flow, which results in more vigorous vortex formation and shedding. The frequency of shedding also rises with higher Reynolds numbers, with more vortex cycles occurring in a given time interval. Notably, the flow field remains consistent across all cases, regardless of changes in conductivity ratio and Prandtl number. This consistency indicates that the fundamental flow behavior, including vortex shedding and wake oscillations, is primarily influenced by variations in Reynolds number rather than other parameters, as was previously shown by others [64,65]. For steady flow, the findings for flow and heat transfer were discussed in the preliminary findings [66].

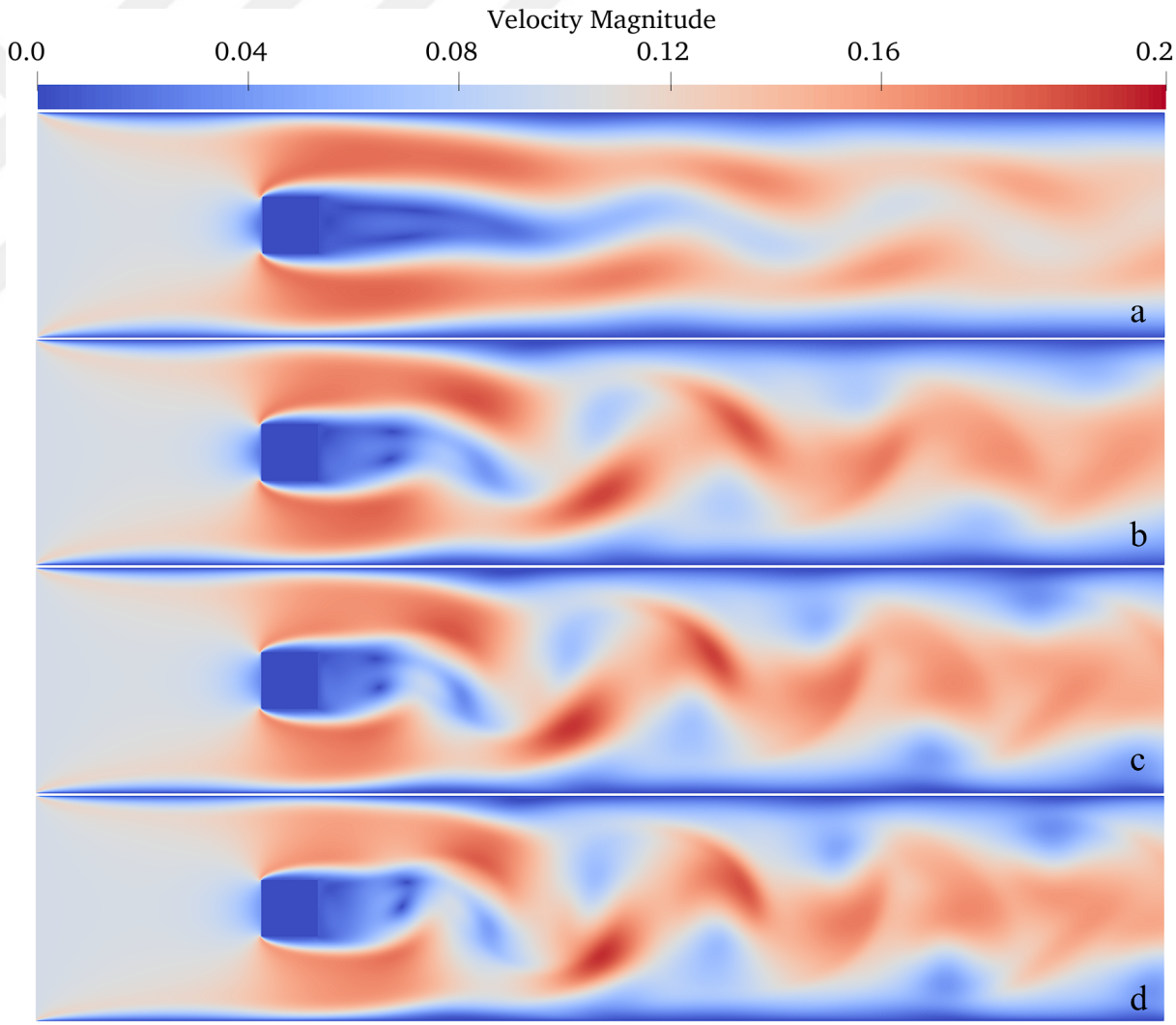


Figure 4.8 : Instantaneous contours of velocity for varying *Re*: a) 50 b) 100 c) 150 d) 200.

4.4.2 Vorticity contours

The distribution of vorticity levels in the domain is calculated from the flow velocity using the equation, $\omega = \nabla \times \mathbf{V}$ to study the behavior and interaction of shear layers. These shear layers originate from the walls of the channel and the square cylinder. Figure 4.9 demonstrates the instantaneous vorticity contours for the cylinder, revealing distinct shear layers emerging from the square cylinder. These shear layers exhibit a characteristic curling pattern, extending upward from the cylinder toward the channel walls near their formation region. This upward extension of the shear layers indicates the presence of intense vortical structures generated by the interaction between the fluid and the square cylinder.

As the Reynolds number (Re) increases, we observe a notable strengthening of the shear layers originating from the cylinder. This increase in the magnitude of the shear layers is accompanied by a corresponding enhancement in their influence on the adjacent channel walls. Consequently, the shear layers exert greater forces on the channel walls, leading to pronounced curling and separation of the shear layers from the wall surface. Variations in conductivity ratio (k_r) and Prandtl number (Pr) do not affect the distribution of vorticity within the computational domain.

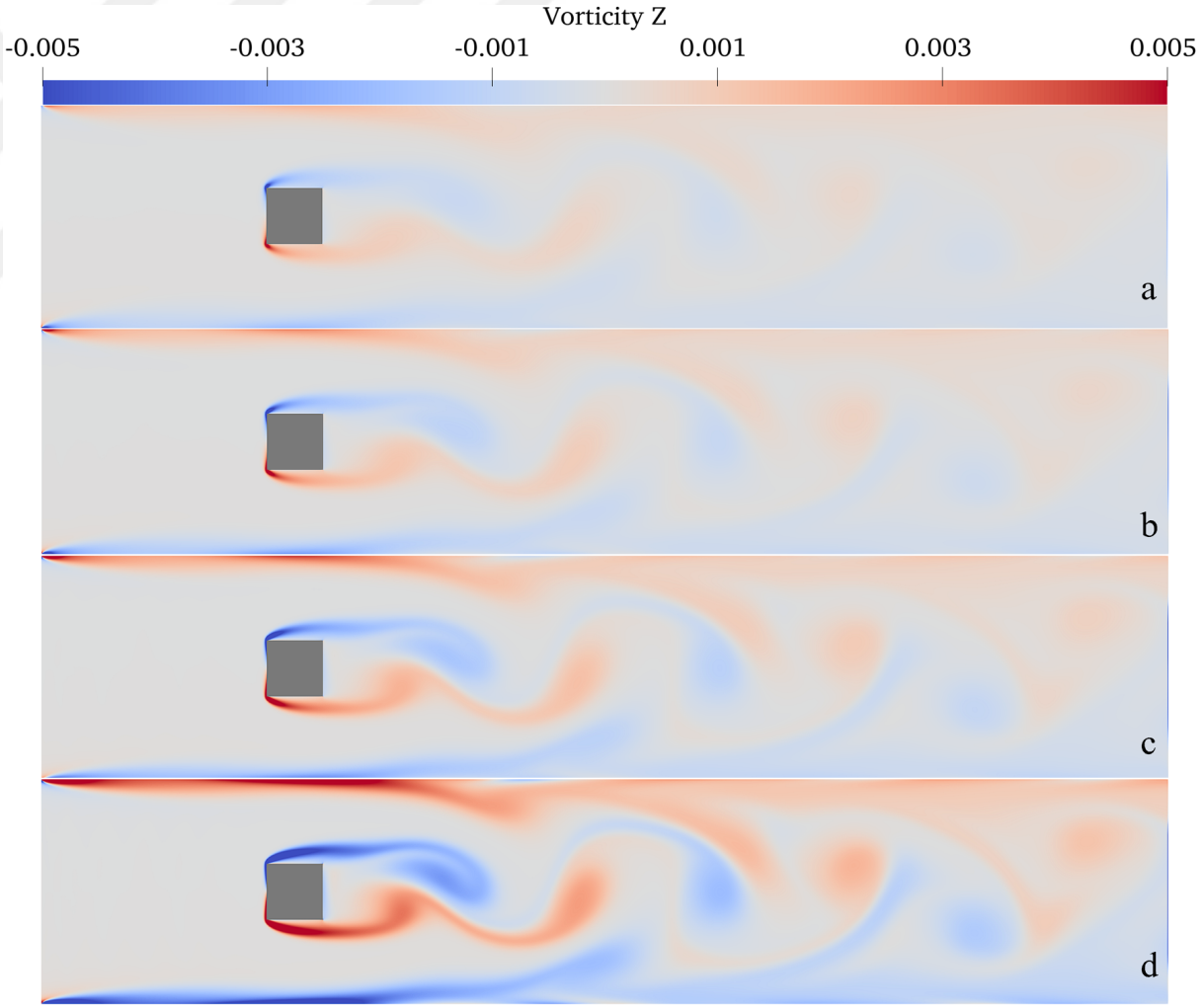


Figure 4.9 : Instantaneous contours of vorticity for varying *Re*: a) 50 b) 100 c) 150 d) 200.

4.4.3 Time periodicity of flow

The periodic nature of the flow was verified by examining the time histories of the drag (C_d) and lift (C_l) coefficients at $Re = 100$. The drag (D) and lift (L) forces acting on the cylinder were computed by considering the pressure and viscous contributions on each face of the cylinder. The net lift and drag coefficients were calculated from the components of viscous and pressure distributions on the cylinder's faces as $C_d = C_{dp} + C_{dv}$ and $C_l = C_{lp} + C_{lv}$. The variations in C_l, C_d, and their components over time followed a sinusoidal waveform, as illustrated in Figure 4.10. As expected, the frequency of C_l and its components matched the natural frequency of vortex shedding, while the frequency of C_d and its components was twice that of C_l. Figure 4.10 also shows that viscous contributions were limited, and pressure contributions were significant in both C_l and C_d.

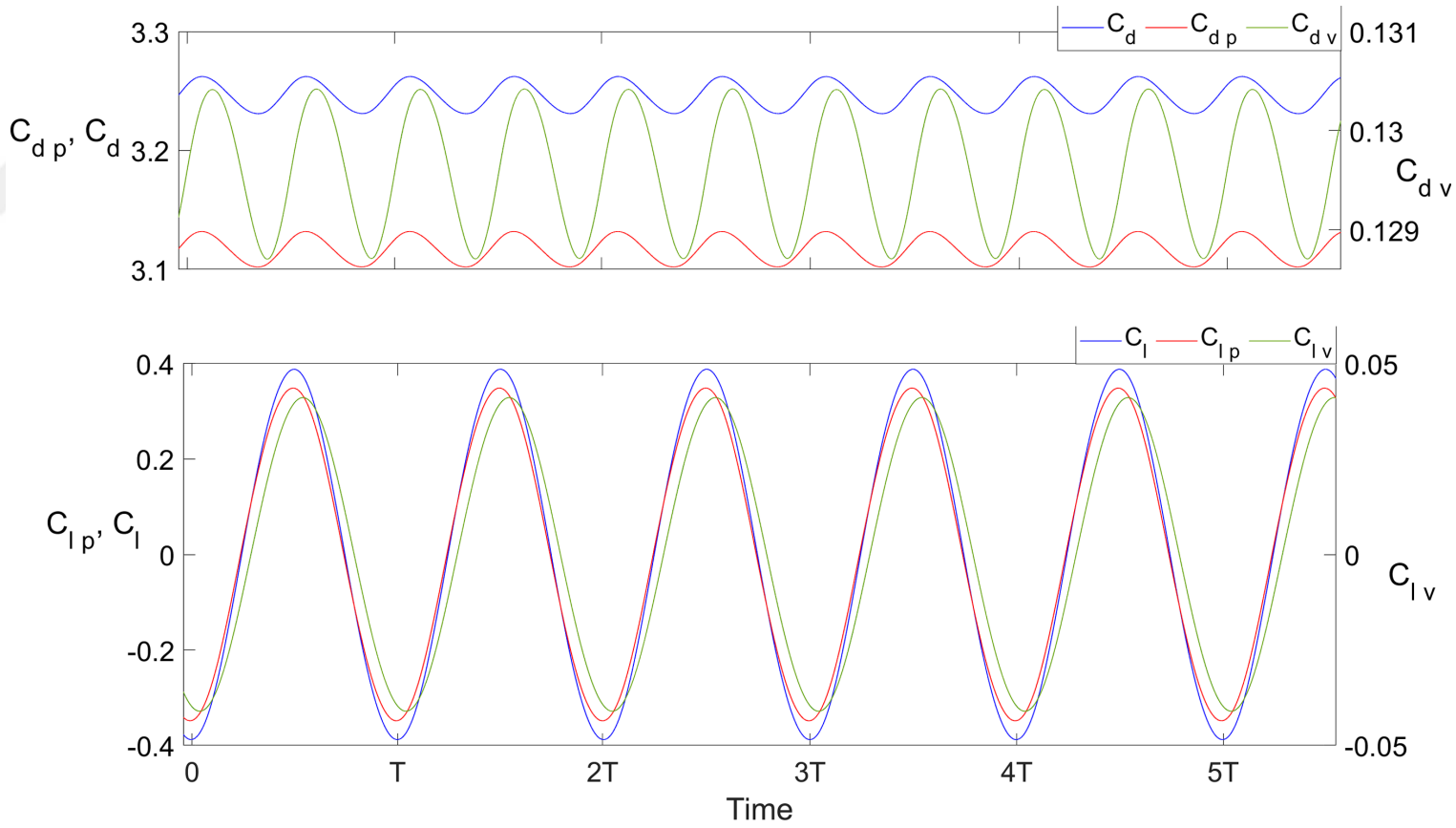


Figure 4.10 : Evolution of lift, drag coefficients and their pressure and viscous components with time.

4.5 Heat Solver Results

The temperature distribution contours are illustrated in Figures 4.11-4.26 for different values of Pr and k_r with varying Re. Unlike the flow field results, which vary only with Re, the heat transfer and temperature distribution for the conjugate system changes with Pr, k_r and Re.

