A Boundary Condition-based Machine Learning Algorithm for Fast Prediction of Hypersonic Flows

A Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Rachakonda Naga Sai Prakash

Roll No: 206103020

Under the Supervision of

Prof. Tapan Krishnakumar Mankodi
and
Prof. Niranjan Sahoo

Department of Mechanical Engineering

Indian Institute of Technology Guwahati

Guwahati – 781039, Assam, India

June 2026

.

Dedicated
to
Sri. Ekkirala Anantha Krishna Garu

DECLARATION

I hereby certify that the work compiled in this dissertation is the outcome of the research work performed by me, unless otherwise stated, under the guidance of Prof. Tapan Krishnakumar Mankodi and Prof. Niranjan Sahoo.

Any part of this work has not been submitted for the award of any degree, diploma, associate-fellowship, fellowship, or its equivalent to any university or institution.

Rachakonda Naga Sai Prakash
Roll No. 206103020
Department of Mechanical Engineering
Indian Institute of Technology Guwahati

CERTIFICATE

This is to certify that the thesis entitled "A Boundary Condition-based Machine Learning Algorithm for Fast Prediction of Hypersonic Flows", submitted by **Rachakonda Naga Sai Prakash** (**Roll No: 206103020**), a research scholar of the Department of Mechanical Engineering, Indian Institute of Technology Guwahati, in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy, has been examined and approved.

The thesis embodies the original research work carried out by the candidate under our supervision and has reached the standard required for submission in accordance with the regulations of the Institute.

Prof. Tapan Krishnakumar Mankodi
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Guwahati

Prof. Niranjan Sahoo
Professor
Department of Mechanical Engineering
Indian Institute of Technology Guwahati

Place: Guwahati
Date: June 30^{th}, 2026

ACKNOWLEDGEMENTS

At first, I would like to express my heartfelt gratitude to my supervisors, **Prof. Tapan Krishnakumar Mankodi** and **Prof. Niranjan Sahoo**, who has relentlessly supported me during the course of my research work with their motivation, encouragement, and colossal knowledge. Their guidance has greatly shaped my basic understanding of this domain. Without their keen interest and help, I would not have been able to put forth the best foot forward during the current research work. I am also deeply obligated to the Director of the Indian Institute of Technology Guwahati for providing such pleasing circumstances for the inception of my research career.

Besides my supervisor(s), I would like to acknowledge the doctoral committee members, **Prof. Bhaskar Kumar**, **Prof. Rishikesh Kulkarni** ,**Prof. Vinayak Kulkarni**, and **Prof. Anoop Kumar Dass** for their valuable recommendations and insightful remarks throughout the course of research activities, which have constantly engaged me towards undertaking a meaningful thesis work. I am extremely grateful to all the faculty and staff members in the Department of Mechanical Engineering for their wholehearted support throughout my stay.

Further, special appreciation for the support staff of the Department of Mechanical Engineering and Computer and Communication Centre, IIT Guwahati, especially for providing access to Param Kamrupa supercomputing facilities for simulations.

I sincerely thank my fellow lab members of the Computational Fluid Dynamics Laboratory, **Mr. Harshal Srivastava**, **Mr. Arnab Kumar Das**, **Mr. Ashirbad Mallick**, **Mr. Nishant Sharma**, **Ms. Sumati Raghav**, Mr. Nikhil Dewangan, **Mr. Sriram Vana Mr. Onkar Vijay Bichkar**, and **Mr. Abhisek Raj**, for their valuable discussions, collaborative spirit, and constant support. The stimulating research environment and mutual encouragement within this group greatly enhanced my doctoral experience and contributed meaningfully to this work.

ABSTRACT

A new deep neural network-based machine learning algorithm has been developed for predicting the flow field around a reentry vehicle in supersonic and hypersonic conditions for the entire trajectory from 85 km to 40 km. The gas flow field depends on the geometry, inlet conditions (ICs), and boundary conditions (BCs). In the conventional approach, the compressible Navier-Stokes-Fourier (NSF) equations are solved on a discretised computational domain with appropriate initial and boundary conditions, using either the Riemann or the Boltzmann approach, which is a computationally intensive task. At higher altitudes, the alternative particle-based direct simulation Monte Carlo (DSMC) is used to study the flow physics. The proposed approach is our first attempt to predict the flow field using deep neural networks without explicitly solving the NSF equations or running DSMC simulations. Further, the proposed "boundary condition-based machine learning algorithm (BCML)" falls under the category of physics-assisted neural network. The training data for the low-altitude BCML model is generated using an in-house Riemann flux solver-based NSF code and `OpenFoam`-based `hy2Foam` solver, and converted into a BCML-compatible format. An in-house DSMC solver was used to generate data for the high-altitude BCML framework. After appropriate training, it was found that the BCML models were able to capture shock structures and other related wave patterns in the flow around a relatively arbitrarily shaped bluff body, with arbitrary initial and boundary conditions at any altitude within the extrema to a reasonable accuracy. The BCML models were able to predict the flow fields within a substantially lower computational time compared to conventional NSF-CFD and DSMC simulations for the cases considered in the present study.

In addition to fast prediction of flow fields, BCML models can improve the overall efficiency of traditional solvers. Two examples, one for the continuum-based NSF solver and the other related to the particle-based DSMC solver, have been focused on in the present thesis. The BCML-generated flow fields can be used as initial conditions for the NSF solver, thereby improving the simulation's convergence. Also, the BCML-generated optimised mesh for the DSMC was found to be approximately 2 times faster than a general mesh. The overall potential of the BCML model, along with its criticisms, is also presented in this thesis.

Keywords: Machine Learning-assisted Computational Fluid Dynamics (ML-CFD), Hypersonic Aerothermodynamics, Non-Equilibrium and Rarefied Gas Dynamics

Contents

List of Figures

List of Tables

List of Symbols

Latin Symbols

$A(U)$	Jacobian matrix (Coefficient matrix)
a	Sonic speed
C_P	Specific heat at constant pressure
C_V	Specific heat at constant volume
e	Total energy per unit mass
$F(U)$	x-direction flux vector
$G(U)$	y-direction flux vector
$H(U)$	z-direction flux vector
K	Eigen vector of the Jacobian matrix corresponding to λ eigen value
k	Coefficient of thermal conductivity
Ma	Mach number
Kn	Knudsen number
Re	Reynolds number
p	Pressure
T	Temperature
U	Vector of conserved quantities
u	Fluid velocity component in the x-direction
v	Fluid velocity component in the y-direction
w	Fluid velocity component in the z-direction

Greek Symbols

ρ	Density of fluid
λ	Eigen value of the Jacobian matrix
γ	Ratio of specific heats
μ	Dynamic viscosity

Subscript Symbols

r	Rotational
t	Translational
tr	Transrotational
v	Vibrational
x	x-direction
y	y-direction
z	z-direction

Abbreviations

$1D$	One-dimensional
$3D$	Three-dimensional
$AUSM$	Advection Upstream Splitting Method
AOA	Angle of Attack
ANN	Artificial Neural Network
BC	Boundary Conditions
$BCML$	Boundary Condition based Machine learning
CFL	Courant Friedrich Lewy
CFD	Computational Fluid Dynamics
CGD	Computational Gas Dynamics
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DL	Deep Learning
$DSMC$	Direct Simulation Monte Carlo
FVM	Finite Volume Method
FDS	Flux Difference Splitting
FVS	Flux Vector Splitting
ML	Machine Learning
MLP	Multi-layer Perceptron
MSE	Mean square error
MAE	Mean Absolute error
MPI	Message Passing Interface
PDE	Partial Differential Equation
$PINN$	Physics Informed Neural Networks
$R-T$	Rotational-Translational
$V-T$	Vibrational-Translational
TPS	Thermal Protection System

Chapter 1

Introduction

1.1 Background

Aerospace technology has long been a subject of scientific curiosity, gaining greater prominence in recent times. Over the past few decades, substantial progress in launch vehicle and satellite technologies has enabled several nations to achieve independent access to space, leading to rapid developments in satellites, spacecraft, and atmospheric re-entry vehicles. This evolution has introduced a wide range of new and complex aerodynamic challenges, particularly in high-speed flight regimes. During an atmospheric re-entry, vehicles operate at extremely high velocities, commonly referred to as the hypersonic regime, typically defined by flight Mach numbers exceeding 5. Hypersonic flows are characterised by distinctive physical phenomena, including thin shock layers, comparatively thick boundary layers, strong vortical structures, pronounced viscous–inviscid interactions, and significant high-temperature effects arising from chemical and thermal non-equilibrium and leading to aerodynamic heating. Mission-specific requirements, such as the flight trajectory and vehicle geometry, strongly influence the intensity of these phenomena. Consequently, the overall understanding of aerodynamics, flight dynamics, propulsion, and structural integrity becomes considerably more critical for hypersonic vehicles than for conventional aircraft. Typical flow features associated with hypersonic flight are illustrated in Figure 1.1.