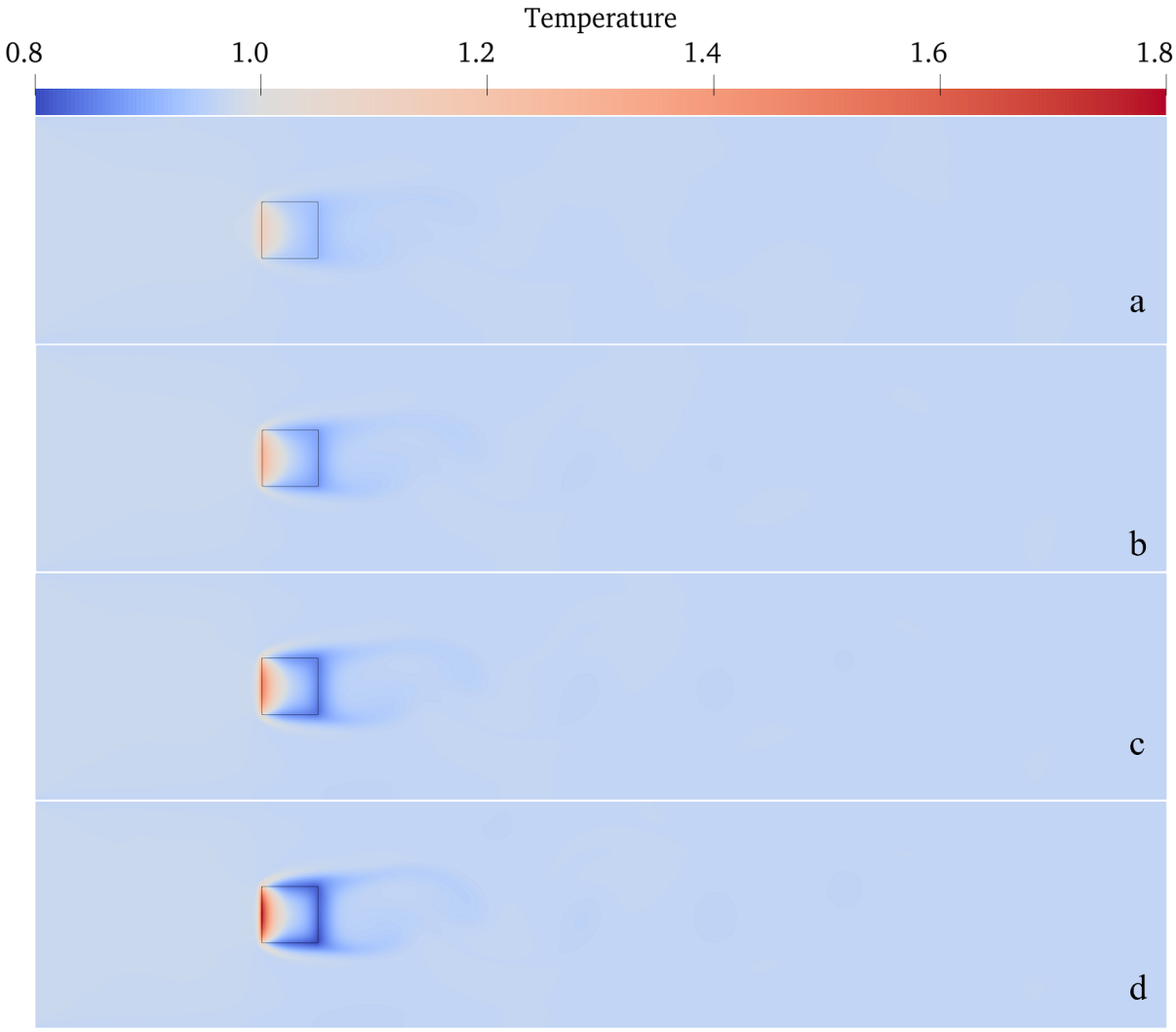


Figure 4.11 : Instantaneous contours of temperature distribution at $Pr = 0.71$ and $k_r = 1$ with varying Re: a) 50 b) 100 c) 150 d) 200.

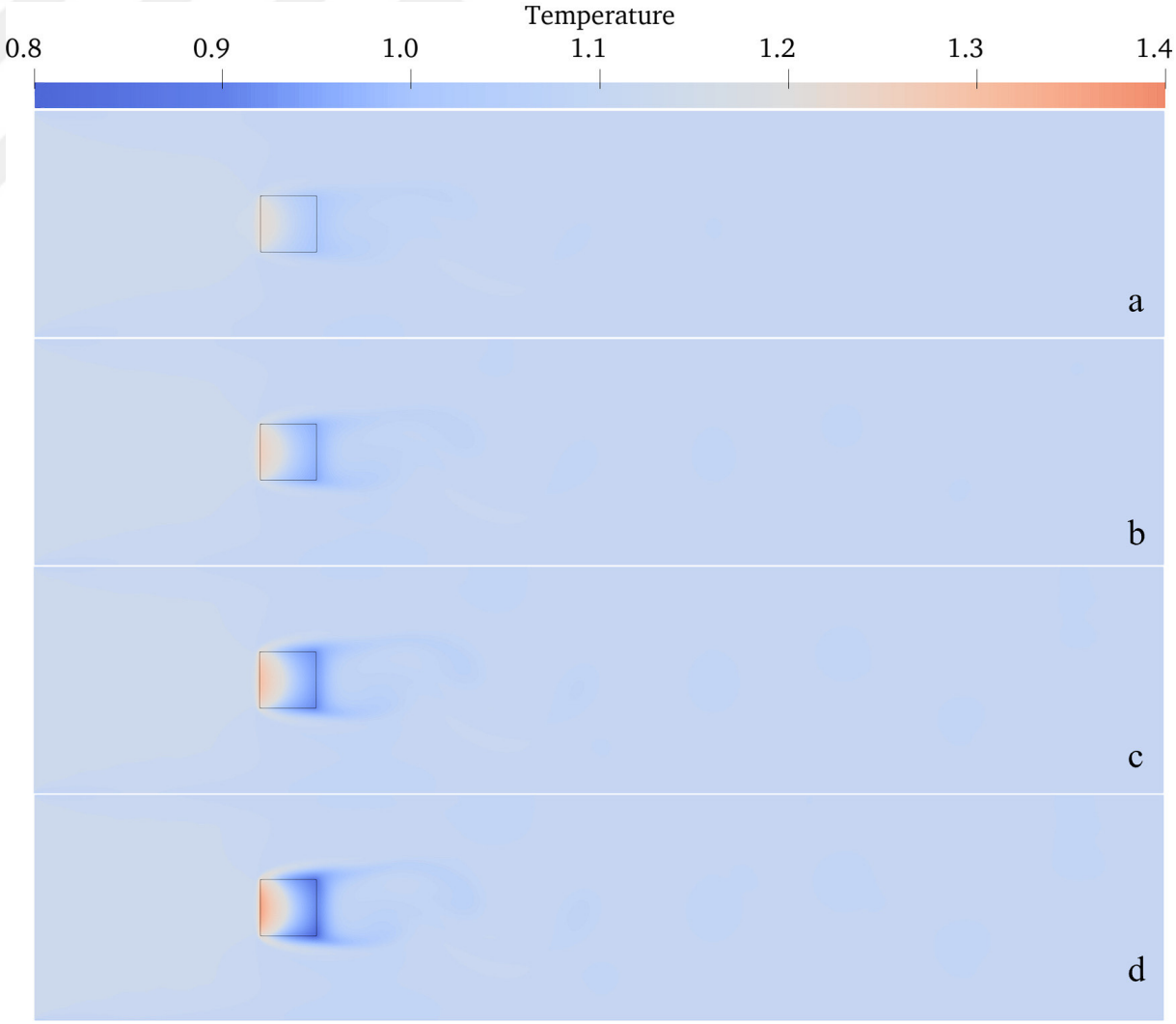


Figure 4.12 : Instantaneous contours of temperature distribution at $Pr = 0.71$ and $k_r = 5$ with varying Re: a) 50 b) 100 c) 150 d) 200.

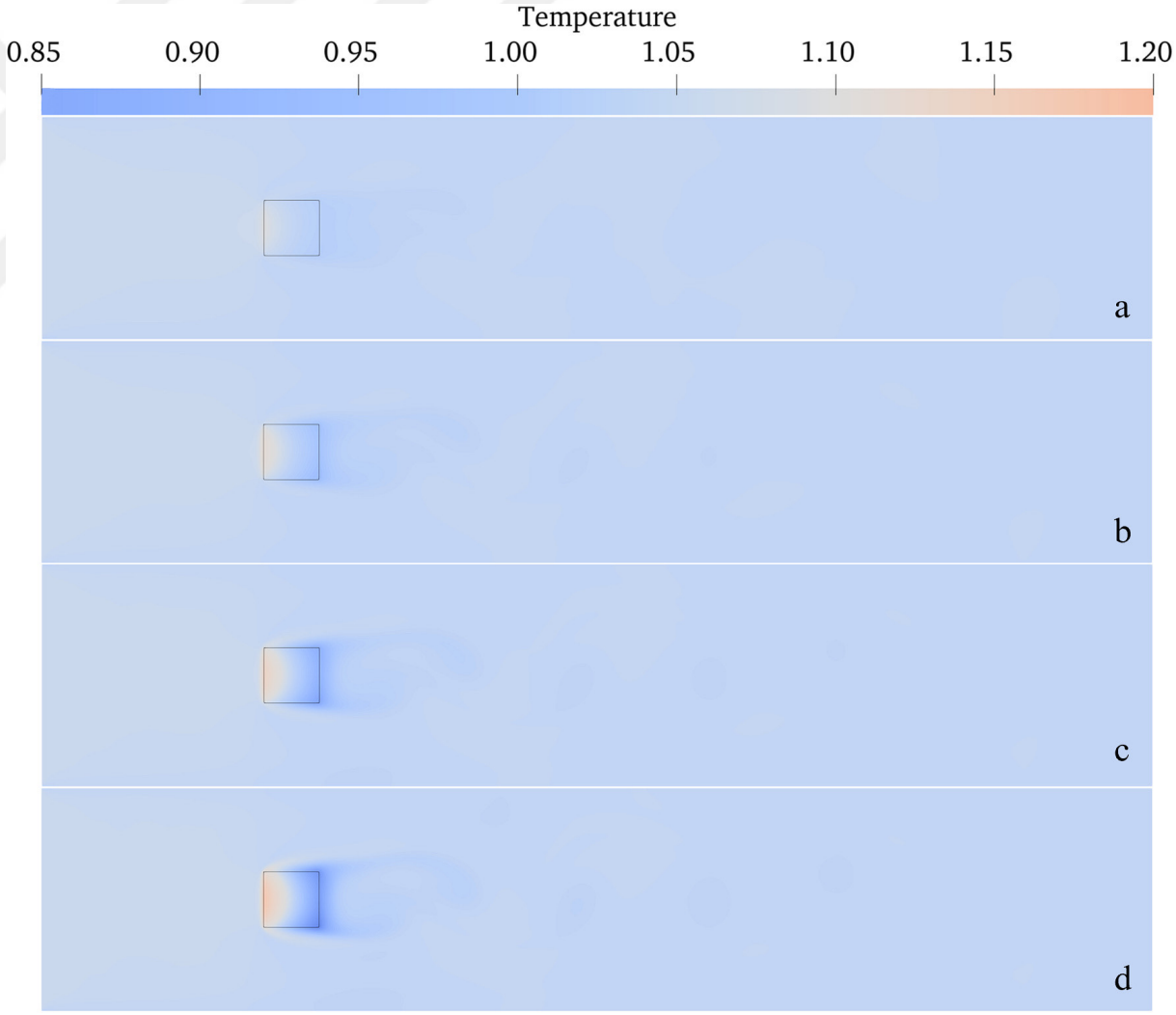


Figure 4.13 : Instantaneous contours of temperature distribution at $Pr = 0.71$ and $k_r = 10$ with varying Re: a) 50 b) 100 c) 150 d) 200.

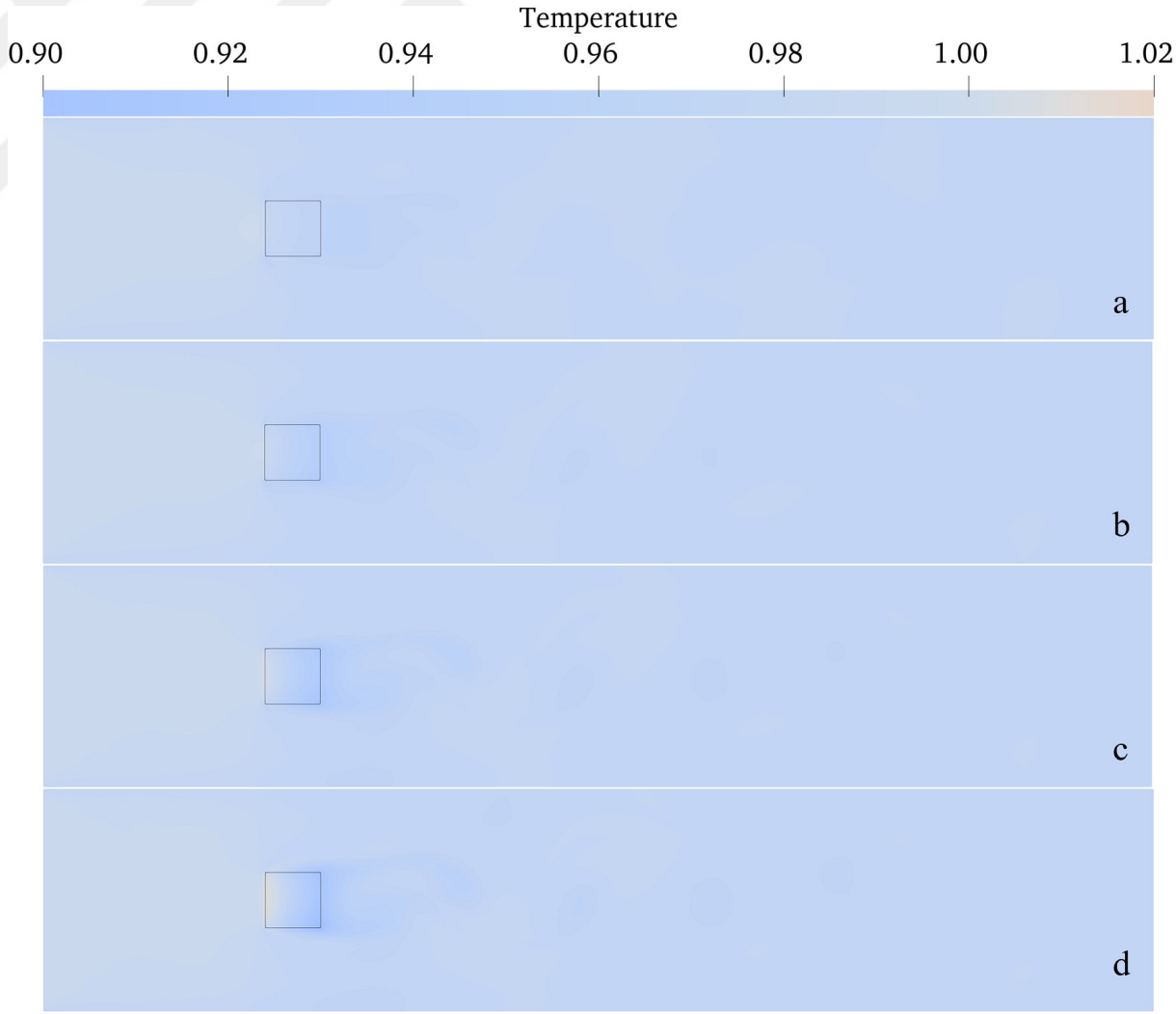


Figure 4.14 : Instantaneous contours of temperature distribution at $Pr = 0.71$ and $k_r = 50$ with varying Re: a) 50 b) 100 c) 150 d) 200.

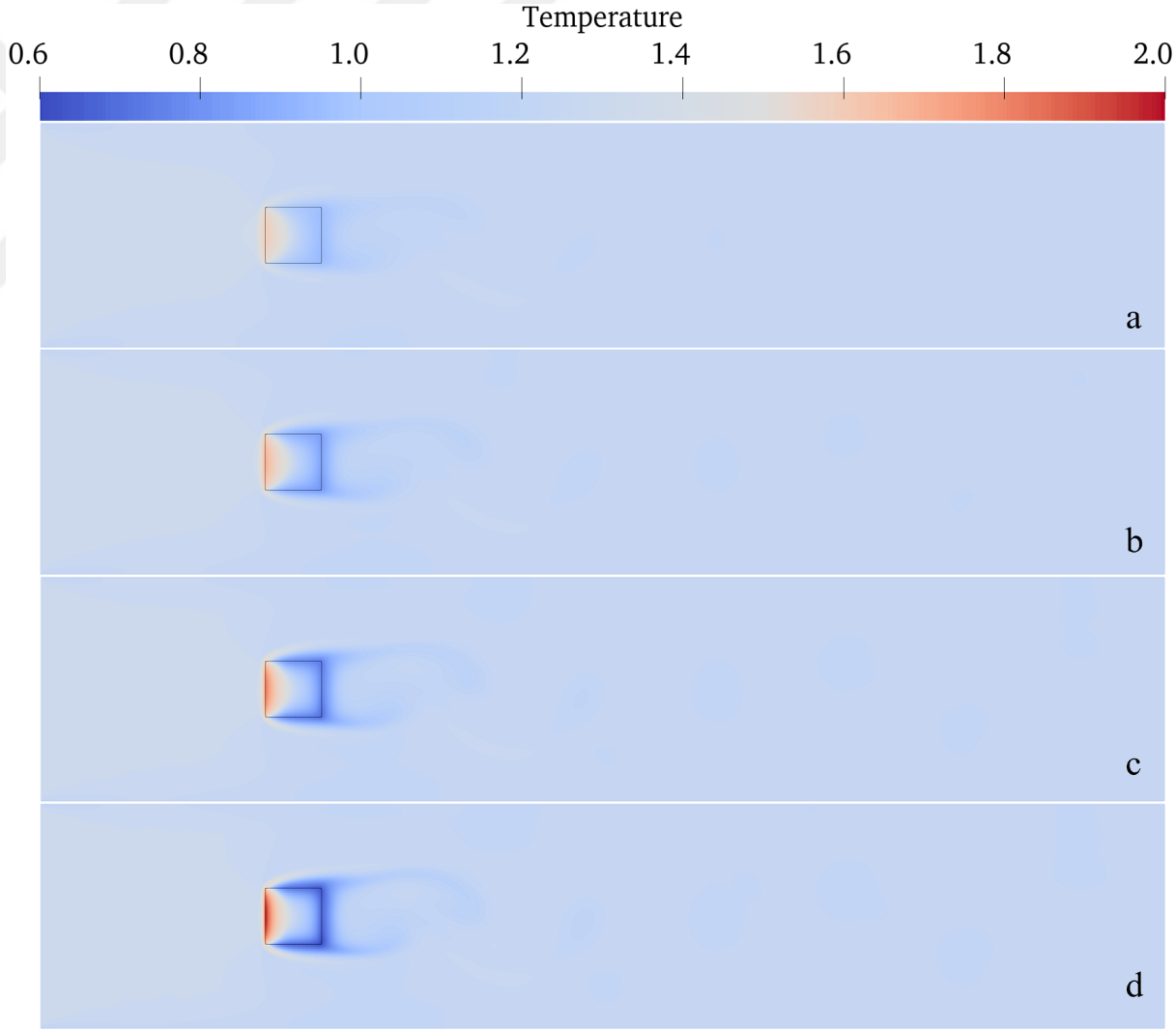


Figure 4.15 : Instantaneous contours of temperature distribution at $Pr = 1$ and $k_r = 1$ with varying Re: a) 50 b) 100 c) 150 d) 200.

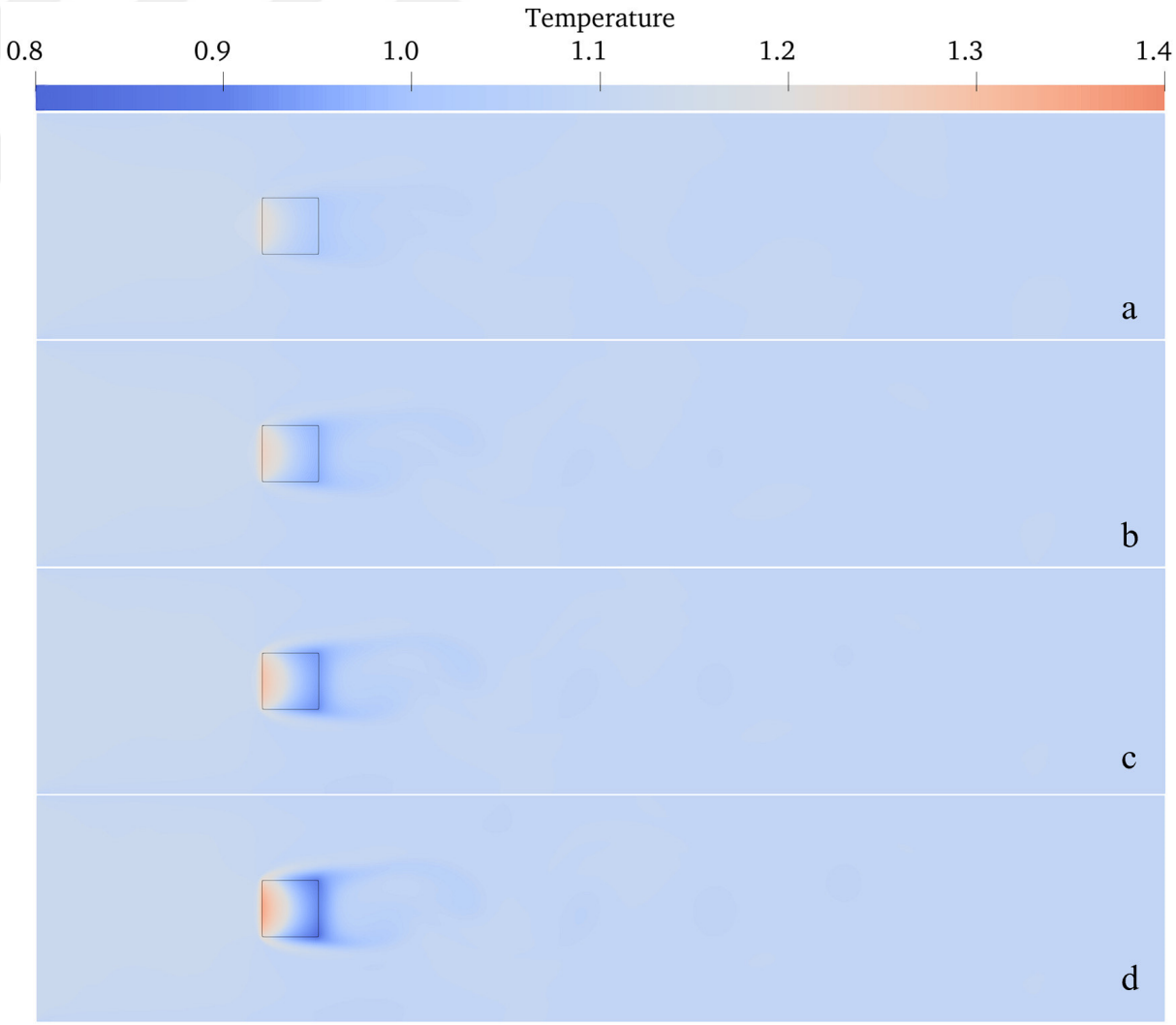


Figure 4.16 : Instantaneous contours of temperature distribution at $Pr = 1$ and $k_r = 5$ with varying Re: a) 50 b) 100 c) 150 d) 200.

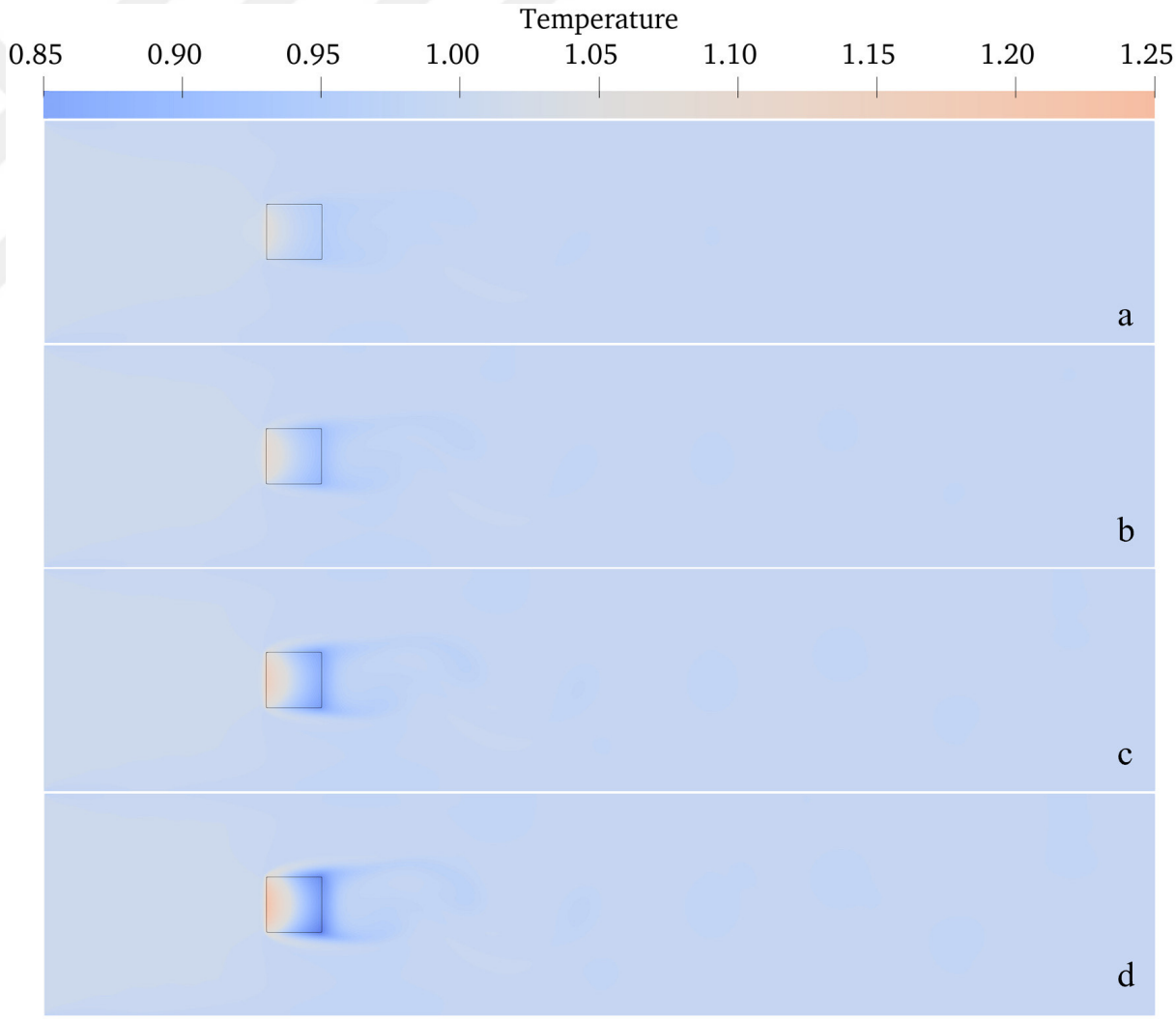


Figure 4.17 : Instantaneous contours of temperature distribution at $Pr = 1$ and $k_r = 10$ with varying Re: a) 50 b) 100 c) 150 d) 200.

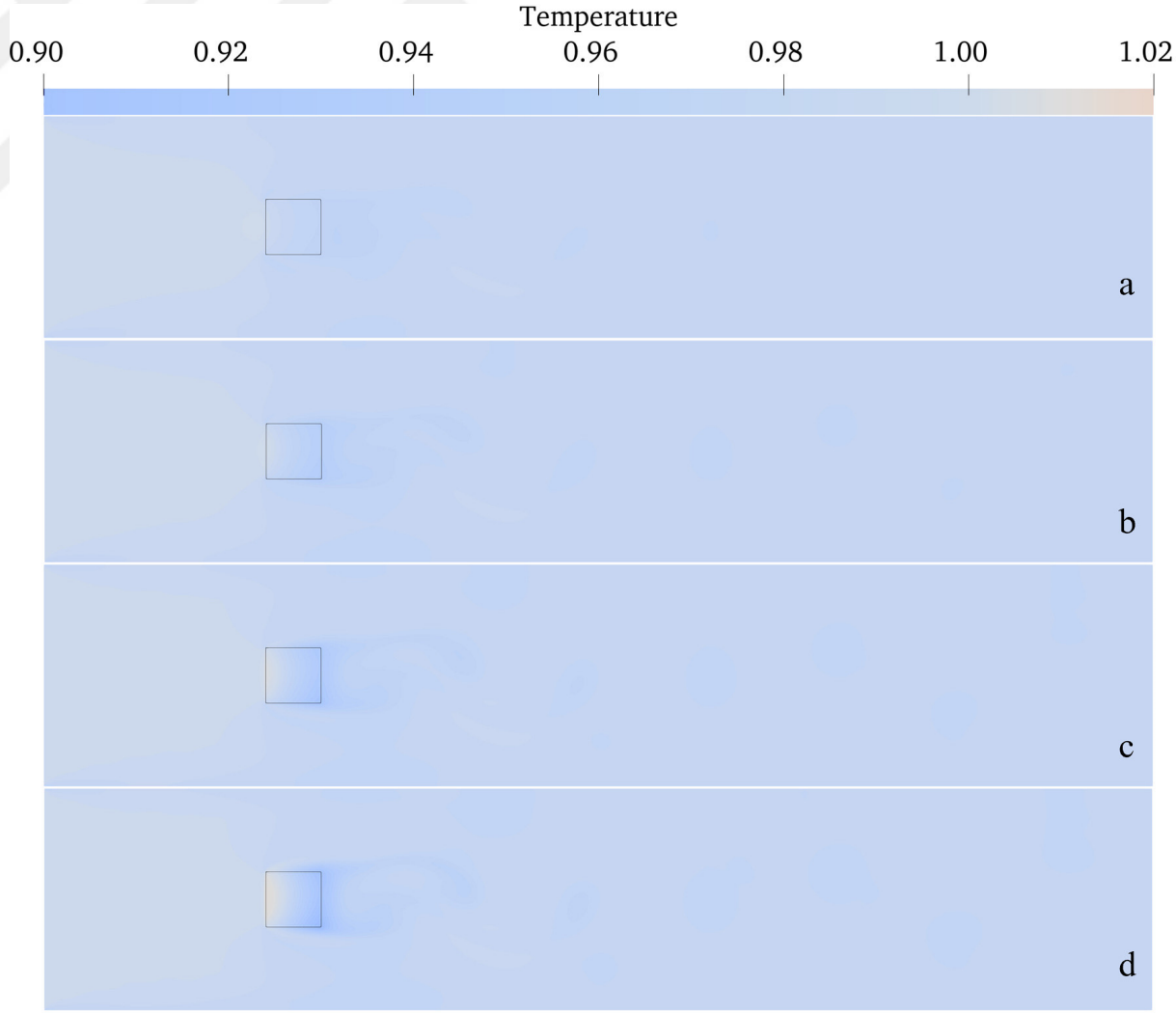


Figure 4.18 : Instantaneous contours of temperature distribution at $Pr = 1$ and $k_r = 50$ with varying Re: a) 50 b) 100 c) 150 d) 200.

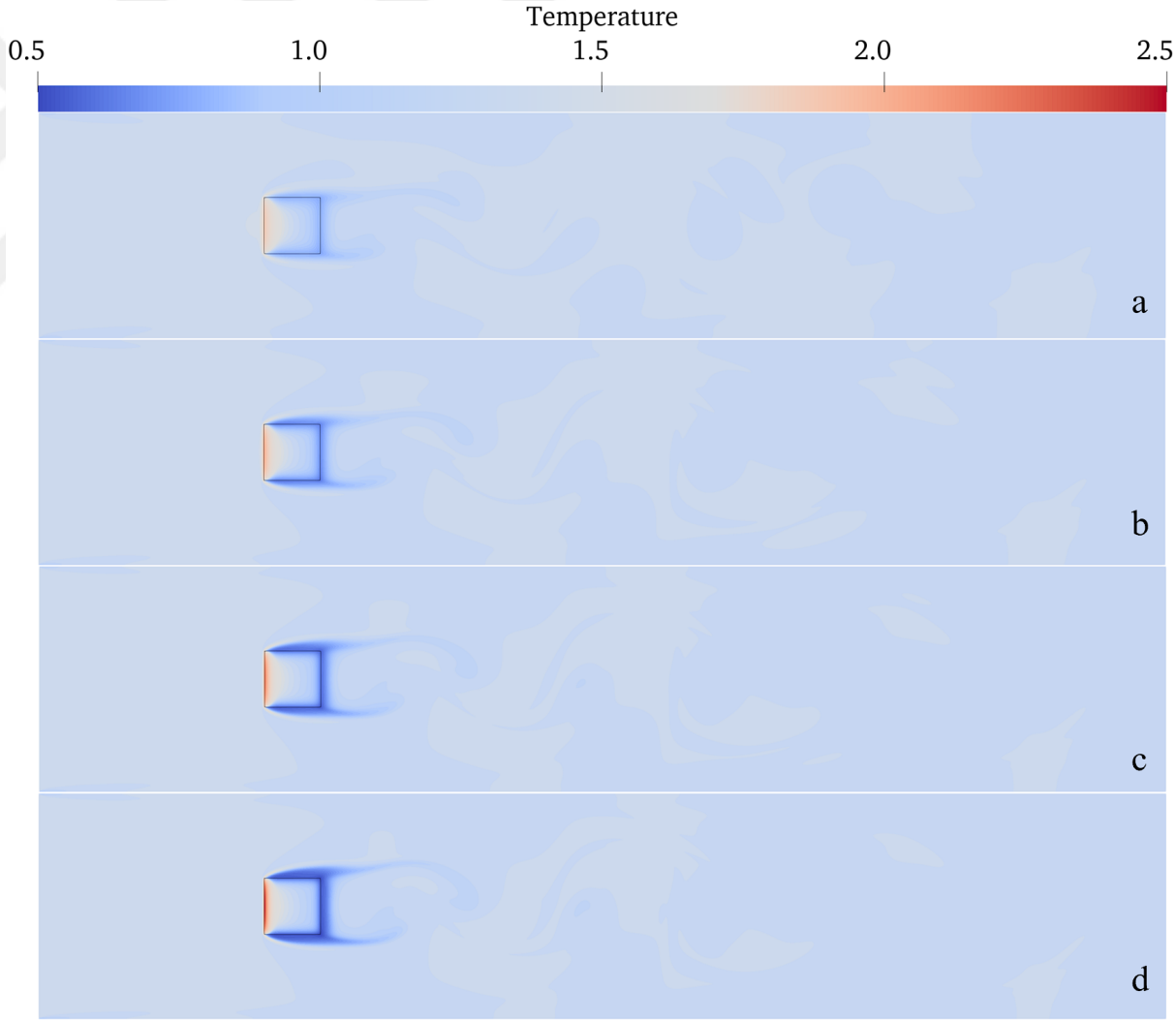


Figure 4.19 : Instantaneous contours of temperature distribution at $Pr = 5$ and $k_r = 1$ with varying Re: a) 50 b) 100 c) 150 d) 200.