In the field of hypersonic aerodynamics, re-entry vehicle design is a crucial aspect central to space exploration. A re-entry vehicle experiences different phases of flow regimes as it enters Earth's atmosphere from outer space. The Knudsen number (Kn), defined as the ratio of the mean free path (the average distance a particle travels between collisions) to the characteristic length, can be used to distinguish among these flow regimes [1].

$$\mathrm{Kn} = \frac{\text{mean free path}}{\text{characteristic length}} \tag{1.1}$$

The mean free path depends strongly on the local number density in the gas flow problem. In the limit of zero Knudsen number, intermolecular collisions dominate, and the gas is in perfect thermodynamic equilibrium. However, as the Knudsen number increases, molecular

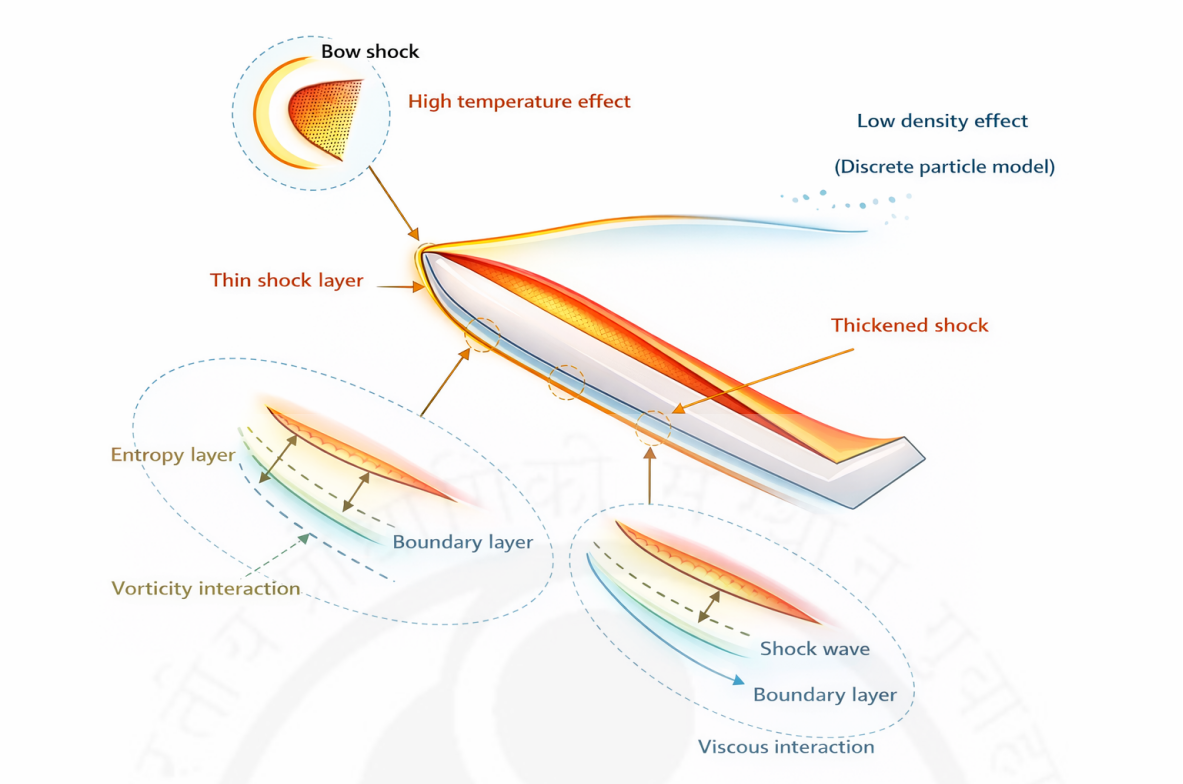

Figure 1.1: Flow characteristics of hypersonic regime

collisions become less frequent until the free molecular limit is reached, where intermolecular collisions are negligible. Since a system is directed toward thermodynamic equilibrium through intermolecular collisions, it is clear that as the Knudsen number increases, non-equilibrium effects become more prominent [2]. The flow is classified into different regimes based on the Knudsen number:

- $\mathrm{Kn} \to 0$: inviscid flow (Euler equations)
- $\mathrm{Kn} < 0.001$: continuum regime (Navier–Stokes–Fourier equations)
- $0.001 \leq \mathrm{Kn} \leq 0.1$: slip regime (Navier–Stokes–Fourier equations with velocity slip and temperature-jump boundary conditions)
- $0.1 < \mathrm{Kn} < 10$: transition regime (Boltzmann equation or particle methods such as DSMC)
- $\mathrm{Kn} \geq 10$: free-molecular regime (Collision-less Boltzmann equation or particle methods such as DSMC)

The aerodynamic and aero-thermal analysis of hypersonic flows directly impacts the design of re-entry vehicles [3]. Upon descending through several layers of Earth's atmosphere, the re-entry vehicle encounters severe aerodynamic heating [4], which helps decelerate the vehicle at very high re-entry speeds ($> Ma$ 10). High re-entry speeds result in high temperatures in the shock and post-shock regions surrounding the re-entry vehicle. High-field temperatures

($>> 5000\ K$) not only excite higher vibrational levels of the molecular species in the air but also drive various chemical and ionisation reactions in the post-shock regions, further adding to the complexity of the flow. At such high temperatures, several thermochemical reactions occur and must be considered for design purposes.

Since experimental analysis of such high-speed high-energy flows is difficult and expensive, numerical techniques are used to study the complex physical phenomena. Accurate simulation of hypersonic flows is crucial to the design of a thermal protection system (TPS) [5], which enables the re-entry vehicle to land safely on Earth's surface. Generally, a Navier-Stokes-Fourier (NSF) based computational fluid dynamics (CFD) algorithm is employed to study the flow physics around re-entry vehicles at lower altitudes (< 60 km), where the continuum assumption is valid. At higher altitudes, NSF-based CFD equations lack accuracy due to the highly rarefied nature of the flow. At intermediate altitudes, second-order and higher-order hydrodynamic models offer an alternative theory [6–13] beyond the NSF framework. Second-order hydrodynamical equations, such as Burnett [6] and Onsager-Burnett [7, 11–13] equations, are derived from classical kinetic theory using the Chapman-Enskog expansion of the Boltzmann equation. These equations often suffer from linear instability and have been extended to include the third-order terms, leading to super-Burnett equations where the high-frequency instabilities of the second-order terms are suppressed. However, these higher-order formulations still fail to provide accurate or stable solutions when encountering highly non-equilibrium conditions. For flows with a high degree of non-equilibrium, an alternative method, the direct simulation Monte Carlo (DSMC) method, is employed [14, 15]. One of the remarkable strengths of the DSMC method is its ability to accurately capture the physics of rarefied gases with a manageable number of representative particles. However, the DSMC method is computationally expensive, particularly for near-continuum flows and three-dimensional problems. Also, three-dimensional NSF-based CFD simulations are computationally costly, though less so than DSMC simulations.

The DSMC method is a Lagrangian method that models rarefied gas flows using particles that represent a large number of real atoms or molecules. These particles move and collide in a statistically consistent manner. Although the DSMC method is a particle method, it uses an underlying mesh to cover the flow domain, which serves as a cut-off parameter for particle collisions. The assumption that the movement and collision modules in DSMC are decoupled is valid when the time step is smaller than the local mean collision time, and the cell size is less than the local mean free path. During the movement step, particles propagate along rectilinear paths according to their velocities, while interactions with walls are modelled using appropriate gas–surface interaction models. In the collision step, particles are grouped within computational cells and stochastic binary collisions are sampled to reproduce the Boltzmann collision integral. Macroscopic flow properties are obtained by time-averaging

molecular quantities over a large number of sampling steps.

A three-dimensional hypersonic flow over a realistic re-entry vehicle often requires 10-100 million cells and close to a billion DSMC particles. The computational cost, even on a parallel architecture, is significantly high. Even in the case of an NSF-based solver, such simulations are computationally expensive, albeit not as high as those in the case of DSMC simulations. One of the key challenges in computational hypersonic gas dynamics is reducing computational cost for such a critical application without compromising accuracy. The present work attempts to do so using machine learning (ML) techniques.

1.2 Machine Learning in Fluid Dynamics

In recent years, substantial research efforts have been devoted to the use of data-driven and Machine learning-based techniques for investigating fluid dynamics, enabling efficient simulation and analysis of a broad spectrum of flow problems [16]. Here, machine learning (ML) techniques can serve as a promising alternative to or complement DSMC/NSF-CFD simulations by accelerating them and subsequently reducing overall computational costs. ML techniques have already been applied to several problems in fluid mechanics [17, 18], receiving particularly positive attention in the field of turbulence modelling [19–22], compressible gas flow modelling [23].

Recently, several physics-informed neural network (PINN) models [24–26] have been reported for rarefied [27, 28] and hypersonic flows [29, 30] in recent years. Tucny et al. [31] recently trained a PINN to predict flows through a two-dimensional array of cylinders over a wide range of Knudsen numbers. The PINN was trained using a DSMC-generated dataset, and the predicted flow fields were found to be highly accurate for Knudsen numbers included in the training. One of the key advantages of PINN models is that the governing physical laws are directly embedded, which ensures that the solutions remain physically meaningful while reducing reliance on massive datasets. However, it is well established that PINNs in their standard form face challenges in handling multi-scale physics problems, such as stiff chemical kinetics equations in hypersonic flows [32] and flows with sharp gradients [33, 34].