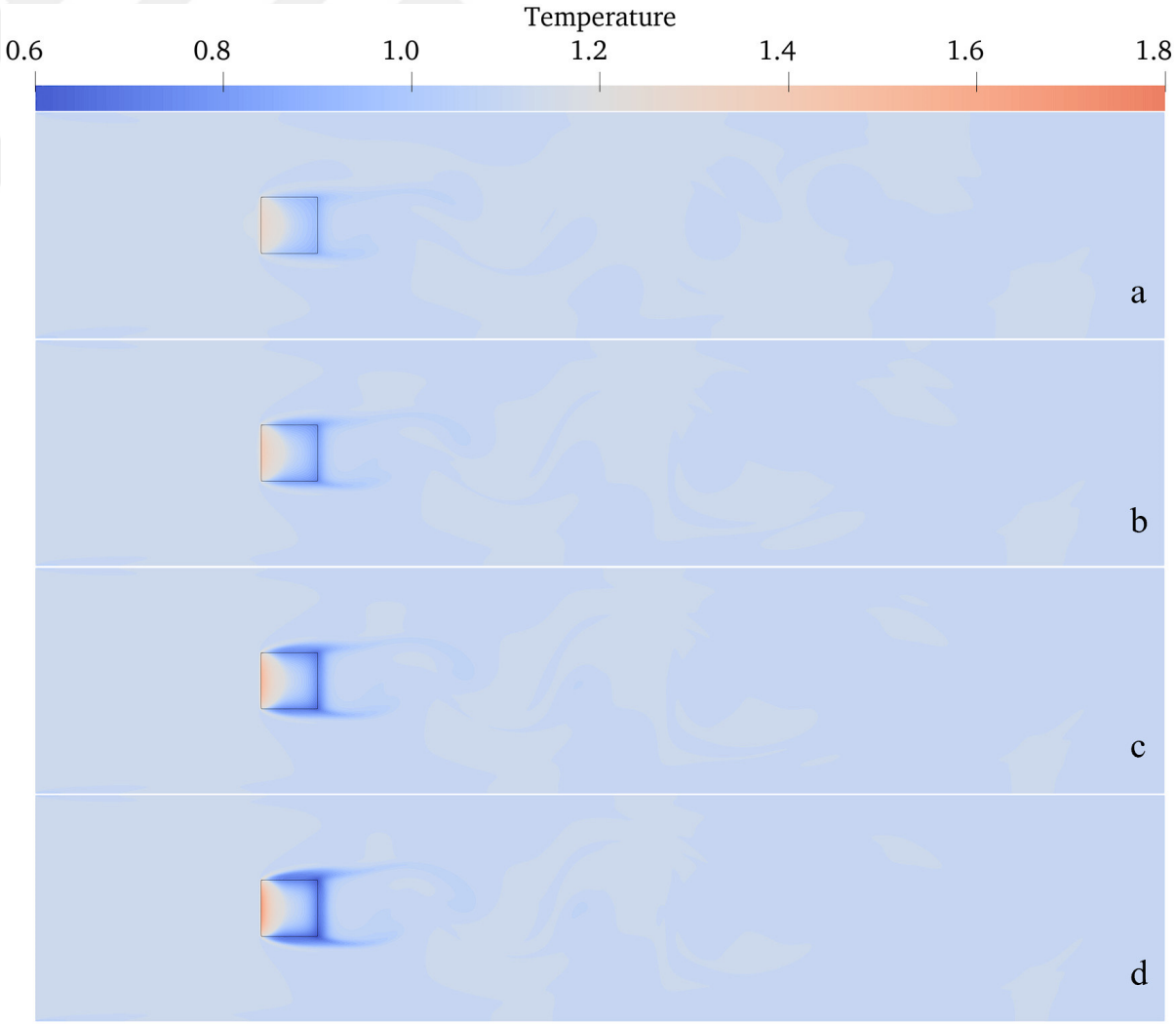


Figure 4.20 : Instantaneous contours of temperature distribution at $Pr = 5$ and $k_r = 5$ with varying Re: a) 50 b) 100 c) 150 d) 200.

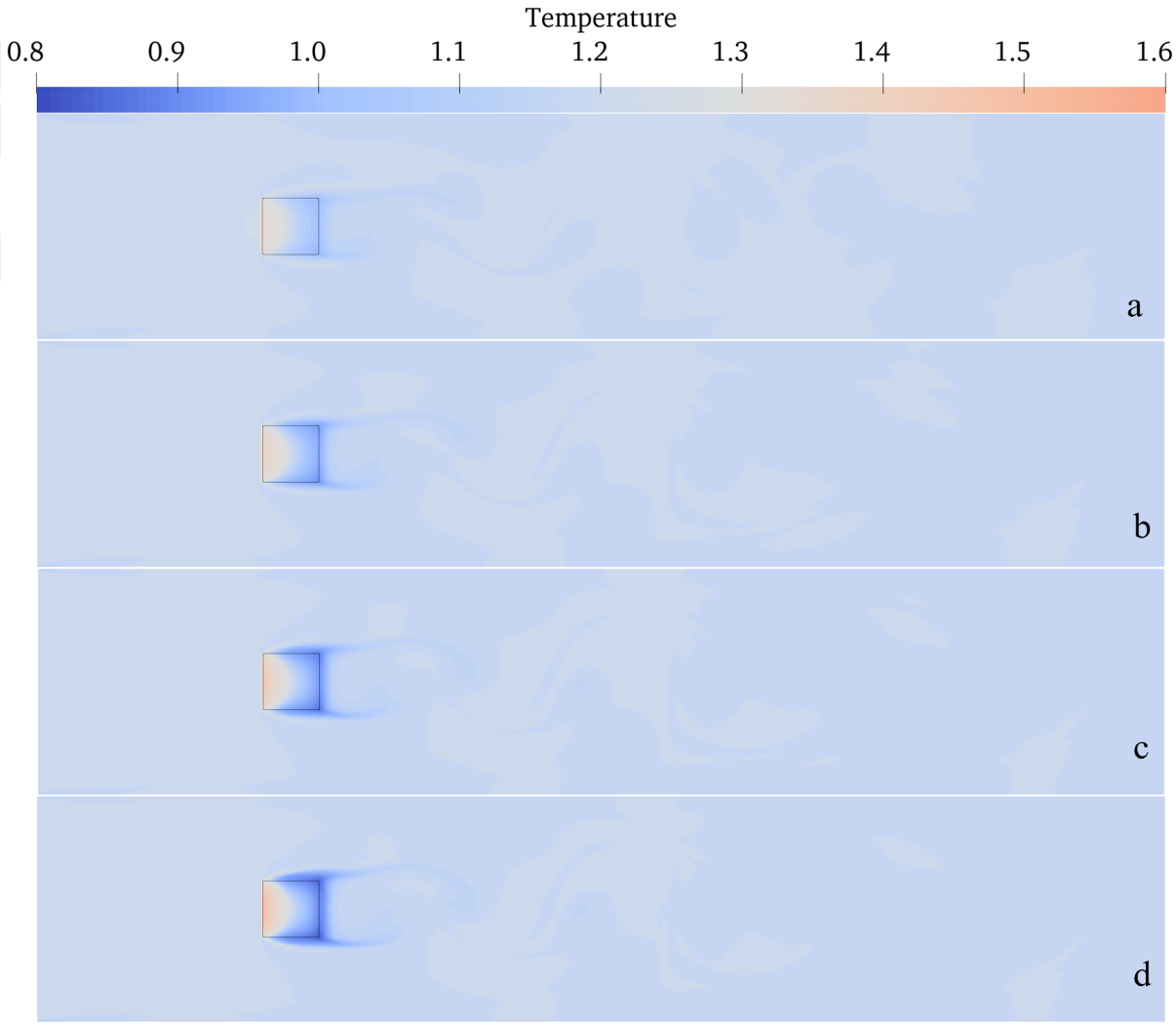


Figure 4.21 : Instantaneous contours of temperature distribution at $Pr = 5$ and $k_r = 10$ with varying Re: a) 50 b) 100 c) 150 d) 200.

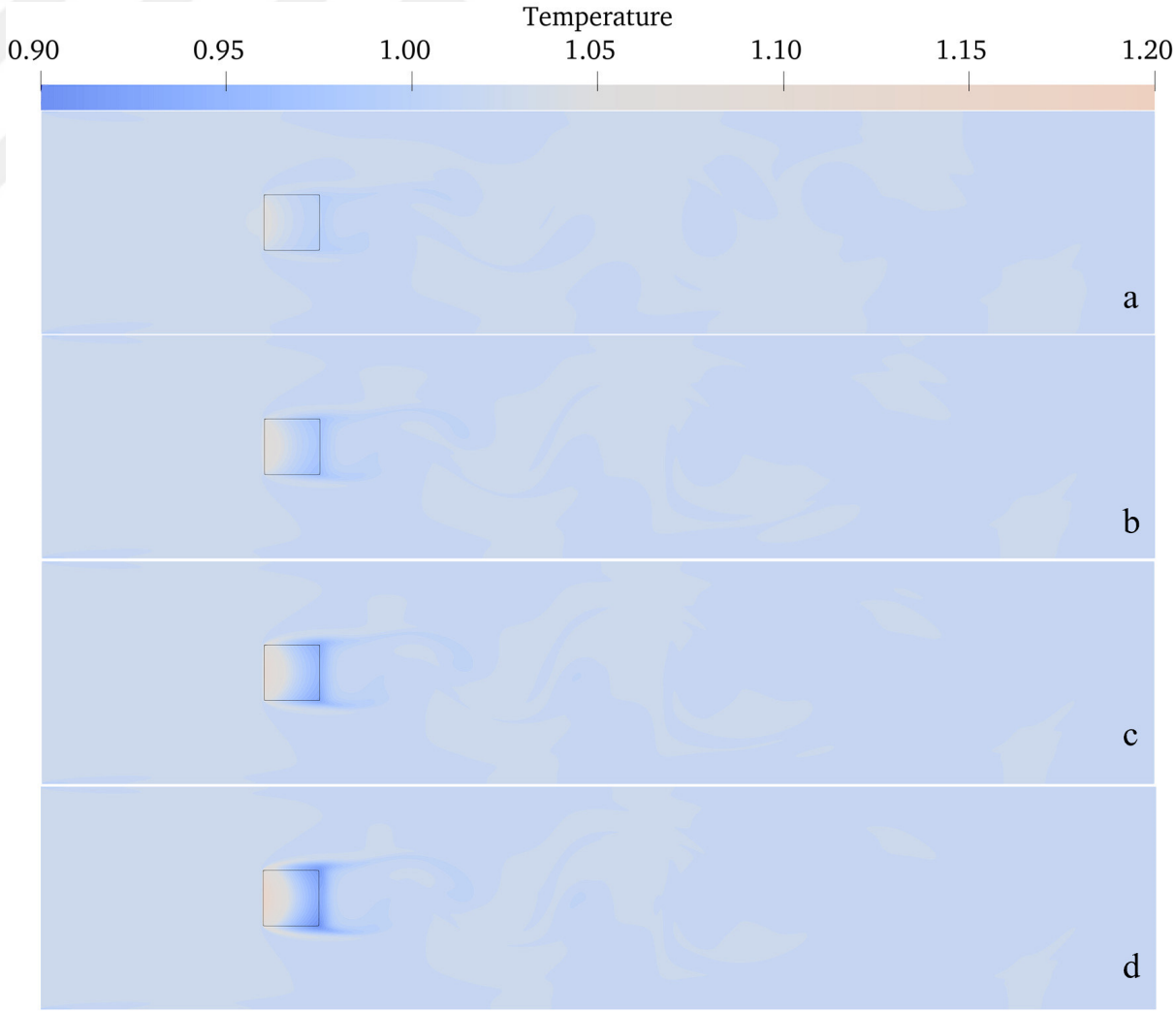


Figure 4.22 : Instantaneous contours of temperature distribution at $Pr = 5$ and $k_r = 50$ with varying Re: a) 50 b) 100 c) 150 d) 200.

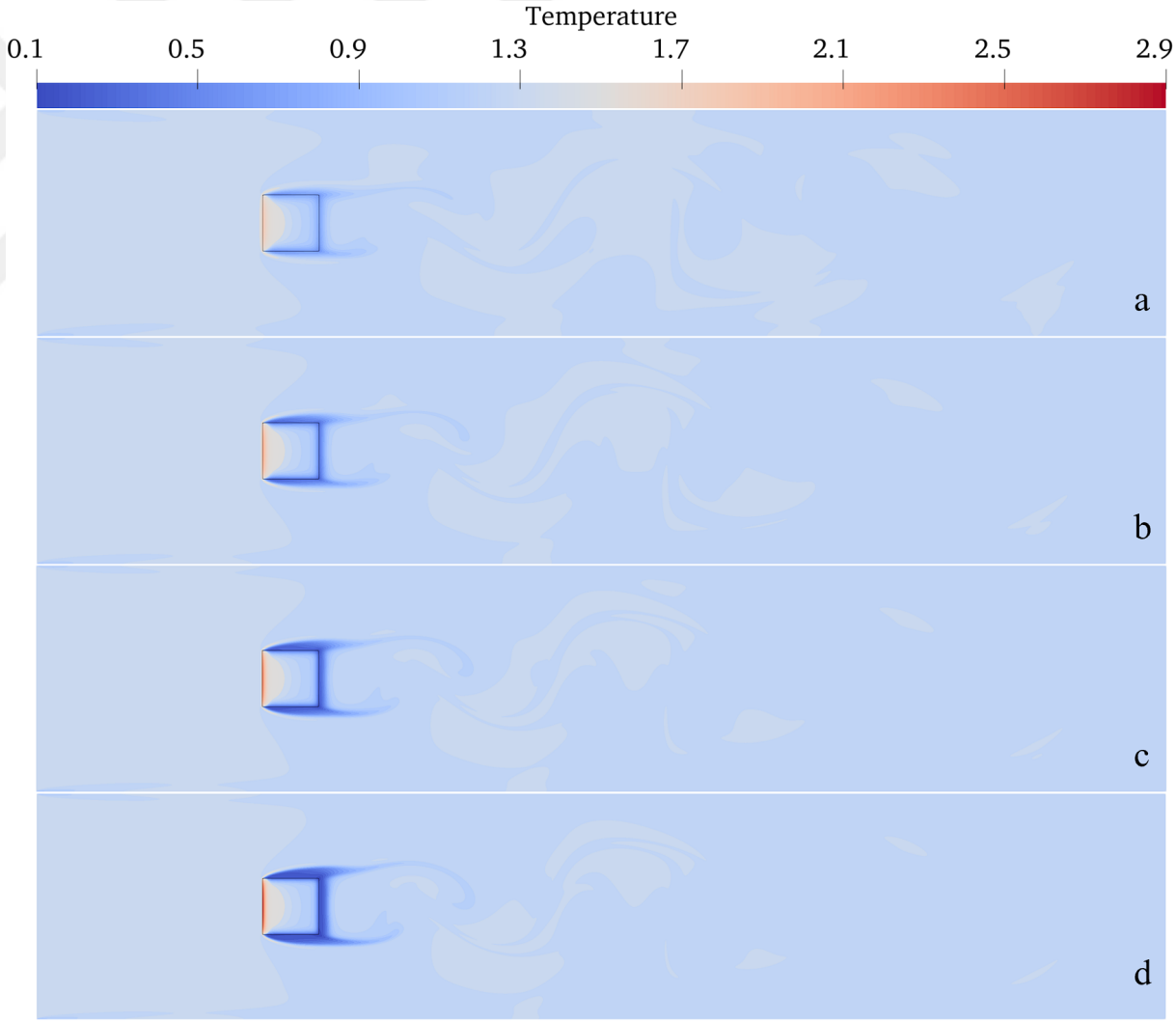


Figure 4.23 : Instantaneous contours of temperature distribution at $Pr = 10$ and $k_r = 1$ with varying Re: a) 50 b) 100 c) 150 d) 200.

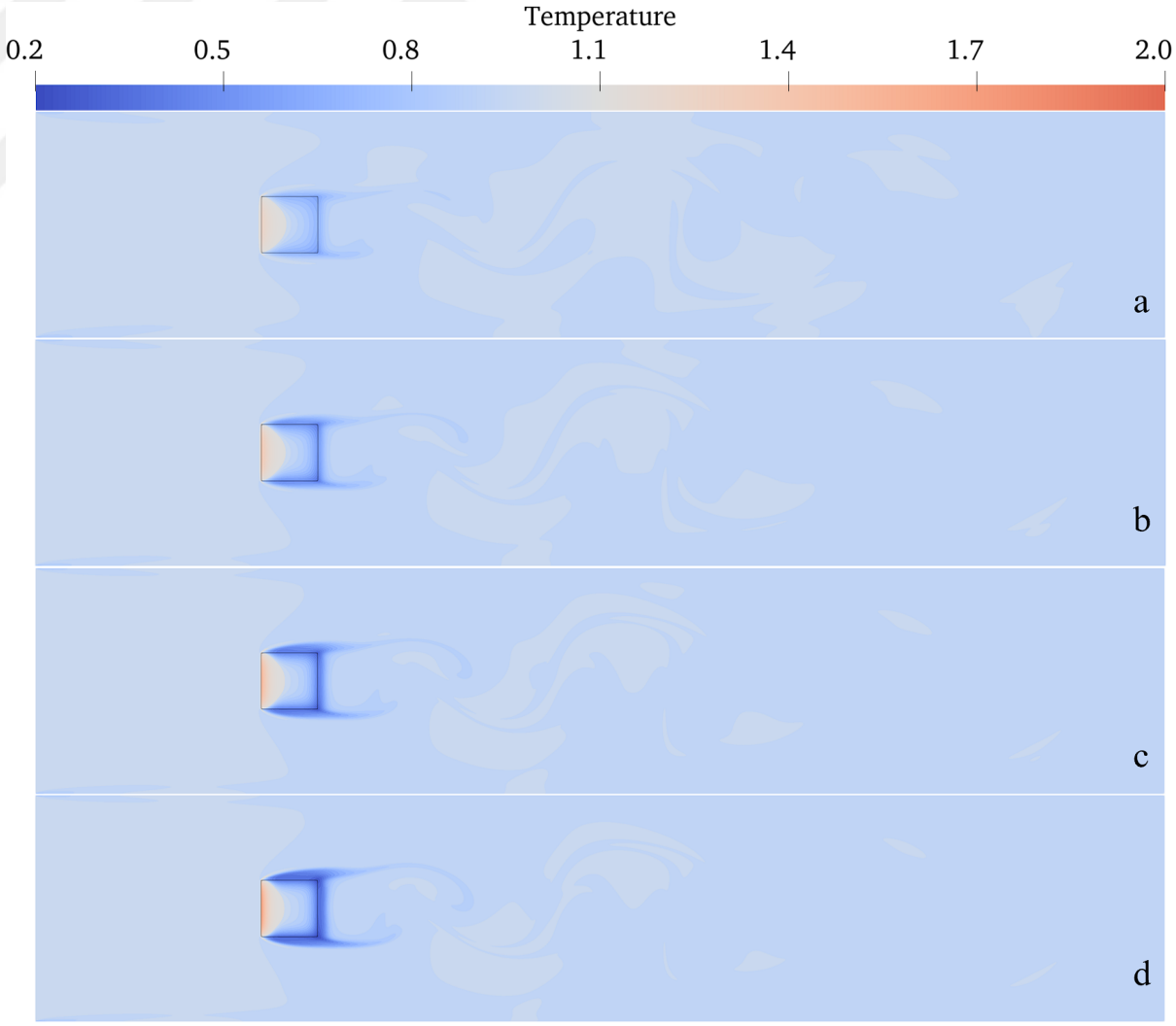


Figure 4.24 : Instantaneous contours of temperature distribution at $Pr = 10$ and $k_r = 5$ with varying Re: a) 50 b) 100 c) 150 d) 200.

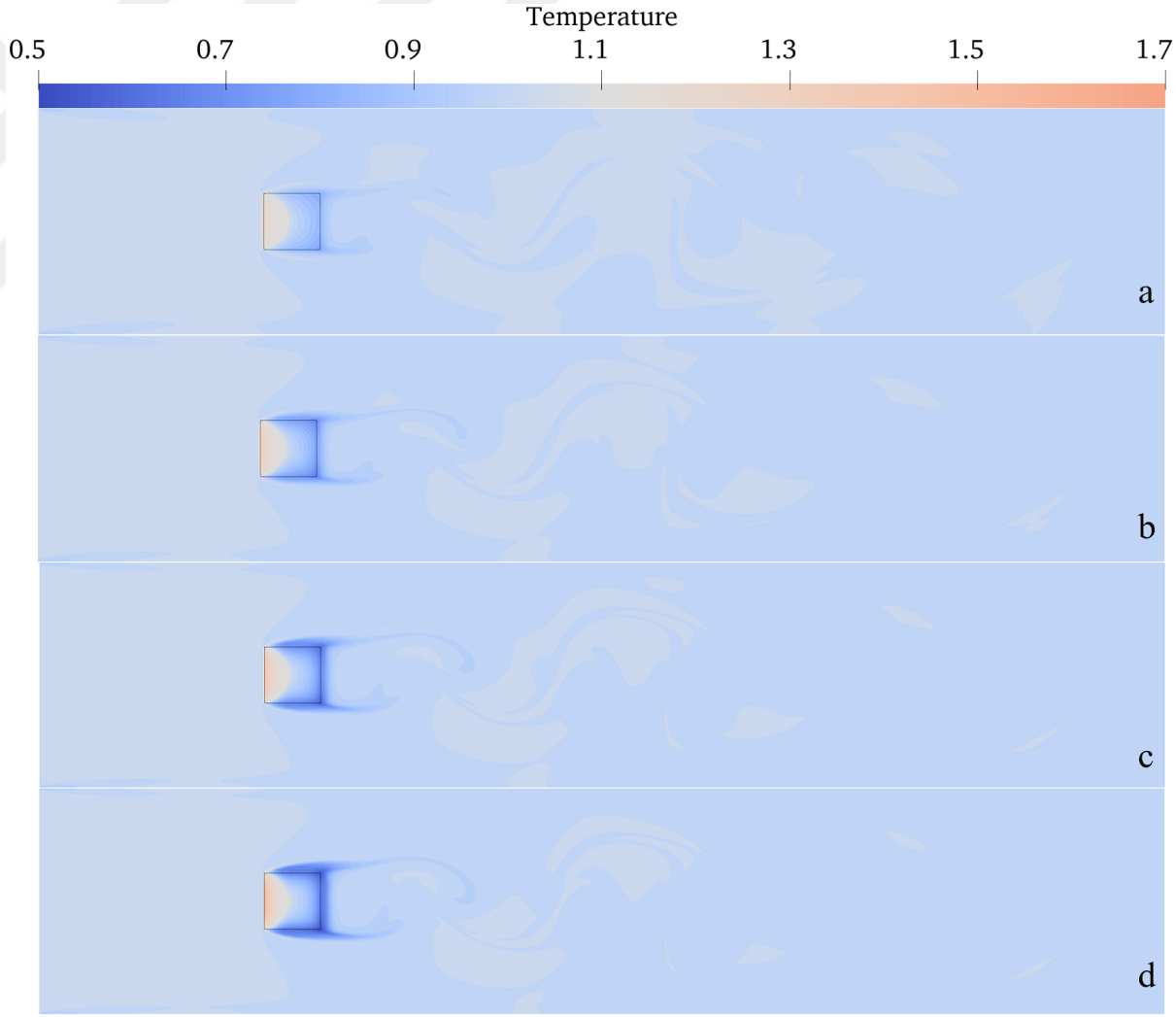


Figure 4.25 : Instantaneous contours of temperature distribution at $Pr = 10$ and $k_r = 10$ with varying Re: a) 50 b) 100 c) 150 d) 200.

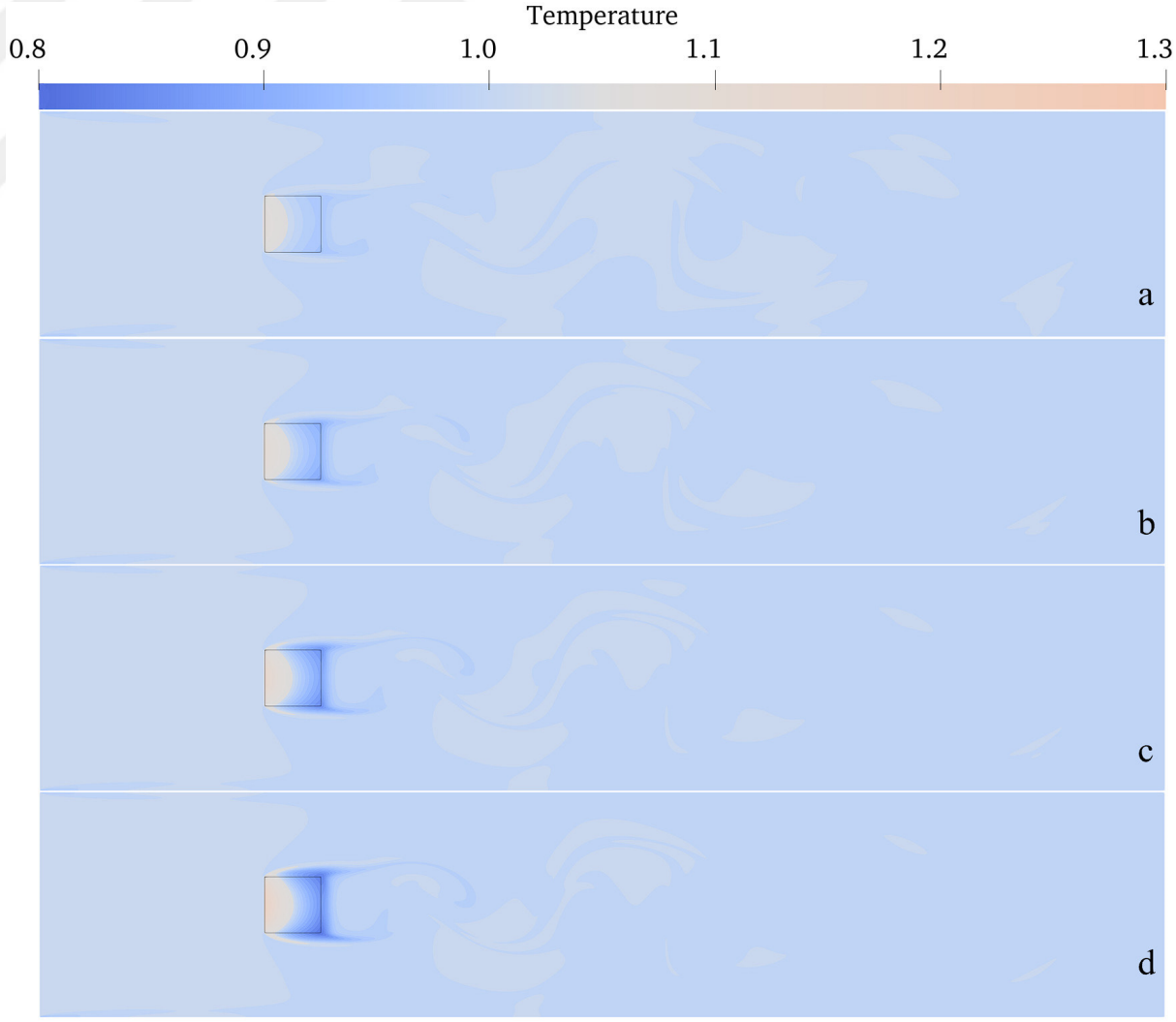


Figure 4.26 : Instantaneous contours of temperature distribution at $Pr = 10$ and $k_r = 50$ with varying Re: a) 50 b) 100 c) 150 d) 200.

5. ANALYSES OF CHT RESULTS

In previous section, the temperature contours of a square cylinder involving conjugate heat transfer were illustrated under varying Reynolds numbers (Re), conductivity ratios (k_r), and Prandtl numbers (Pr). In Chapter 5, we identify the most representative results from these temperature contours to emphasize our primary findings.

5.1 Effect of Reynolds Number

From the observations and insights of the instantaneous temperature contours shown in Figure 5.1, it is noticed that increasing Reynolds number (Re) has a consistent impact

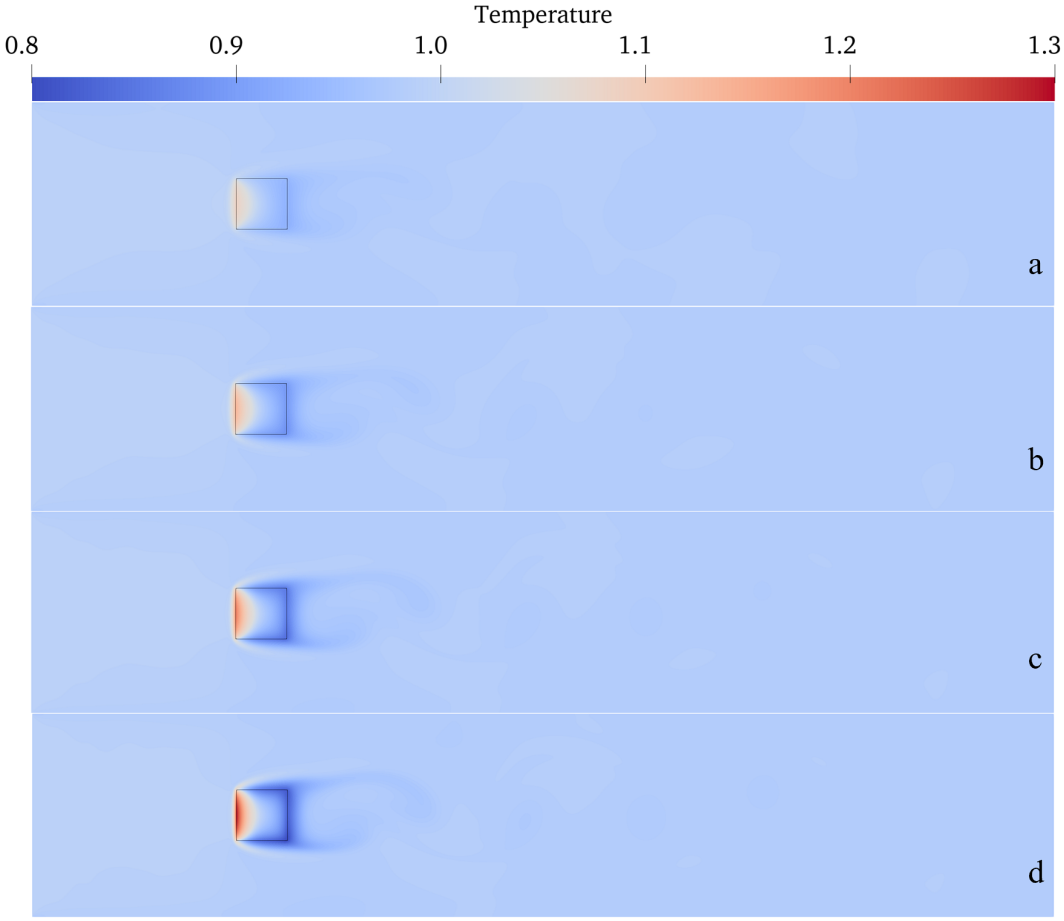


Figure 5.1 : Temperature contours for varying Re: a) 50 b) 100 c) 150 d) 200 at $Pr = 1$ and $k_r = 50$.

on the heat transfer and temperature distribution of the square cylinder, irrespective of conductivity ratio (k_r) and Prandtl number (Pr). As Re increases, the local temperatures along the front face of the solid square rise. The local temperature at the front face of the square cylinder consistently remains above the fluid temperature due to direct exposure to the incoming flow, which leads to efficient heat transfer from the fluid to the square surface. However, the top, bottom, and rear faces of the cylinder remain at relatively lower temperatures.