Broadly, the current interest in ML-CFD can be classified into two major approaches: (a) employing ML models for the closure of partial differential equations and (b) flow-field reconstruction. However, many studies in the latter category are essentially simplistic regression models with limited applicability, especially when test conditions lie outside the training data. Moreover, such ML models are often restricted to specific flow regimes and geometries. To address these limitations, the present work introduces a new ML approach: the **boundary condition-based machine learning (BCML)** model, which can predict flow fields over

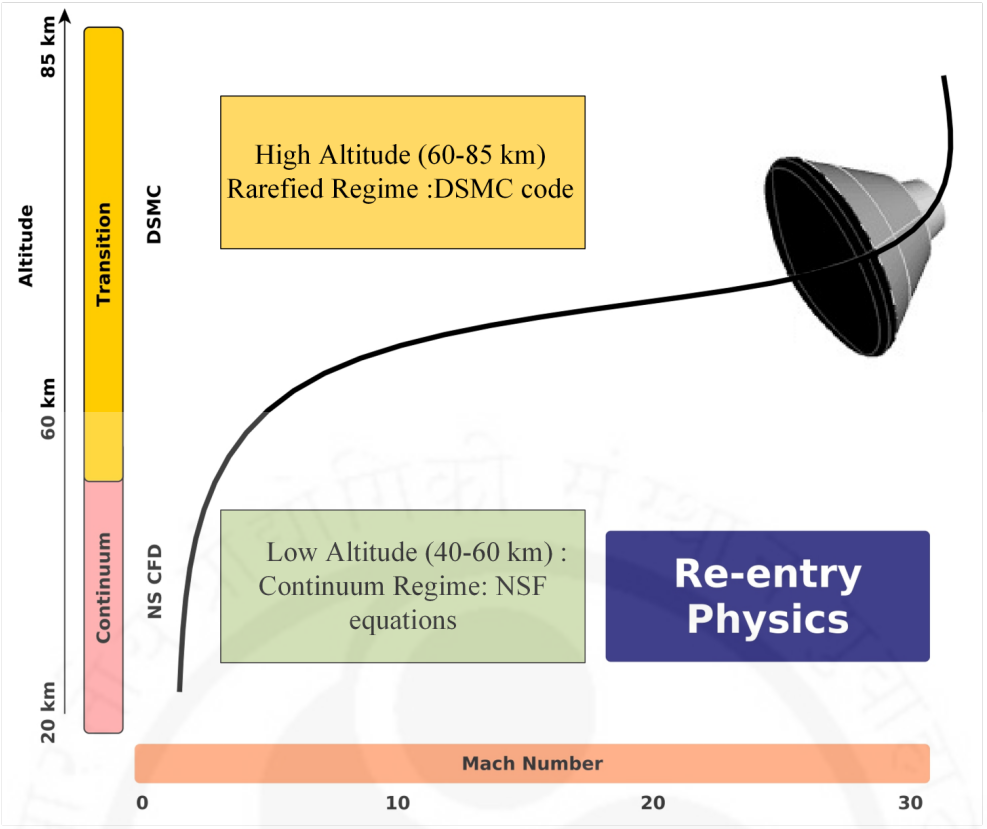

Figure 1.2: Classification of the new BCML models along the trajectory of the re-entry vehicle during deceleration into Earth's atmosphere

a wider range of flow conditions and for arbitrary geometries not included in the training set. A key feature of the model is that it incorporates the influence of various boundary conditions on a point in the domain through the concept of generators. In the present work, several BCML models have been developed for the class of re-entry flow, spanning a range from 40 km to 85 km, and covering a wide range of Mach numbers (Ma 2-35), encompassing the entire critical trajectory as shown in Figure 1.2.

1.3 Motivation

In many practical cases, the computational time required to obtain converged solutions for such problems ranges from several days to months, limiting rapid design iteration and real-time analysis. Consequently, there is a strong need to develop alternative methodologies that can predict hypersonic flow fields with comparable accuracy at substantially reduced computational cost. With the increasing availability of high-fidelity datasets from CFD and DSMC simulations, data-driven and machine-learning-based approaches have gained attention in computational fluid dynamics. These methods exploit existing simulation data to learn complex flow physics and have demonstrated the potential to reduce solution times from days to minutes for highly complex problems. This provides strong motivation to

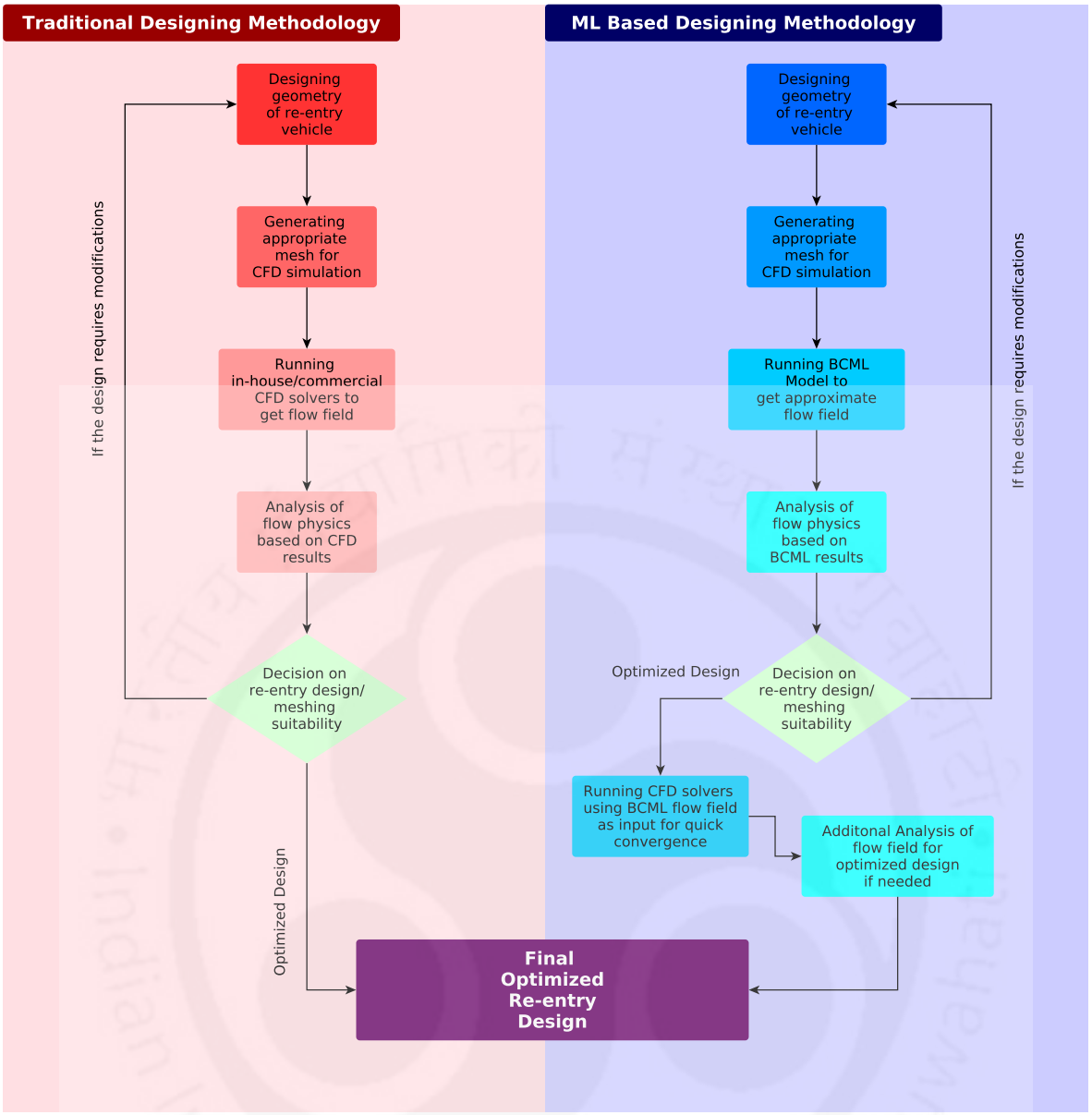

Figure 1.3: Flowchart describing the contrast between traditional and ML-based designing philosophy.

investigate data-driven frameworks as efficient surrogate models for rapid flow-field prediction across both continuum and rarefied regimes.

The utility of the proposed BCML model is to aid in the design of re-entry vehicles by reducing overall computational cost and dependence on conventional simulation techniques, rather than replacing them. Traditionally, a researcher designs the features of an object, in our case, the re-entry vehicle geometry. Following this, an appropriate domain and mesh are constructed to obtain flow-field features using computational gas-dynamics solvers. Since the wave features in the flow domain are not known a priori, the first set of meshes is coarser. Based on the final converged results of the computational gas dynamics solver, the mesh is updated to resolve the shock waves and capture other wave features more accurately. Once the flow field results for the final optimised mesh are obtained, a

decision must be made on the re-entry vehicle design. Overall, designing is an iterative process that includes several simulations, which are often time-consuming. In contrast to the traditional method, the ML-based design methodology for a re-entry vehicle uses a new BCML model to obtain an approximate flow-field solution at a fraction of the computational cost. These BCML-generated flow fields are used to obtain a final optimised mesh and/or an updated design. Computational gas-dynamics solvers are used to obtain accurate flow-field results on the optimised mesh/design. Furthermore, estimated flow-field solutions from the BCML can be used as initial conditions to improve convergence. This significantly reduces the computational time and helps design the re-entry vehicle geometry. Overall, the proposed ML-based designing methodology employs BCML-assisted CFD algorithms, which are summarised in Figure 1.3.

1.4 Organisation of the Thesis

Chapter II presents a comprehensive review of the existing literature on data-driven fluid dynamics, with particular emphasis on machine learning–based approaches for flow-field prediction and mesh generation, and their relevance to incompressible, compressible, hypersonic, and rarefied gas flows. Chapter III outlines the computational methodology adopted in this work. This chapter discusses the governing equations solved by the continuum-based solver and the DSMC algorithm. In addition, the overall idea and the implementation of the boundary condition–based machine learning (BCML) algorithm are discussed in detail. The idea of the BCML algorithm is tested on a simple lid-driven cavity problem to assess preliminary feasibility.