From the vorticity contours in Figure 4.9, we observed that free shear layers extend over the top and bottom surfaces of the cylinder, causing flow separation that intensifies with an increase in Re. As the vorticity strength increases, the flow separates immediately from the top and bottom faces at the front face corners. This creates a recirculation zone around the top, bottom, and rear faces of the cylinder, reducing effective heat transfer. This recirculation leads to the formation of low-temperature zones near these faces, as observed in the figure. The vortex shedding facilitates a cooling mechanism at the trailing edge, which was previously demonstrated by others [10]. As Re increases, a temperature gradient is observed within the cylinder, resulting in a non-uniform temperature field inside the cylinder. Even for the highest conductivity ratio, the temperature field within the cylinder is non-uniform, indicating that heat conduction inside the cylinder is influenced by flow physics. In contrast, at lower Re (50-100), the temperature distribution in the cylinder is relatively uniform, indicating efficient heat transfer from the fluid to the solid.

5.2 Effect of Conductivity Ratio

The conductivity ratio, given by the ratio of the thermal conductivity of the solid to the thermal conductivity of the fluid ($k_r = \frac{k_{solid}}{k_{fluid}}$), exerts a profound control on the conduction of heat inside the square cylinder confined in the channel. In this study, the fluid's conductivity is kept constant at $k_{fluid} = 1$. By examining the instantaneous temperature contours in Figure 5.2 for varying k_r values of 50, 10, 5, and 1, we observe that a decrease in the conductivity ratio leads to ineffective heat transfer within the cylinder and a local buildup of heat along the front face.

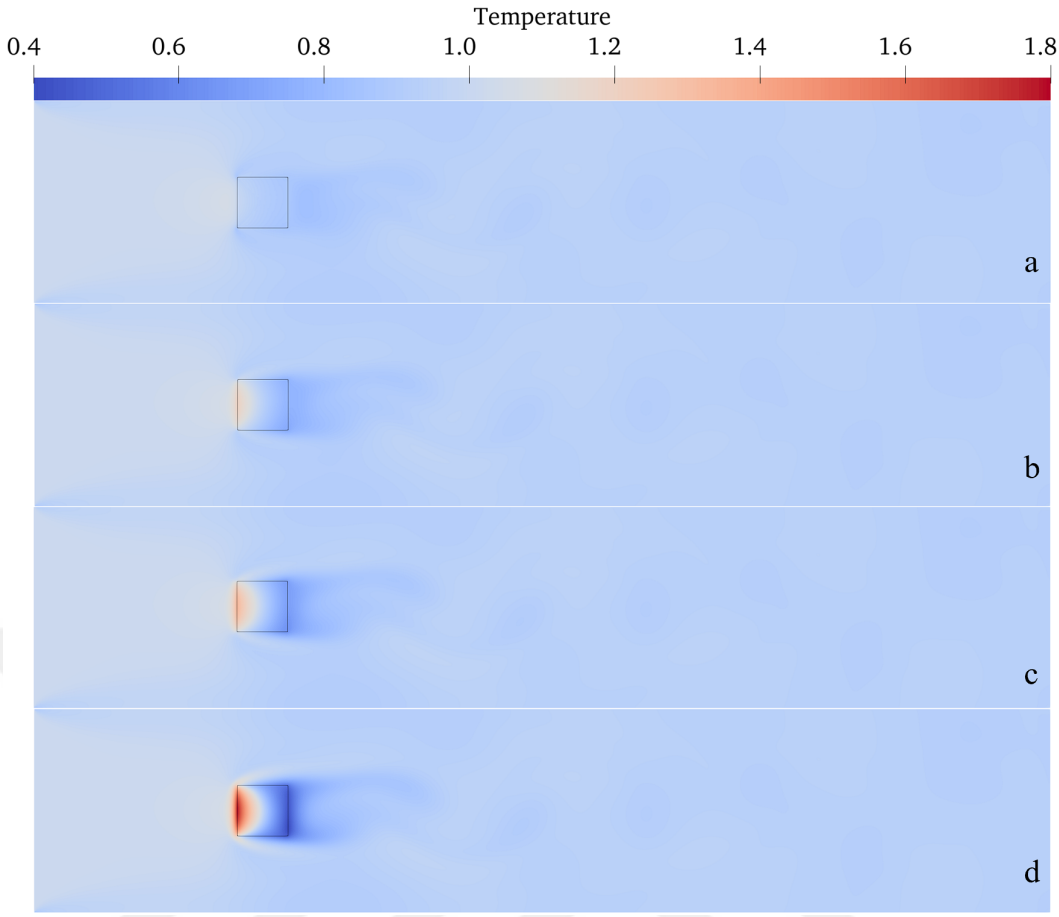


Figure 5.2 : Temperature contours for varying k_r: a) 50 b) 10 c) 5 d) 1 at $Pr = 1$ and $Re = 200$.

As k_r decreases, the temperature difference between the rear and front faces increases, resulting in a non-uniform temperature distribution in the solid cylinder. In Figure 5.2 (a), where the cylinder is characterized by the highest conductivity ratio ($k_r = 50$), heat conduction inside the cylinder is most effective, leading to a uniform temperature distribution. For Figures 5.2 (b), (c), and (d), as k_r decreases, heat conduction in the cylinder becomes increasingly limited, leading to a rise in local temperatures along the front face. Consequently, heat is not transferred adequately to the cylinder's rear face, which maintains a relatively low temperature.

It is noticeable that although the cylinder is under the same effect of flow, the temperature gradient between the fluid and the cylinder at the top and bottom faces is significant for low k_r values, indicating inefficient heat transfer. At the rear face, however, the cylinder and the fluid are at the same temperature level for all k_r values due to the formation of a dead zone, which prevents the entry of hot fluid. Overall,

when we compare Figures 5.2 and 5.3, we observe that decreasing k_r causes greater temperature non-uniformity inside the cylinder than increasing Re, as is evident from the wider temperature range.

5.3 Effect of Prandtl Number

Prandtl number (Pr) influences the heat transfer at the fluid-solid interfaces. Unlike Reynolds number (Re), which characterizes flow physics, Pr affects heat and mass transfer characteristics within fluid. By definition, it is the ratio of viscosity to thermal diffusivity of fluid. The variations in Pr impact the manner in which heat is transferred at the interfaces because at these boundaries, the fluid interacts with solid surfaces. The temperature contours of cylinder in channel for varying Pr are illustrated in Figure 5.3. Observations from the figure reveal that impact of Pr is effective in the flow field compared to both Re and k_r. Its influence extends way downstream of the cylinder

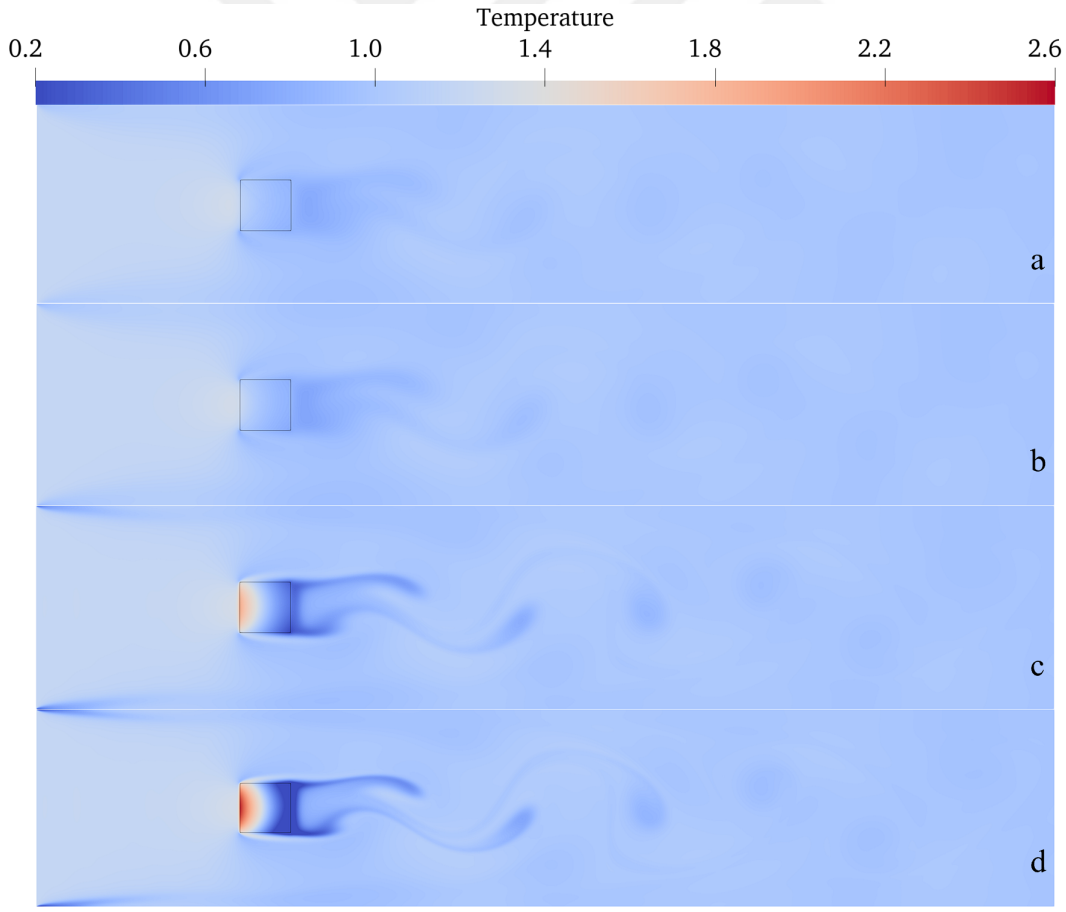


Figure 5.3 : Temperature contours for varying Pr: a) 0.71 b) 1 c) 5 d) 10 at $k_r = 50$ and $Re = 200$.

which is evident from visible temperature gradients in the wake region. For high Pr, dominant momentum diffusion along with periodic vortex shedding mechanism allows fluid to enter the wake region, creating a high temperature gradient which is correlated with vorticity contour at $Re = 200$. Additionally, we can observe from the figure that despite cylinder's high thermal conductivity, temperature distribution inside the cylinder becomes non-uniform for high Pr. As momentum diffusivity of flow is dominant, heat transfer becomes inefficent on top and bottom faces leading to formation of low temperature regions in the cylinder's vicinity. Overall, increasing Pr results in greater non-uniformity in temperature distribution within the cylinder as well as in the duct compared to the effects of k_r and Re.

5.4 Effect of Time Periodicity on Conjugate Heat Transfer

Study of unsteady behaviour of heat transfer and temperature distribution is significant for understanding the thermal transient processes. Unsteady temperature variations provide crucial insights into the temporal behavior of heat exchange mechanisms within the system. This reflects the dynamic response of the fluid-solid interfaces to changing flow and thermophysical conditions. The periodicity induced by vortex shedding in our system is a characteristic phenomenon of flow around bluff bodies. We are interested to find how profoundly it influences the distribution of temperature in the system and heat flow mechanisms.

5.4.1 Average cylinder temperature

The influence of time periodic nature of fluid flow and its effect on the instantaneous evolution of mean temperature of cylinder is studied for $Re = 100$, $k_r = 1$, and $Pr = 1$. The magnitude of velocities at a distance of D perpendicular to the trailing edge were tracked over multiple time periods. The normalised magnitudes of velocity and mean temperature of cylinder were evaluated over time. Figure 5.4 reveals that periodic velocity of fluid flow in the wake of cylinder and mean temperature of cylinder oscillate at natural frequency of vortex shedding and have a marginal phase deviation of 2.32%.

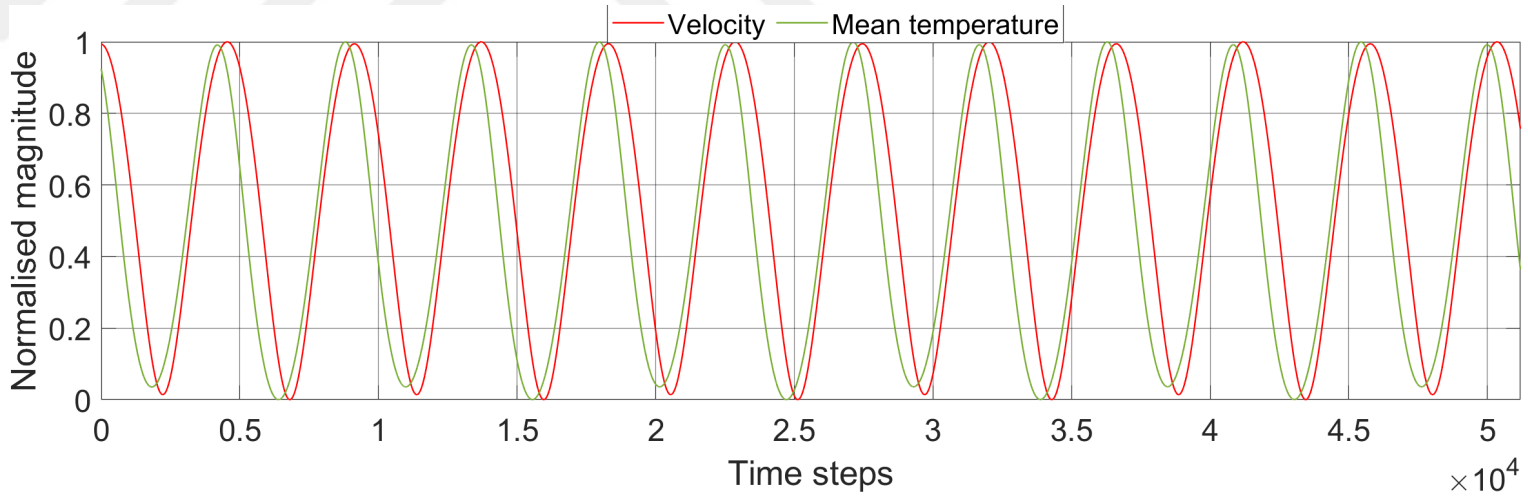


Figure 5.4 : Evolution of wake velocity and mean temperature of cylinder with time.

5.4.2 Heat transfer rate

The definition for the rate at which heat is transferred is calculated as $\dot{Q} = -k\frac{dT}{dr}$. We calculate it along each face of the cylinder and vary it with time for $Re = 100$, $k_r = 50$, and $Pr = 1$. Observations from Figure 5.5 reveal that the rate at which heat is transferred on each face is periodic. For front face (AB) and rear face (CD), frequencies of $\dot{Q_{AB}}$ and $\dot{Q_{CD}}$ are twice that of shedding frequency. On the other hand, for top face (BC) and bottom face (CD), $\dot{Q_{BC}}$ and $\dot{Q_{DA}}$ oscillate at vortex shedding frequency out of phase. Only $\dot{Q_{AB}}$ has positive magnitude where as $\dot{Q_{CD}}$, $\dot{Q_{BC}}$ and $\dot{Q_{DA}}$ have negative and limited magnitudes. This implies that front face acts as a source absorbing heat whereas rest faces of cylinder act as sinks by dissipating the heat.