Chapter IV focuses on the development of the BCML algorithm for low-altitude flow regimes (40–60 km), where the continuum assumption holds. This chapter presents the details of the NSF-based CFD simulations, including ambient conditions, physical and numerical model parameters, and the training of machine learning models under both frozen and reacting flow conditions. A comprehensive evaluation of the proposed DNN model for chemical reaction, named the ChemDNN model, is conducted to assess its ability to predict species mole fractions. The analysis quantifies model accuracy and generalisation across varying flow conditions, demonstrating reliable representation of complex chemical kinetics. Similarly, Chapter V presents the development of the BCML algorithm for high-altitude flow regimes (60–84 km), where rarefaction effects become significant. This chapter details the DSMC simulations, ambient conditions, model parameters, and the corresponding machine learning model training. In Chapters IV and V, a detailed validation of the predicted flow fields using the developed BCML models is presented.

Chapter VI demonstrates additional applications of the BCML-predicted flow fields to accelerate conventional CFD and DSMC simulations, highlighting the potential computational savings and practical applicability of the proposed framework. Finally, Chapter VII provides a summary and conclusion of the thesis, summarises the developed low and high-altitude BCML models, and offers a critical analysis and an examination of the future prospects of the BCML framework.

Chapter 2

Review of Literature

In recent years, data-driven solutions have accelerated various numerical approaches across a wide range of fields, from image classification to computer vision [35, 36], from speech recognition [21] to autonomous driving [37] and healthcare [38]. Moreover, advances in high-performance computing, especially with the proliferation of both Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), along with the availability of open-source software, have significantly reduced the cost of generating the necessary data. In recent times, machine learning (ML) algorithms have gained much prominence in scientific computing and data analysis [39] in fluid dynamics [16, 40, 41]. A variety of ML approaches such as deep neural networks (DNN) [42–44], physics-informed neural networks (PINN) [24, 26, 45–48], reduced order modelling (ROM) [49], principal component analysis (PCA)/proper orthogonal decomposition (POD) [50–53] have been employed to model various fluid dynamics applications such as flow field reconstruction, flow optimisation and control, and turbulence closure modelling [54]. ML models have also been extensively applied to optimisation problems in experimental fluid dynamics [16].

A comprehensive review of existing research relevant to the objectives of the thesis is presented here. The review of fundamental theories [4, 55–57] and various computational methods [58–65] for modelling complex hypersonic and non-equilibrium flows is omitted here, since the present survey is confined to existing studies related to ML algorithms in flow prediction, particularly in high-speed and hypersonic flow regimes. The ML algorithms in this literature survey are classified by flow type: incompressible and compressible. Recent ML-related research works on hypersonic flows have been included in the latter. In addition, various data-driven modelling approaches for mesh generation have been discussed in detail. This is mainly because one of the applications of the new model, the boundary condition-based machine learning (BCML) model reported in this work, focused on mesh optimisation, particularly for the particle-based DSMC method. The chapter concludes with a discussion of existing research gaps and unresolved challenges, along with the research objectives pursued in this thesis.

2.1 Machine Learning Approaches for Modelling Incompressible Fluid Flows

Sekar et al. [17] proposed a combination of ML techniques to predict the flow field over an arbitrarily shaped airfoil for incompressible laminar flows. The first part of the model consisted of a convolutional neural network (CNN) that parametrised the airfoil shape from an image into a set of 16 parameters. It was reported that the CNN achieves 99.5% accuracy on training airfoils and 99% on test airfoils. The second component of the model was a deep neural network (DNN), which used the parameters generated by the first component as input to predict the flow field over airfoils. In addition to the shape parameters obtained from CNN, the Reynolds number (Re) of the freestream flow, the angle of attack of the flow, and the coordinates of an arbitrary point in the flow domain form the input layer of the DNN. The corresponding flow-field variables, such as pressure and velocity components, form the DNN's output layer. Sufficient flow field data for a wide range of flow conditions and a wide range of airfoil shapes were used to train such a model. It was concluded that the trained model achieved more than 99% accuracy on the training cases and more than 97% on the test cases. Sekar et al. [17] showed, through this research, that deep learning-based neural networks can learn and approximate Navier–Stokes solutions with reasonable accuracy for geometries not explicitly included in the training data.

Jin et al. [66] developed a data-driven modelling approach for predicting the velocity flow field in the vicinity of a circular cylinder using a fusion CNN with input pressure fields. For this purpose, the Reynolds number is varied from 60 to 1100 to train and test the dataset and model. A fusion CNN novel architecture is designed for this purpose. This CNN has two paths: one with a pooling layer and another without. This kind of architecture can capture spatial and temporal information, as well as features invariant to small translations, from the pressure-fluctuation series on the cylinder. Numerical simulations at various Reynolds numbers are performed to obtain training and testing data. The velocity and vorticity predicted by the CNN model agree very well with those from CFD. The model can only learn the intrinsic flow features within the Reynolds number range represented in the training dataset. It was also observed that vortex shedding does not occur at $Re = 10$, and therefore the underlying flow physics associated with this regime are absent from the training dataset spanning $Re = 70$–900. Consequently, the CNN model developed by Jin *et al.* [66] was unable to accurately predict the flow field for the $Re = 10$ case.

Ribeiro et al.[67] developed a CNN-based model, known as *DeepCFD*, which provides an approximate solution of two-dimensional non-uniform steady velocity and pressure fields for internal laminar flows with arbitrarily shaped bluff bodies. The CNN-based model was about three orders of magnitude faster than the conventional CFD simulations. For the flow field analysis over a circular object, the maximum absolute error in velocity magnitude and

pressure between the CNN-based model and CFD solutions was less than 1%.

In recent years, several studies have employed ML for turbulence modelling, primarily to develop data-driven closure models. Ling and Templeton [19] used various ML algorithms, including support vector machines (SVMs), AdaBoost-based decision trees, and random forests, to estimate the flow field for subsonic airfoil configurations governed by the Reynolds-averaged Navier-Stokes (RANS) equations. The model calculated the lift, drag, and skin-friction coefficients for turbulent flow conditions. A mapping function was also investigated between the turbulent eddy viscosity and the mean flow variables using a neural network.

Guastoni et al. [68] investigated CNN–based models for predicting two-dimensional instantaneous velocity fluctuation fields in turbulent open-channel flows. The models employed wall shear stress components and wall pressure as input features and were trained using direct numerical simulation (DNS) data corresponding to friction Reynolds numbers of 180 and 550. The results demonstrated that CNN-based approaches outperformed extended proper orthogonal decomposition methods in reconstructing velocity fields, highlighting the effectiveness of deep learning techniques for predicting wall-bounded turbulent flows.

Wang et al. [20] developed the subgrid-scale (SGS) model for large-eddy simulations (LES). It was reported that their neural network (NN) model provided a better solution for this regression problem. The model was applied to the convolution and deconvolution of coarse-grained fields, emphasising sub-grid-scale turbulence effects in LES, which was also discussed in detail.

Duraisamy et al. [54] reported that in fluid mechanics, a large amount of experimental and direct numerical simulation data for turbulence modelling is available, and the application of RANS equations yields solutions for small-scale engineering models using these data. However, there are some discrepancies and uncertainties regarding the solution. The use of ML for reducing these uncertainties associated with RANS equations and quantifying data has been studied.

2.2 Machine Learning Approaches for Modelling Hypersonic Flows

Mao et al. [69] aimed to develop a DNN model for solving coupled flow and finite-rate chemistry in hypersonic flows. They built a specialised neural network, *DeepONet*, to approximate non-linear operators. This new framework, called *DeepM&MNet*, was developed to address

multiphysics and multi-scale (M&M) problems, such as hypersonic flows. They predicted five species (O_2, N_2, N, O, NO) in non-equilibrium chemistry downstream of a normal shock in hypersonic flows. Their results showed that *DeepONet* can be up to five times faster than conventional CFD solvers while maintaining a promising level of accuracy. The trained model can generate solutions in a fraction of a second.

Ivan et al. [29] introduced a model to analyse rovibrational relaxation and dissociation in gas mixtures, specifically applied to an O_2–O system. Their ML approach significantly outperforms traditional numerical integration methods, achieving a nearly two orders of magnitude speed-up while maintaining a relative error of less than 2%. This advancement represents a promising step toward integrating efficient ML and coarse-grained (CG) surrogate models with CFD simulations to better capture thermochemical non-equilibrium phenomena. The study demonstrates that the Physics-Informed DeepONet model delivers reliable accuracy, with relative L_2 errors below 2%, while achieving computational speeds that far exceed those of conventional numerical simulations.

Novello et al. [70] designed a hybrid numerical scheme that combines a traditional fluid dynamics solver with a neural network for faster predictions of fluid flows. They simulated a hypersonic planetary re-entry problem under chemical equilibrium conditions and employed a supervised neural network with five hidden layers within this framework. The problem was tested for four different cases: a toy problem (3 species), Earth (18 species), Jupiter (36 species), and cloudy Jupiter (64 species). The reported L_2 errors were 7.59×10^{-8}, 1.55×10^{-7}, and 4.99×10^{-7} for the toy, Earth, and Jupiter cases, respectively. These results demonstrate the potential of neural networks to provide reliable predictions for production-level simulations.

Monti et al. [71] proposed a technique to construct a physics-constrained neural network that introduces corrections to the constitutive relations in the Navier–Stokes model. Their primary focus was on enhancing deep learning's computational capabilities for predicting hypersonic flows. The model is designed to be general-purpose, as it is independent of specific cases or geometries. The test case considered was a 1D shock in a low-pressure argon flow. The model was trained for three freestream Mach numbers ($M = 2, 5, 8$) and tested for $M = 3, 4, 6, 7, 9$. The results demonstrate the potential of deep learning approaches to improve the accuracy of Navier–Stokes predictions for rarefied gas flow simulations, providing precise solutions at a cost comparable to solving the computationally affordable Navier–Stokes equations.