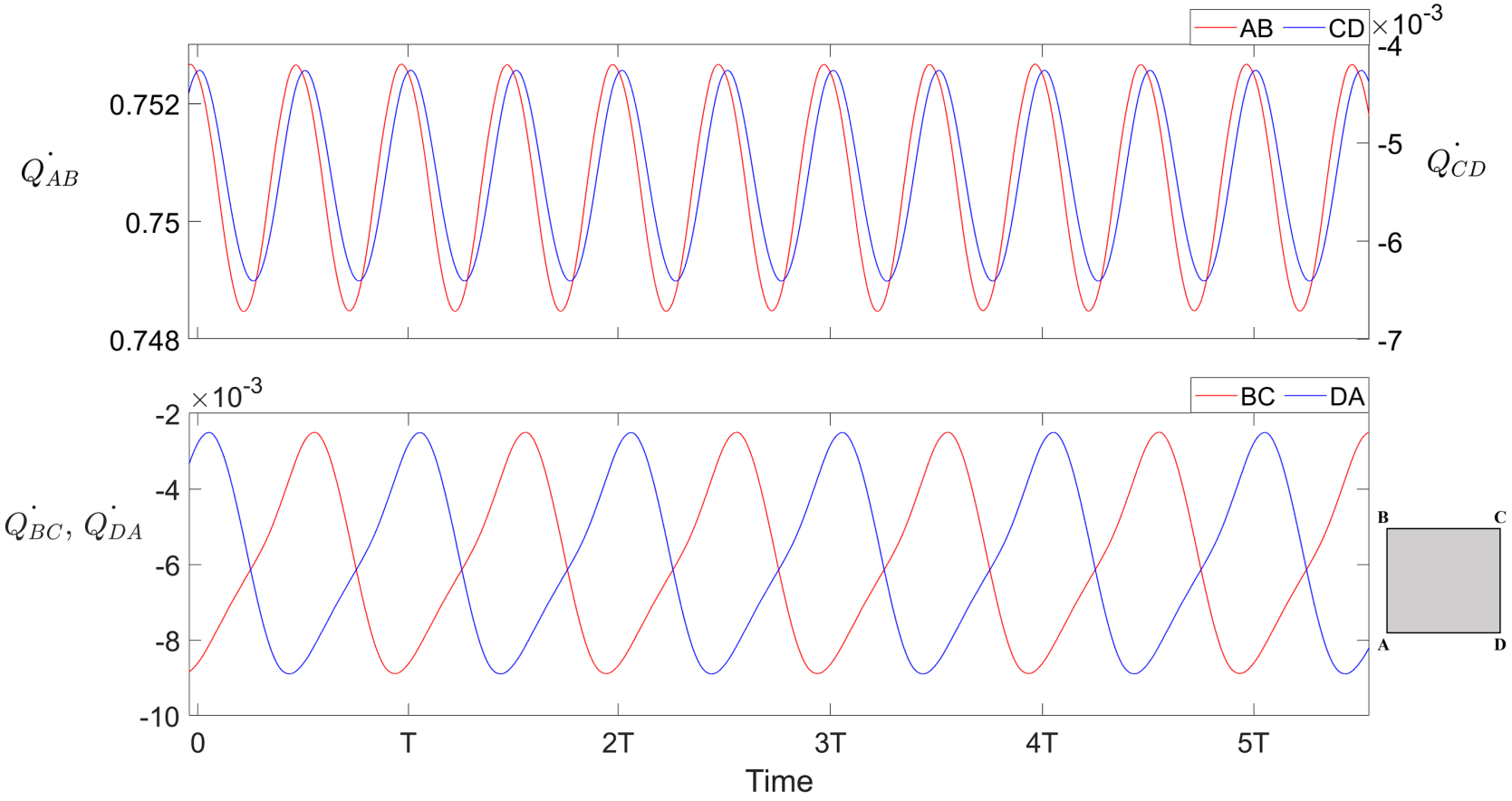


Figure 5.5 : Variation of $\dot{Q}$ on the faces of cylinder with time.

5.4.3 Surface temperature distribution

The temperature distribution over the faces of the cylinder reflects local transfer of heat on various sides. This information is crucial for pointing out areas of interest, such as regions prone to overheating and those where heat transfer is less effective. Due to the periodic nature of flow, the local temperatures are averaged over a time period as:

$$T_{av} = \frac{1}{\Gamma}\int_0^{\Gamma} T\,dt \tag{5.1}$$

The time-averaged values of temperature along the solid-fluid interfaces for different values of k_r at $Pr = 1$ are demonstrated in Figure 5.6. Each solid line represents a distinct Re value. Observations from the plots show that for any Re and k_r, the temperature on front face is highest than rest faces of cylinder. For any k_r, increasing Re raises the front face temperature while slightly lowering temperatures on other faces of cylinder. As k_r increases, we observe that peak value of temperature on the front face gets smaller whereas temperature on top, bottom and rear face rises. As a result, temperature difference between front face and rest faces of cylinder decreases and becomes minimum for $k_r = 50$. The discontinuities at the front corners become distinct for high k_r values as front and rear face are at almost same temperature. Thus, for high k_r, a uniform temperature distribution on the cylinder's surface is observed. These results confirm our earlier findings about effect of Reynolds number and conductivity ratio.

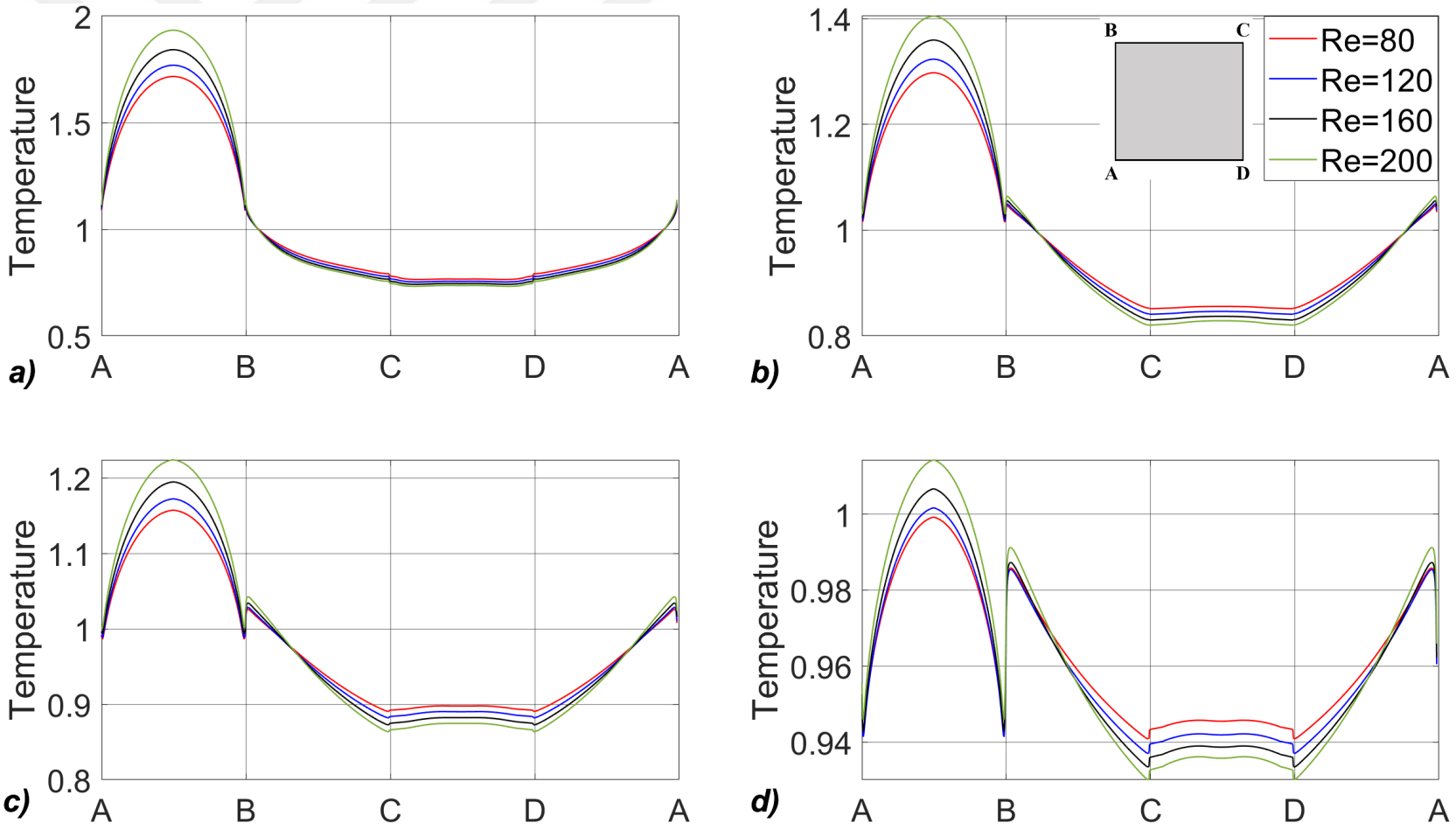


Figure 5.6 : Variation of temperature on cylinder (clockwise) at $Pr = 1$ for k_r: a) 1 b) 5 c) 10 d) 50 varying with Re.

5.5 Biot Number Study for Overall Performance

Biot number is an important parameter for quantifying the effective mechanism for transfer of heat via convection and conduction in heat transfer problems. It plays a key role in this regard and provides insights into the dominant heat transfer mechanisms at play. It offers a perspective by considering both convective heat transfer at interfaces and heat conduction in the solid material. By definition, it is calculated as ratio of convective resistance at interface to conductive resistance inside the solid. A high Biot number indicates that convective heat transfer is dominant at the interfaces and internal conduction is limited where as a low Biot number suggests that conduction in the solid dominates. For $Bi \leq 1$, no significant temperature gradients exist inside the solid and it behaves like a lumped system [67]. On the other hand, when $Bi >> 1$, the solid behaves as a thermally thick body with considerable temperature variations inside. [67]. For problems related to conjugate heat transfer, understanding the significance of Bi helps in determining the dominant mode of heat transfer. It allows engineers and researchers to assess whether convection is the dominant mode of heat transfer on the interfaces or if internal conduction within the solid material plays a more substantial role. Unlike Nusselt number which neglects the influence of solid material properties, Biot number encapsulates the heat transfer characteristics inside the solid as well as on the fluid-solid interface. This approach is essential for designing efficient thermal systems and understanding heat dissipation mechanisms.

Mathematically, Biot number is defined as $Bi_L = \frac{hD}{k}$ where D is dimension of cylinder, k is coefficient of conductivity for solid and h is coefficient of convection. We performed calculations at lattice nodes on the fluid-solid interface for computing local Biot number.Since flow is periodic in our case, we computed the surface averaged biot number, Bi_k on each face of cylinder and time averaged it over a vortex shedding period. The final value of cylinder biot number was obtained by averaging Bi_k for all faces as shown:

$$Bi_{av} = \frac{1}{4}\sum_{k=1}^{4}\frac{1}{\Gamma}\int_0^{\Gamma} Bi_k dt \tag{5.2}$$

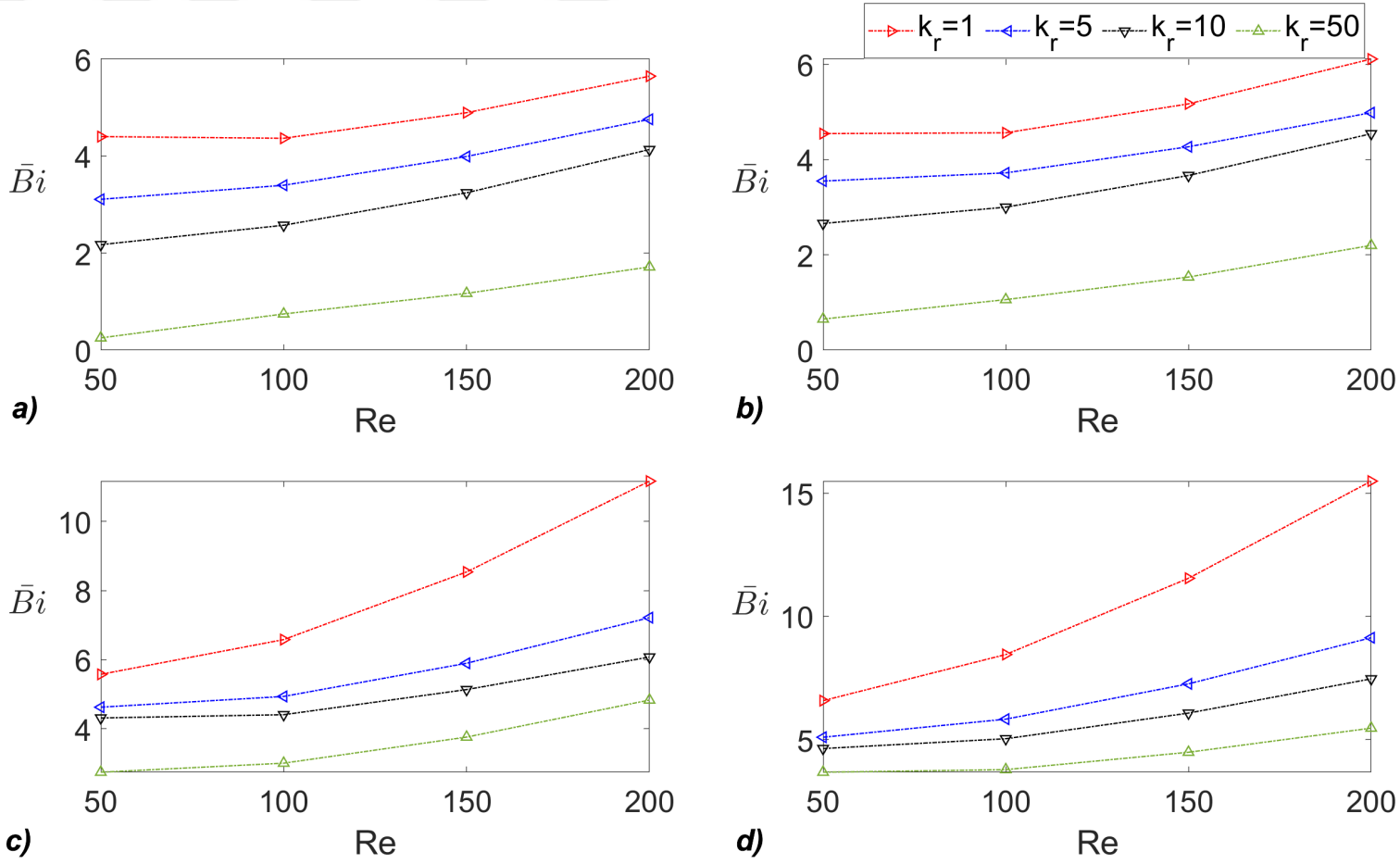


Figure 5.7 : Biot number of cylinder w.r.t *Re* for *Pr*: a) 0.71 b) 1 c) 5 d) 10 with varying k_r.

Figure 5.7 illustrates that the Biot number increases consistently with Reynolds number (Re). Particularly for low Prandtl number (Pr) flows ($Pr \leq 1$) and a conductivity ratio (k_r) of 50, Re exhibits a linear influence on the average Biot number (Bi). As Pr increases, it enhances the impact of Re, resulting in a higher Biot number. Conversely, increasing k_r leads to a reduction in the average Biot number's magnitude. Thus, the selection of flow and thermophysical properties (Re, k_r, and Pr) fundamentally determines the Biot number for the solid cylinder in this study. This analysis can be generalized for any conjugate heat transfer (CHT) problem involving external flow over a solid object. To achieve better heat conduction and a uniform temperature distribution inside the solid ($Bi \leq 1$), one should minimize Re and Pr while maximizing k_r. In this study, this behavior is observed for $Re \leq 100$ and $k_r = 50$ when Pr is 0.71 and 1. For $Pr \geq 5$, heat conduction within the cylinder becomes less efficient, leading to increased non-uniformity and temperature gradients ($Bi >> 1$).