Recently, Dai et al. [72] have reported a positional encoder-based PINN model (PE-PINN) that could accurately capture large gradients present in the shock regions of the supersonic flow fields with a maximum relative error of less than 1%. However, the current version of the

PE-PINN model does not include effects from chemical reactions, multi-species interactions, and non-equilibrium conditions. It is currently limited to continuum compressible flow modelling based on the Navier–Stokes-Fourier equations.

ML algorithms have also handled rarefied flows encountered by re-entry vehicles at higher altitudes. Schouler et al. [73] reported an ML model that predicts aerodynamic and aero-thermodynamic coefficients for re-entry vehicle surfaces in hypersonic, rarefied conditions, which plays a vital role in design. Zhang et al. [74] addressed the practical difficulties in lunar landing space explorations. The direct simulation Monte Carlo (DSMC) [75], a traditional method for simulating this problem, is often computationally intensive. Hence, they developed a CNN-based DSMC method for a three-dimensional DSMC simulation model to address challenges in lunar space exploration. This CNN-DSMC-3D model accelerates the convergence rate by four to six times compared to the conventional DSMC algorithm.

Yao et al. [76] proposed a CNN-based data-driven non-linear constitutive relation (DNCR) for non-equilibrium rarefied flows. The training data for the DNCR-CNN-based model were obtained from finite-volume method (FVM)- based Navier–Stokes solvers and the unified gas-kinetic scheme (UGKS) solver. The DNCR model mapped UGKS stress and heat flux to the flow field conditions and the local Knudsen number. Yao et al. simulated the one-dimensional shock structure problem and the two-dimensional flow over a cylinder at rarefied supersonic conditions. They found that the results from the DNCR model matched those obtained from the UGKS simulations.

Nair *et al.* [77] developed a DNN-based closure framework to assist the conventional NSF equations for rarefied flow applications. Their model predicts non-equilibrium stress tensors and heat fluxes from the training against high-fidelity kinetic data from Boltzmann and direct simulation Monte Carlo (DSMC) solutions. The framework was validated for a wide range of problems, such as 1D shock structure profiles and hypersonic transitional flows over sharp geometries. Their work demonstrates that data-driven closures can accurately capture non-local rarefaction and slip effects, offering a highly efficient alternative to particle methods in the slip and early transitional regimes.

Garg et al. [78, 79] integrated DNN-based ML models in a finite volume method (FVM) implementation of the Navier-Stokes-Fourier equations to replicate DSMC-level predictions of high Mach-number shock structures in gases. Constitutive relations mapping the viscous stress tensor and heat flux with first-order gradients of velocity and temperature were first constructed by training a DNN model on the DSMC data for the monoatomic gases [78]. The approach was then extended to diatomic gases using both two-temperature (translational and rotational) and three-temperature (translational, rotational, and vibrational) frameworks [79], with DNN models developed for the viscous stresses and heat fluxes associated with these

energy modes. The resulting FVM–DSMC–ML solver reproduced shock-structure solutions with high accuracy while achieving a substantial reduction in computational cost. The ML-integrated FVM code required approximately one-tenth of the computational resources compared to conventional DSMC simulations.

2.3 Machine Learning–Augmented Mesh Generation Algorithms for Fluid Dynamics

The generation of unstructured meshes for conventional CFD algorithms is a well-developed procedure and has been widely reported in the literature in the last few decades. Baker et al. [80] introduced a method for creating tetrahedral meshes based on a Delaunay triangulation algorithm. The approach is designed to manage objects of varying levels of complexity. To preserve the surface triangulation of solid objects, the Delaunay property may need to be modified, with adjustments made to the mesh when new points are added near solid boundaries. Chew et al. [81] defined the constrained Delaunay triangulation (CDT) for a set of n planar vertices and a collection of non-intersecting straight-line segments as a triangulation that contains all prescribed segments as edges, while satisfying the Delaunay empty-circumcircle property wherever these constraints do not violate it. Reichert et al. [82] explored the mesh generation process for two-dimensional problems and introduced a novel mesh generator that uses expert-system techniques. The algorithm examines the corner-angle conditions at each vertex of a closed polygon to generate different types of mesh primitives. During each step of the mesh generation, potential conflicts with the existing mesh are identified, and the best solution is chosen. This approach allows for successful coverage of any polygonal region, even those with complex boundaries, using a base mesh. The initial mesh is then improved with side-swapping and node-shifting algorithms, which refine both element shapes and node positions.

Papagiannopoulos et al. [83] proposed an ML-based scheme for generating 2D simplicial meshes assisted by neural networks. The neural networks are trained on data extracted from meshed contours, enabling them to approximate the number of vertices to add within the contour cavity, along with their positions and connectivity. The accuracy of the approach is assessed by comparing the mesh quality produced by the neural networks against that of a reference mesher. Using an element quality metric, tests on contours with varying edge counts reveal a maximum average deviation of 15.2% in mean quality and 27.3% in minimum quality between the meshes generated by the proposed scheme and those from the reference mesher. Despite these differences, the scheme can produce meshes suitable for meshing tasks. Additionally, it is extended to generate larger-scale meshes using a recursive implementation, with results suggesting its potential adaptation for 3-D mesh generation.

Song et al. [84] applied ML to optimise unstructured oceanographic meshes using K-means clustering, enhancing mesh quality by minimising long, narrow triangles. Lu et al. [85] proposed an ML-driven advancing double-front method (ADFM) to automate isotropic triangular mesh generation, significantly improving meshing efficiency.

2.4 Objectives of Thesis

Recent years have witnessed remarkable progress in the application of machine learning to computational fluid dynamics, particularly for incompressible flows, laminar flow prediction, turbulence modelling, reduced-order modelling, and data-driven closure formulations. Nevertheless, compared with incompressible flow applications, there remains considerable scope for the development of machine learning approaches for compressible, hypersonic, and chemically reacting flows, where strong shock waves, thermochemical non-equilibrium, and multi-scale physical phenomena pose significant challenges. The objective of the present work is to introduce the concept of a new ML approach, the **Boundary Condition-based Machine Learning** (BCML) model, that can be used to predict the flow field across a wider range of flow conditions and for arbitrary geometries not included in the training data. The utility of the BCML model extended beyond fast prediction of flow-field data and could be employed to accelerate convergence in FVM-based NSF solvers and optimise DSMC simulations. For the latter, it was found that meshes generated by engineers using traditional CFD codes or commercial software are often unsuitable for DSMC simulations. Towards this, a mesh-generation algorithm that uses the BCML-generated flow field can improve the computational efficiency of DSMC simulations. The overall objectives of the present thesis are as follows:

- To develop a physics-assisted deep neural network-based machine learning model that incorporates the effect of boundary conditions on the flow physics and evaluate its performance for the incompressible lid-driven cavity problem.
- To develop the boundary condition-based machine learning (BCML) model for predicting flow re-entry vehicles at lower altitudes, trained using data generated using the Navier-Stokes-Fourier-based CFD solvers (frozen and reacting solvers).
- To extend the concept of the BCML model-based framework for complete analysis of non-equilibrium hypersonic flows (flow prediction, chemistry modelling, and estimating surface properties such as heat flux) at higher altitudes using direct simulation Monte Carlo (DSMC) simulations.
- To apply BCML-based flow-field reconstruction to accelerate numerical simulations:

– To evaluate the ability of BCML-based flow fields to serve as effective initial conditions that accelerate the convergence of NSF-based CFD simulations.

– To generate optimised meshes using BCML-based flow fields for efficient DSMC simulations.

Chapter 3

Computational Methodology

Data generation is central for any supervised machine learning (ML) algorithm. The objective of this work is to develop a BCML model for hypersonic flows over arbitrarily shaped bluff bodies across a wide range of altitudes. The altitude range spans 40 km to 84 km. This range is of interest for most reentry applications since the vehicle experiences intense heat and extremely high temperatures. Additionally, the reentry vehicles endure severe deceleration in the 40-84 km range as the atmosphere acts as a brake. Also, the plasma sheath forming around the reentry vehicle causes a radio blackout. Understanding the flow physics in this altitude range is essential for mission control, else the reentry vehicle might burn up or skip back into space.

The altitude range of interest in the present thesis spans the continuum and transition regimes of the flow. For the lower altitudes (40-60 km), the continuum-based solvers, specifically the Navier–Stokes–Fourier (NSF) equations, are employed to generate training datasets. The first low-altitude BCML framework was reported for reentry flows under frozen-flow assumptions, using an in-house finite-volume method-based code. This model was then extended to the non-equilibrium NSF equations, enabling the generation of training data for reacting flows within the same low-altitude framework. A comprehensive description of the governing equations, including the associated transport property models, mixture rules, chemical reaction modelling, and non-equilibrium models, along with details of the `OpenFOAM`-based `hy2Foam` solver, is provided in this chapter.

At higher altitudes (60-84 km), the continuum assumption is no longer valid, and using first-order accurate NSF equations is not appropriate. At these altitudes, the DSMC method is employed. The fundamental principles and numerical implementation of the DSMC method, along with its underlying computational characteristics, are also discussed in this chapter. The DSMC framework is employed to produce training data for the high-altitude BCML model under both frozen and chemically reacting flow conditions.

The BCML algorithm, the concept of generators, and the framework's overall workflow are also introduced. To begin with, the BCML concept was tested on a simple incompressible lid-driven cavity, and the ML model results were compared with those reported in the literature. This exercise was mainly done to gain confidence in the validity of the BCML idea.