6. CONCLUSIONS

This study underscores the effect of time dependent flow behaviour on conjugate heat transfer problems. It highlights the importance of flow and thermophysical properties such as Reynolds number, conductivity ratio, and Prandtl number. The concluding insights and results of this study are:

- Heat transfer characteristics in the cylinder are most significantly influenced by changes in Pr, followed by k_r and Re.
- For $Pr \geq 5$, temperature distribution in the cylinder becomes significantly non-uniform regardless of Re and k_r.
- A uniform temperature distribution inside the cylinder can be achieved when $Re \leq 100$, $k_r \geq 50$, and $Pr \leq 1$.
- Increasing Pr significantly alters the temperature distribution in the downstream region of the duct.
- High Re induces stronger flow separation, re-circulation zones, and increased vortex shedding around the cylinder, resulting in relatively low temperature zones near fluid-solid interfaces.
- High k_r promotes effective conduction within the cylinder, but the temperature distribution is also influenced by flow physics and fluid properties.

In conclusion, by strategically adjusting Re, k_r and Pr, we can effectively control heat transfer and temperature distribution for the cylinder. This control is vital for optimizing thermal performance in practical applications providing solutions to enhance heat dissipation and fluid-solid interaction dynamics. Further research is needed to explore transient effects and higher-dimensional flow configurations to extend the applicability of these findings. Additionally, investigating the impact

of turbulence modeling and validating our simulations with experimental data will strengthen the reliability of our computational approach. Furthermore, integrating optimization techniques and exploring novel materials for enhanced thermal properties will contribute to advancing the field of conjugate heat transfer for diverse engineering applications. Practical applications of this research include optimizing heat dissipation strategies in engineering designs such as enhancing cooling efficiency for square cylinder geometries.

REFERENCES

[1] **Mohamad, A.A.** (2019). *Lattice Boltzmann Method: Fundamentals and Engineering Applications with Computer Codes*, Springer London.

[2] **Sudhakar, T. and Das, A.K.** (2020). Evolution of multiphase lattice Boltzmann method: A review, *Journal of The Institution of Engineers (India): Series C*, *101*(4), 711–719.

[3] **NVIDIA** (2011). *NVIDIA CUDA C programming guide*, NVIDIA Corporation.

[4] **Cheng, J.**, **Grossman, M. and McKercher, T.** (2014). *Professional CUDA C programming*, John Wiley Sons.

[5] **NVIDIA** (2012). *CUDA C Best Practices Guide*, NVIDIA Corporation.

[6] **Timm, K.**, **Kusumaatmaja, H.**, **Kuzmin, A.**, **Shardt, O.**, **Silva, G. and Viggen, E.** (2016). *The Lattice Boltzmann method: Principles and Practice*, Springer Switzerland.

[7] **Delavar, M.A.**, **Farhadi, M. and Sedighi, K.** (2009). Effect of the heater location on heat transfer and entropy generation in the cavity using the lattice Boltzmann method, *Heat Transfer Research*, *40*(6).

[8] **Nia, S.N.**, **Rabiei, F.**, **Rashidi, M. and Kwang, T.** (2020). Lattice Boltzmann simulation of natural convection heat transfer of a nanofluid in a L-shape enclosure with a baffle, *Results in Physics*, *19*.

[9] **Zhou, L.**, **Cheng, M. and Hung, K.** (2005). Suppression of fluid force on a square cylinder by flow control, *Journal of fluids and structures*, *21*(2), 151–167.

[10] **Sharma, A. and Eswaran, V.** (2004). Heat and fluid flow across a square cylinder in the two-dimensional laminar flow regime, *Numerical Heat Transfer, Part A: Applications*, *45*(3), 247–269.

[11] **Nickolls, J.**, **Buck, I.**, **Garland, M. and Skadron, K.** (2008). Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?, *Queue*, *6*(2), 40–53.

[12] **Ye, Y. and Li, K.** (2013). Entropic lattice Boltzmann method based high Reynolds number flow simulation using CUDA on GPU, *Computers & Fluids*, *88*, 241–249.

[13] **Succi, S.** (2001). *The Lattice Boltzmann equation: for Fluid Dynamics and Beyond*, Oxford University Press.

[14] **Spinelli, G.G. and Celik, B.** (2015). GPU Accelarated multi-block lattice Boltzmann solver for viscous flow problems, *9th Ankara International Aerospace Conference*.

[15] **Hardy, J.**, **Pomeau, Y. and De Pazzis, O.** (1973). Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions, *Journal of Mathematical Physics*, *14*(12), 1746–1759.

[16] **Contrino, D.**, **Lallemand, P.**, **Asinari, P. and Luo, L.S.** (2014). Lattice-Boltzmann simulations of the thermally driven 2D square cavity at high Rayleigh numbers, *Journal of Computational Physics*, *275*, 257–272.

[17] **Rahmati, A.**, **Ashrafizaadeh, M. and Shirani, E.** (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, *Journal of Applied Fluid Mechanics*, *7*(1), 89–102.

[18] **Jahanshaloo, L.**, **Sidik, N.A.C.**, **Fazeli, A. and HA, M.P.** (2016). An overview of boundary implementation in lattice Boltzmann method for computational heat and mass transfer, *International Communications in Heat and Mass Transfer*, *78*, 1–12.

[19] **Yu, H.**, **Luo, L.S. and Girimaji, S.S.** (2006). LES of turbulent square jet flow using an MRT lattice Boltzmann model, *Computers & Fluids*, *35*(8-9), 957–965.

[20] **Grunau, D.**, **Chen, S. and Eggert, K.** (1993). A lattice Boltzmann model for multiphase fluid flows, *Physics of Fluids A: Fluid Dynamics*, *5*(10), 2557–2562.

[21] **Pan, C.**, **Luo, L.S. and Miller, C.T.** (2006). An evaluation of lattice Boltzmann schemes for porous medium flow simulation, *Computers & fluids*, *35*(8-9), 898–909.

[22] **Wood, A.M.** (1999). *Lattice Boltzmann Magnetohydrodynamics*, University of Glasgow, UK.

[23] **Lätt, J.**, **Chopard, B.**, **Succi, S. and Toschi, F.** (2006). Numerical analysis of the averaged flow field in a turbulent lattice Boltzmann simulation, *Physica A: Statistical Mechanics and its Applications*, *362*(1), 6–10.

[24] **Spinelli, G.G. and Celik, B.** (2017). Conjugate heat transfer computations with lattice Boltzmann method and comparisons its results with finite volume method, *Proceedings of 9th Ankara International Aerospace Conference*.

[25] **Tarokh, A.**, **Mohamad, A. and Jiang, L.** (2013). Simulation of conjugate heat transfer using the lattice Boltzmann method, *Numerical Heat Transfer, Part A: Applications*, *63*(3), 159–178.

[26] **Widyaparaga, A.** (2013). Simulation of conjugate convective-conductive heat transfer in a microchannel within the slip regime using GPU accelerated lattice Boltzmann method, *11th International Conference on Nanochannels, Microchannels, and Minichannels*, American Society of Mechanical Engineers.

[27] **Ren, Q. and Chan, C.L.** (2016). Natural convection with an array of solid obstacles in an enclosure by lattice Boltzmann method on a CUDA computation platform, *International Journal of Heat and Mass Transfer*, *93*, 273–285.

[28] **Hosseini, S.A.**, **Darabiha, N. and Thévenin, D.** (2019). Lattice Boltzmann advection-diffusion model for conjugate heat transfer in heterogeneous media, *International Journal of Heat and Mass Transfer*, *132*, 906–919.

[29] **Chen, S.**, **Yan, Y. and Gong, W.** (2017). A simple lattice Boltzmann model for conjugate heat transfer research, *International Journal of Heat and Mass Transfer*, *107*, 862–870.

[30] **Fedorov, A.G. and Viskanta, R.** (2000). Three-dimensional conjugate heat transfer in the microchannel heat sink for electronic packaging, *International Journal of Heat and Mass Transfer*, *43*(3), 399–415.

[31] **Gombosi, T.I.** (1994). *Gaskinetic theory*, Cambridge University Press.

[32] **Cercignani, C.**, (2002). The Boltzmann Equation and Fluid Dynamics, Handbook of Mathematical Fluid Dynamics, volume 1, Elsevier, pp.1–69.

[33] **Wolf-Gladrow, D.A.** (2004). *Lattice-gas Cellular Automata and Lattice Boltzmann Models: An Introduction*, Springer.

[34] **Bhatnagar, P.L.**, **Gross, E.P. and Krook, M.** (1954). A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Physical review*, *94*(3), 511–525.

[35] **Karlin, I.V.**, **Ferrante, A. and Öttinger, H.C.** (1999). Perfect entropy functions of the lattice Boltzmann method, *Europhysics Letters*, *47*(2), 182–188.

[36] **Lallemand, P. and Luo, L.S.** (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, *Physical Review E*, *61*(6), 6546–6562.

[37] **Huang, H.**, **Krafczyk, M. and Lu, X.** (2011). Forcing term in single-phase and Shan-Chen-type multiphase lattice Boltzmann models, *Physical Review E*, *84*(4).

[38] **Shan, X. and Doolen, G.** (1995). Multicomponent lattice-Boltzmann model with interparticle interaction, *Journal of statistical physics*, *81*(1-2), 379–393.

[39] **Mattila, K.K.**, **Siebert, D.N.**, **Hegele Jr, L.A. and Philippi, P.C.** (2013). High-order lattice-Boltzmann equations and stencils for multiphase models, *International Journal of Modern Physics C*, *24*(12).

[40] **Zibart, A. and Kenig, E.** (2021). Numerical investigation of conjugate heat transfer in a pillow-plate heat exchanger, *International Journal of Heat and Mass Transfer*, *165*, 120567.

[41] **Kalantari, H.**, **Ghoreishi-Madiseh, S.A.**, **Kurnia, J.C. and Sasmito, A.P.** (2021). An analytical correlation for conjugate heat transfer in fin and tube heat exchangers, *International Journal of Thermal Sciences*, *164*.

[42] **Islam, M.S.**, **Saha, S.C.** *et al.* (2021). Heat transfer enhancement of modified flat plate heat exchanger, *Applied Thermal Engineering*, *186*.

[43] **Li, Y. and Kong, S.C.** (2011). Coupling conjugate heat transfer with in-cylinder combustion modeling for engine simulation, *International Journal of Heat and Mass Transfer*, *54*(11-12), 2467–2478.

[44] **Chen, Y.**, **Wei, H. and Zu, Y.** (2018). Experimental study on the conjugate heat transfer of double-wall turbine blade components with/without pins, *Thermal Science and Engineering Progress*, *8*, 448–456.

[45] **Laitinen, A.**, **Saari, K.**, **Kukko, K.**, **Peltonen, P.**, **Laurila, E.**, **Partanen, J. and Vuorinen, V.** (2020). A CFD study by conjugate heat transfer in OpenFOAM: A liquid cooling concept for high power electronics, *International Journal of Heat and Fluid Flow*, *85*, 108654.

[46] **Stone, H.A.**, **Stroock, A.D. and Ajdari, A.** (2004). Engineering flows in small devices: microfluidics toward a lab-on-a-chip, *Annu. Rev. Fluid Mech.*, *36*, 381–411.

[47] **Wang, J.**, **Wang, M. and Li, Z.** (2007). A lattice Boltzmann algorithm for fluid–solid conjugate heat transfer, *International journal of thermal sciences*, *46*(3), 228–234.

[48] **Horvat, A. and Catton, I.** (2003). Numerical technique for modeling conjugate heat transfer in an electronic device heat sink, *International Journal of Heat and Mass Transfer*, *46*(12), 2155–2168.

[49] **Chen, X. and Han, P.** (2000). A note on the solution of conjugate heat transfer problems using SIMPLE-like algorithms, *International Journal of Heat and Fluid Flow*, *21*(4), 463–467.

[50] **Perelman, T.** (1961). On conjugated problems of heat transfer, *International Journal of Heat and Mass Transfer*, *3*(4), 293–303.

[51] **Li, L.**, **Chen, C.**, **Mei, R. and Klausner, J.F.** (2014). Conjugate heat and mass transfer in the lattice Boltzmann equation method, *Physical Review E*, *89*(4).

[52] **Hu, Y.**, **Li, D.**, **Shu, S. and Niu, X.** (2015). Full Eulerian lattice Boltzmann model for conjugate heat transfer, *Physical Review E*, *92*(6).

[53] **Nee, A.**, **Kim, B. and Chamkha, A.J.** (2023). Three-dimensional simulation of full conduction-convection-radiation coupling with high Rayleigh numbers, *International Journal of Thermal Sciences*, *184*.

[54] **Çelik, B. and Spinelli, G.G.** (2017). Conjugate heat transfer computations via lattice Boltzmann method, *Journal of Aeronautics and Space Technologies*, *10*(2), 49–59.

[55] **Kuzmin, A.**, **Mohamad, A. and Succi, S.** (2008). Multi-relaxation time lattice Boltzmann model for multiphase flows, *International Journal of Modern Physics C*, *19*(06), 875–902.

[56] **Chang, C.**, **Liu, C.H. and Lin, C.A.** (2009). Boundary conditions for lattice Boltzmann simulations with complex geometry flows, *Computers & Mathematics with Applications*, *58*(5), 940–949.

[57] **Liu, C.H.**, **Lin, K.H.**, **Mai, H.C. and Lin, C.A.** (2010). Thermal boundary conditions for thermal lattice Boltzmann simulations, *Computers & Mathematics with Applications*, *59*(7), 2178–2193.

[58] **Zou, Q. and He, X.** (1997). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, *Physics of fluids*, *9*(6), 1591–1598.

[59] **Maier, R.S.**, **Bernard, R.S. and Grunau, D.W.** (1996). Boundary conditions for the lattice Boltzmann method, *Physics of Fluids*, *8*(7), 1788–1801.

[60] **Karpiński, M.**, **Bialik, R.J. and Rowiński, P.M.** (2013). Application of Lattice Boltzmann Method for generation of flow velocity field over river bed-forms, *Experimental and Computational Solutions of Hydraulic Problems: 32nd International School of Hydraulics*, 327–335.

[61] **Chen, Q.**, **Zhang, X. and Zhang, J.** (2013). Improved treatments for general boundary conditions in the lattice Boltzmann method for convection-diffusion and heat transfer processes, *Physical Review E*, *88*(3).

[62] **Filippova, O. and Hänel, D.** (2000). Acceleration of lattice-BGK schemes with grid refinement, *Journal of Computational Physics*, *165*(2), 407–427.

[63] **Hecht, M. and Harting, J.** (2010). Implementation of on-site velocity boundary conditions for D3Q19 lattice Boltzmann simulations, *Journal of Statistical Mechanics: Theory and Experiment*, *2010*(01).

[64] **Beskok, A.**, **Raisee, M.**, **Celik, B.**, **Yagiz, B. and Cheraghi, M.** (2012). Heat transfer enhancement in a straight channel via a rotationally oscillating adiabatic cylinder, *International Journal of Thermal Sciences*, *58*, 61–69.

[65] **Celik, B.**, **Raisee, M. and Beskok, A.** (2010). Heat transfer enhancement in a slot channel via a transversely oscillating adiabatic circular cylinder, *International Journal of Heat and Mass Transfer*, *53*(4), 626–634.

[66] **Hussain, A. and Celik, B.** (2024). Conjugate Heat Transfer for a 2D Square Cylinder Using Lattice Boltzmann Method, ***S. Singh, P.J. Ramulu and S.S. Gautam****, editors, Recent Advances in Aerospace Engineering*, Springer Nature Singapore, Singapore, pp.361–372.

[67] **Lienhard IV, J.H.** (2024). *A Heat Transfer Textbook*, Cambridge.

CURRICULUM VITAE

Name: Aanif Hussain

EDUCATION:

- **Undergraduate :** B. Tech in Aerospace Engineering (2017-21), Amity University, Noida, India.
- **Graduate :** M.Sc. in Aeronautical and Astronautical Engineering (2022-24), Istanbul Technical University, Istanbul, Türkiye.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Hussain, A., Celik, B.** (2024) Conjugate Heat Transfer for a 2D Square Cylinder Using Lattice Boltzmann Method. In: Singh, S., Ramulu, P.J., Gautam, S.S. (eds) Recent Advances in Aerospace Engineering. MRAE 2023. *Lecture Notes in Mechanical Engineering. Springer, Singapore,pp. 361-72.*