3.1 Navier-Stokes-Fourier Equation-based Continuum Solvers

The training data for the low-altitude BCML framework is generated using the finite volume method-based continuum solvers. Continuum solvers assume the gas can be treated as a continuous medium, which is valid when the local mean free path (λ) is much smaller than the characteristic flow length scale (L). Under this assumption, macroscopic flow variables such as density, velocity, and temperature are defined at every spatial location and evolve according to conservation laws. These solvers are widely employed for low-altitude and high-density flow regimes, where intermolecular collisions are sufficiently frequent to maintain local thermodynamic equilibrium. The classical Navier–Stokes (NS), or more appropriately, the Navier–Stokes–Fourier (NSF) formulation, describes the conservation of mass, momentum, and total energy for viscous compressible flows, accounting for molecular transport effects through first-order constitutive relations for stress and heat flux. In their conventional form, the NSF equations assume a single temperature field and equilibrium thermodynamic behaviour, making them suitable primarily for near-equilibrium flows. In non-equilibrium applications, the NSF equations are further generalised to include separate temperature fields for different energy modes (translational–rotational and vibrational–electronic modes), along with species transport and chemical source terms. This multi-temperature NSF framework is particularly important for hypersonic and high-enthalpy flows, where thermal and chemical non-equilibrium effects become significant.

3.1.1 Single-temperature NSF Formulation

The single-temperature Navier-Stokes-Fourier (NSF) equations describe the behaviour of viscous compressible flows. The conservative form of the three-dimensional NSF system can be written compactly by introducing the conserved state vector U, which consists of density (ρ), momentum components ($[\rho u, \rho v, \rho w]$), and total energy ($E = \rho e$), along with the corresponding flux vectors $F(U)$, $G(U)$, and $H(U)$ in the x, y, and z directions, respectively, as presented below:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)_I}{\partial x} + \frac{\partial G(U)_I}{\partial y} + \frac{\partial H(U)_I}{\partial z} = \frac{\partial F(U)_V}{\partial x} + \frac{\partial G(U)_V}{\partial y} + \frac{\partial H(U)_V}{\partial z} \tag{3.1}$$

where the suffixes I and V indicate inviscid and viscous components respectively and

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{bmatrix}, \; F(U)_I = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho e + p) \end{bmatrix}, \; G(U)_I = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\rho e + p) \end{bmatrix}, \; H(U)_I = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(\rho e + p) \end{bmatrix}$$

$$F(U)_V = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix}, \quad G(U)_V = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \tag{3.2}$$

$$H(U)_V = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix}$$

The total energy density (e), comprising the specific internal energy (IE) and the kinetic energy part, and H is the total specific enthalpy given by

$$e = \frac{u^2}{2} + IE, \quad IE = \frac{p}{(\gamma - 1)\rho}, \quad H = \frac{e + p}{\rho} \tag{3.3}$$

where γ denotes the ratio of specific heats

$$\gamma = \frac{C_p}{C_v} \tag{3.4}$$

The sound speed is defined as

$$a = \sqrt{\frac{\gamma p}{\rho}} \tag{3.5}$$

Assuming Stokes' hypothesis for a Newtonian fluid and Fourier's law of heat conduction, the stress and heat flux terms in the viscous flux vectors can be expressed as first-order approximations,

$$\tau_{xx} = \mu \left\{ \frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}) \right\}, \tau_{yy} = \mu \left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}) \right), \tau_{zz} = \mu \left(\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \right),$$
$$\tau_{xy} = \tau_{yx} = \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}), \tau_{yz} = \tau_{zy} = \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}), \tau_{zx} = \tau_{xz} = \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}),$$
$$q_x = \kappa\frac{\partial T}{\partial x}, q_y = \kappa\frac{\partial T}{\partial y}, q_z = \kappa\frac{\partial T}{\partial z} \tag{3.6}$$

where μ is the coefficient of dynamic viscosity and κ is the thermal conductivity.

The in-house three-dimensional, message-parsing interface (MPI) based FVM solver employs a wide range of inviscid and viscous flux solvers. In the present work, the Harten-Lax-van Leer-Contact (HLLC) is used to model the convective fluxes. The HLLC scheme is

an approximate Riemann solver specifically designed to handle contact discontinuities and shock waves using a three-wave solution structure. Figure 3.1 illustrates the HLLC wave configuration for the solution.

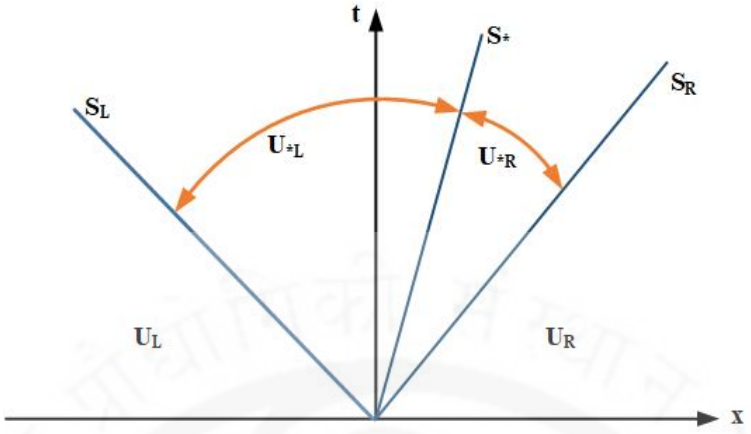

Figure 3.1: Illustration of waves in the HLLC scheme.

In the HLLC scheme, S_L and S_R represent speeds of the slowest and fastest waves across the Riemann problem between the left (U_L) and the right states (U_R), respectively. The acoustic waves (S_L and S_R) are calculated using Einfeldt wave speed estimates:

$$S_L = \min(u_L - a_L, \tilde{u} - \tilde{a}),\ S_R = \max(u_R + a_R, \tilde{u} + \tilde{a}) \tag{3.7}$$

where

- u_L, u_R: Normal velocities of the left and right states.
- a_L, a_R: Local speed of sound in the left and right states.
- $\tilde{u}, \tilde{a}$: Roe-averaged velocity and sound speed.

In addition to the S_L and S_R, a middle wave representing the contact discontinuity with S_* speed is given by:

$$S_* = \frac{p_R - p_L + \rho_L u_L(S_L - u_L) - \rho_R u_R(S_R - u_R)}{\rho_L(S_L - u_L) - \rho_R(S_R - u_R)} \tag{3.8}$$

Together with the acoustic waves, the three waves in the HLLC method divide the space-time $(x - t)$ domain into four regions (U_L, U_{*L}, U_{*R} and U_R). Pressure and velocity remain constant ($p_{*L} = p_{*R} = p_*, u_{*L} = u_{*R} = u_* = S_*$) across the contact discontinuity. By applying Rankine-Hugoniot conditions across each of the waves, the flux for the intermediate portions (U_{*L}, U_{*R}) of the space-time domain is given as

$$F_{*L} = F(U_L) + S_L(U_{*L} - U_L),\quad F_{*R} = F(U_R) + S_R(U_{*R} - U_R) \tag{3.9}$$

where U_{*L} and U_{*R} are given by:

$$U_{*L/*R} = \begin{bmatrix} \rho(\frac{S_{*L/*R}-u_{L/R}}{S_{*L/*R}-S_*}) \\ \rho S_*(\frac{S_{*L/*R}-u_{L/R}}{S_{*L/*R}-S_*}) \\ \rho(\frac{S_{*L/*R}-u_{L/R}}{S_{*L/*R}-S_*}) \left(e + (S_* - u_{L/R})(S_* + \frac{P_{*L/*R}}{\rho(S_{*L/*R}-u_{L/R})})\right) \end{bmatrix} \tag{3.10}$$

Thus, the formulation for the flux vector at the intercell is given as

$$F^{hllc} = \begin{cases} F(U_L) & \text{if } 0 \leq S_L \\ F_{*L} & \text{if } S_L \leq 0 \leq S_* \\ F_{*R} & \text{if } S_* \leq 0 \leq S_R \\ F(U_R) & \text{if } 0 \geq S_R \end{cases} \tag{3.11}$$

The set of discretised equations is solved explicitly using the Euler time integrator with an appropriate time step (Δt). The maximum wave speed is obtained for all cells ($i \in N_{cell}$) in every iteration as

$$S_i = |u_i| + a_i, \quad S_{max} = \max(S_i) \tag{3.12}$$

The inviscid time step (Δt_I) is governed by the Courant-Friedrichs-Levy (CFL) condition, which employs the maximum wave speed. The viscous time step (Δt_V) is governed by the diffusion of momentum. The time step chosen for an iteration is given by the minimum of the two, as given by:

$$\Delta t_I = \frac{\Delta x \; CFL}{S_{max}}, \quad \Delta t_V = \frac{\rho \Delta x^2 \; CFL}{2\mu} \tag{3.13}$$

$$\Delta t = \min(\Delta t_I, \; \Delta t_V) \tag{3.14}$$

The in-house code has been extensively validated. The simulation of the monoatomic shock structure of argon gas (Ar) is reviewed here, as analytical results are available. The initial conditions on the driver's side are characterised by the upstream Mach number (M_1), temperature (T_1), and pressure (P_1). Using the Rankine Hugoniot conditions [86], the post-shock conditions on the driven side, namely M_2, T_2, and P_2, are calculated from the given set of initial conditions.

$$M_2^2 = \frac{1 + (\frac{\gamma-1}{2})M_1^2}{\gamma M_1^2 - (\frac{\gamma-1}{2})}, \qquad \frac{\rho_2}{\rho_1} = \frac{u_1}{u_2} = \frac{(\gamma+1)M_1^2}{2 + (\gamma-1)M_1^2} \tag{3.15}$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1}(M_1^2 - 1), \qquad \frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)\left(\frac{\rho_1}{\rho_2}\right) \tag{3.16}$$

The mean free path is defined based on the pre-shock conditions [87]

$$\lambda_1 = \frac{\sqrt{\frac{\pi}{2}}\mu_{ref}\left(\frac{T_1}{T_{ref}}\right)^{\omega}}{\rho_1\sqrt{RT_1}} \tag{3.17}$$

where ω is the viscosity index in the power law for calculating viscosity

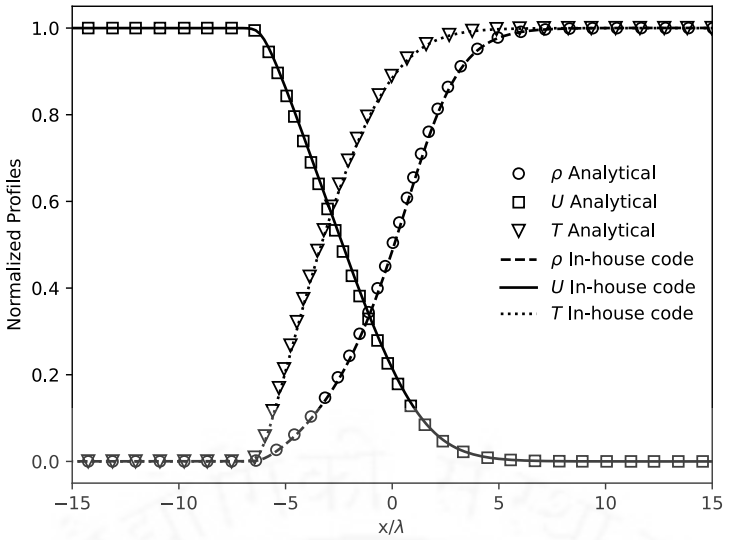

Figure 3.2: Comparison plots of normalized profiles of density, temperature, and velocity obtained at Mach 10 from analytical expression and from the in-house FVM code employing the HLLC scheme.

The one-dimensional simulation domain is taken as a multiple of the mean free path, $L = 60\lambda_1$ with a total of 1000 equi-sized cells. From Figure 3.2, it is clear that the results ($M_1 = 10$) generated by the in-house FVM code employing the HLLC flux scheme closely match the exact solution of the Navier-Stokes-Fourier (NSF) equations. The in-house FVM code is capable of solving three-dimensional gas flows using both unstructured (tetrahedral, pyramidal and triangular prism) and structured (hexahedral) cells.

3.1.2 Multi-temperature NSF Formulation

Hypersonic flows with chemically reacting effects in the continuum regime are commonly modelled using the two-temperature, non-equilibrium Navier–Stokes–Fourier (NSF) formulation. In the present work, `OpenFOAM`-based `hy2Foam` solver [88] was used to generate the training data for a reacting (chemical and ionisation reactions) gas mixture consisting of N_s species. The corresponding system of governing equations solved in the non-equilibrium NSF solver can be expressed as follows:

$$\frac{\partial U}{\partial t} + \frac{\partial F_{inv,i}}{\partial x_i} - \frac{\partial F_{vis,i}}{\partial x_i} = \dot{W} \tag{3.18}$$

where, the conserved quantities vector, U, is defined as

$$U = (\rho, \rho_s, \rho u, \rho v, \rho w, E_{ve}, E)^T \tag{3.19}$$

here $s \in N_s$, u, v, and w denote the velocity components along the x, y, and z directions, respectively. The symbol ρ represents the total mass density of the mixture, while ρ_s corresponds to the density of species s. The overall flux vector is decomposed into inviscid ($F_{inv,i}$) and viscous ($F_{vis,i}$) contributions, as outlined below in Einstein's notation:

$$F_{\text{inv},i} = \begin{pmatrix} \rho u_i \\ \rho_s u_i \\ \rho u_i u + \delta_{i1} p \\ \rho u_i v + \delta_{i2} p \\ \rho u_i w + \delta_{i3} p \\ E_{ve} u_i \\ (E + p) u_i \end{pmatrix} \tag{3.20}$$

$$F_{\text{vis},i} = \begin{pmatrix} 0 \\ J_{s,i} \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ -q_{ve,i} - e_{ve} J_{n,i} \\ \tau_{ij} u^j - q_{tr,i} - q_{ve,i} - \sum_{r \neq e} h_r J_{r,i} \end{pmatrix} \tag{3.21}$$

where the subscript i refers to one of the three spatial directions, and δ denotes the Kronecker delta. The quantities E and E_{ve} represent the total energy and the vibro-electronic energy, respectively. The variable h corresponds to the specific enthalpy of species s, J denotes the diffusion flux, and τ represents the shear stress tensor. The pressure p is evaluated using Dalton's law of partial pressures, given by:

$$p = \sum_{s \neq e} p_s + p_e = \sum_{s \neq e} (\rho_s R_s T_{tr}) + \rho_e R_e T_e \tag{3.22}$$

where R_s is the specific gas constant, T_{tr} is the trans-rotational component temperature, ρ_e is the electron density, (T_e) is the electron temperature, which is set equal to the vibro-electronic temperature (T_{ve}).

In Equation (3.21), τ_{ij} represents the component of the viscous stress tensor, which can be written as follows:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + (\lambda + \mu_b) \frac{\partial u_k}{\partial s_k} \delta_{ij} \tag{3.23}$$

where, μ is the shear viscosity of the mixture, λ is the second coefficient of viscosity of the mixture and μ_b is the mixture bulk viscosity. `hy2Foam` employs Stokes's hypothesis, which results in the $\mu_b = 0$ and the second viscosity is given by $\lambda = -\frac{2}{3}\,\mu$. `hy2Foam` employs multiple models where the shear viscosities can either be considered as temperature-independent, or set to follow a power law or Blottner's curve fits [89]. The corresponding expression for the temperature-dependent models (Power law and Blottner's curve fitting model) for species s is given by:

$$\mu_s = \mu_{ref,s} \left(\frac{T_{tr}}{T_{ref}} \right)^{\omega_s} \tag{3.24}$$

and

$$\mu_s = 0.1 \times \exp\left((A_{B,s} \ln(T_{tr}) + B_{B,s}) \ln(T_{tr}) + C_{B,s} \right) \tag{3.25}$$

where, ω_s denotes the viscosity exponent for species s, T_{ref} represents the reference temperature ($T_{ref} = 273$ K in present work), and $A_{B,s}$, $B_{B,s}$, and $C_{B,s}$ are the Blottner curve-fit coefficients.

The spatial components of the heat conduction vector, for the trans-rotational component, and the vibro-electronic component, are assumed to follow Fourier's law.

$$q_{tr,i} = -\kappa_{tr} \frac{\partial T_{tr}}{\partial x_i} \tag{3.26}$$

where κ_{tr} is the mixture thermal conductivity and

$$q_{ve,i} = -\kappa_{ve} \frac{\partial T_{ve}}{\partial x_i} \tag{3.27}$$

where $\kappa_{ve,s}$ is the mixture thermal conductivity for the vibro-electronic component of energy. Similar to the modelling of the coefficient of viscosity in `hy2Foam`, the thermal conductivity $\kappa_{tr,s}$ and $\kappa_{ve,s}$ can be considered as temperature-independent, or set to follow Eucken's relation [90, 91]. The trans-rotational and vibro-electronic thermal conductivities for an individual species s are given by

$$\kappa_{tr,s} = \frac{5}{2}\,\mu_s C_{v_{t,s}} + \mu_s C v_{r,s} \quad \kappa_{ve,s} = 1.2\,\mu_s C v_{ve,s} \tag{3.28}$$

where $C_{v_{t,s}}$, $C_{v_{r,s}}$ and $C_{v_{v,s}}$ are the specific heats at constant volume for the translational, rotational and vibrational degrees of freedom, respectively. Alternatively, transport coefficients may be evaluated using collision cross-section data derived from Gupta's curve-fit models [92]. Compared to the Blottner and Eucken formulations, Gupta's method is reported to provide improved physical fidelity for temperatures exceeding 10,000 K, particularly in weakly ionised flow regimes [93].

The overall transport properties of the gas mixture are calculated using Wilke's mixing rules [94] or the Armaly-Sutton model [95, 96] in the `hy2Foam` solver. Wilke's mixing rule is given by:

$$\mu = \sum_s \frac{\mu_s X_s}{\phi_s}, \qquad s \in N_s \tag{3.29}$$

where X_s denotes the molar fraction and ϕ_s is a scaling factor defined as

$$\begin{aligned} \phi_s = X_s + \sum_{r \neq s} X_r \left[1 + \sqrt{\frac{\mu_s}{\mu_r}} \left(\frac{\mathcal{M}_r}{\mathcal{M}_s}\right)^{1/4}\right]^2 \\ \times \left[\sqrt{8\left(1 + \frac{\mathcal{M}_s}{\mathcal{M}_r}\right)}\right]^{-1}, \qquad r \in N_s \end{aligned} \tag{3.30}$$

A similar formula can be applied to thermal diffusivity. Palmer and Wright [93] reported that the accuracy of Wilke's mixing rule model deteriorates for ionised gas mixtures and for thermodynamic states beyond this temperature range. The mixing rule proposed by Armaly and Sutton [95] addresses this limitation and has been demonstrated to yield improved agreement with the multicomponent viscosity formulation for ionised gas mixtures. The scaling factor ϕ_s appearing in Eq. 3.29 is modified according to the following expression:

$$\begin{aligned}\phi_s = X_s + \sum_{r \neq s} X_r \left[\frac{5}{3\,\mathcal{A}^*_{sr}} + \frac{\mathcal{M}_r}{\mathcal{M}_s}\right]\left[1 + \frac{\mathcal{M}_r}{\mathcal{M}_s}\right]^{-1} \\ \times \left[1 + \mathcal{B}_{sr}\sqrt{\frac{\mu_s}{\mu_r}\left(\frac{\mathcal{M}_r}{\mathcal{M}_s}\right)^{1/4}}\right]^2 \left[\sqrt{8\left(1 + \frac{\mathcal{M}_s}{\mathcal{M}_r}\right)}\right]^{-1}, \qquad r \in N_s\end{aligned} \tag{3.31}$$

where $\mathcal{A}$ and $\mathcal{B}$ are various modelling parameters for the pair of species (s and r species).

The source term vector, $\dot{W}$, in the governing equation is written as

$$\dot{W} = (0,\ \dot{\omega}_s,\ 0,\ 0,\ 0,\ \dot{\omega}_{v,m},\ 0)^T \tag{3.32}$$

where $\dot{\omega}_s$ is the net mass production of species s relevant for the species equation and $\dot{\omega}_v$ is the source term representing the rate of change of vibrational energy. The mass-production term is calculated with the 11-species chemistry model and rate coefficients from the Arrhenius law. The backward rate coefficients for the reverse reactions are calculated directly from equilibrium constants. `hy2Foam` employs Park's TTv model where the effective temperature for chemical reactions is assumed to be a geometric mean of trans-rotational and vibro-electronic temperatures. A more general expression is given by:

$$T_P = T_{tr}^{\alpha_P} \times T_{ve}^{1-\alpha_P} \tag{3.40}$$

where it was found that $\alpha_P = 0.7$ was found to be a better choice from the standpoint of accuracy [90].

The rate of change of vibrational energy term, $\dot{\omega}_v$, is composed of four additive terms representing the various energy transfer and coupling effects within the flow given by

$$\dot{\omega}_v = Q_{v-T} + Q_{v-v} + Q_{c-v} + Q_{e-v} \tag{3.33}$$

where Q_{v-T} represents the exchange of energy between the vibrational mode of molecules and the translational mode of the gas, Q_{v-v} accounts for the exchange of vibrational energy between two different molecular species, Q_{c-v} links chemical reactions (dissociation and recombination) to vibrational energy, and Q_{e-v} represents the energy exchange between free electrons and the vibrational mode of heavy molecules. Here Q_{v-T} is the most dominant term. The details of these relaxation models are summarised by Casseau [88].

3.2 Direct Simulation Monte Carlo Method

The direct simulation Monte Carlo (DSMC) method [14, 15] is a popular particle method for studying rarefied gas flow simulations. It is particularly effective for studying hypersonic flows involving chemical reactions, ionisation, and non-equilibrium phenomena at rarefied conditions. The four primary sub-parts of the DSMC algorithm involve particle movement, particle indexing, collision modelling, and flow field sampling. These procedures are carried out in each time step. In DSMC, each simulated particle represents a large number of actual gas molecules or atoms. The position of the simulated DSMC particle is randomly assigned in a cell. The velocity of the DSMC particle is sampled from the Maxwell-Boltzmann distribution at the initial cell temperature. Particle movement and collision modelling can be decoupled by selecting a time step shorter than the mean collision time. As in traditional Eulerian CFD algorithms, the computational domain is typically divided into a grid of cells. The cell size should be less than the local mean free path in the domain. Each cell handles particle collisions in the post-molecular movement module using probabilistic models.

This work utilises an in-house DSMC solver to investigate the hypersonic flow field around bluff bodies at rarefied ambient conditions. The in-house DSMC code is an MPI-parallelised 3D Fortran code that can simulate gas flows on both structured and unstructured meshes. The overall DSMC, as shown in Figure 3.3, algorithm for a single time step, can be summarised as:

- Update the position of all N particles in the system in the movement module using the following expression

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t = \mathbf{r}_i(t) + \Delta t\, \mathbf{r}_i \tag{3.34}$$

  The ray tracing algorithm is used to move particles in the in-house DSMC code, which is suitable for both structured and unstructured meshes.

- The particle distribution in each computational cell is updated to prepare for the subsequent collision routine.

- Compute the number of collisions to attempt in each computational cell. The no-time counter (NTC) model is used in the in-house code.

- Perform the collisions according to the binary collision model. The in-house code includes several models, such as the variable hard sphere (VHS), variable soft sphere (VSS), generalised hard sphere (GHS), generalised soft sphere (GSS), and ab initio models. The VHS model is used across all DSMC simulations conducted throughout this thesis. The VHS model is one of the most widely used models in DSMC owing to its ability to reproduce the experimentally observed temperature dependence of gas

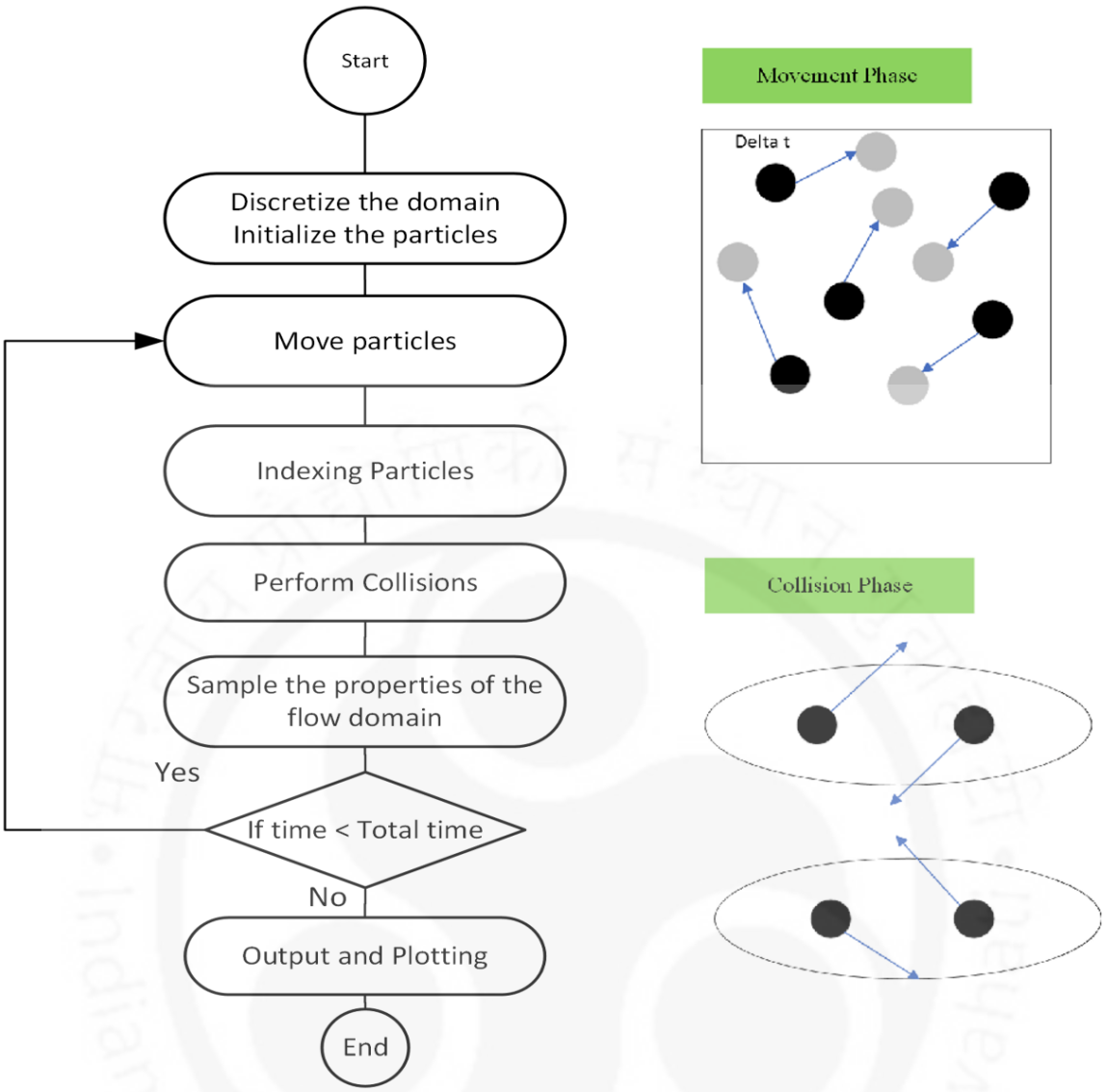

Figure 3.3: Flow chart of basic DSMC time integration scheme

transport properties while remaining computationally efficient.

- Sample the particle positions, velocities, and internal energies that are required to calculate the desired macroscopic values using the kinetic theory formulation.
- Go back to Step 1, where now $t \rightarrow t + \Delta t$ and repeat until the simulation end-time trend has been reached.

A contrasting feature of the DSMC algorithm from the deterministic simulation techniques, such as molecular dynamics, is its treatment of collisions. The overall objective of the collision model is to simulate a representative number of collisions between randomly selected pairs of molecules within each cell, with appropriate post-collision updates to the particles' velocities

and internal energies. The number of pairs (N_P) shortlisted for checking for collision is first estimated using the NTC method, given by:

$$N_P = \left(\frac{1}{2V_c}\right) F_N N(N-1)(\sigma_T C_r)_{\max} \Delta t \tag{3.35}$$

where N is the instantaneous number of particles in the cell, V_c is the cell volume, $(\sigma_T C_r)_{\max}$ is the maximum product of collision cross-section and relative speed of all possible particle pairs in the cell, Δt is the time step size, and F_N is the weight of the DSMC representative particle (1 DSMC particle = F_N real atoms/molecules/ions). A pair of particles (i, j) is chosen at random from all particles in the cell. Each collision pair (i, j) is then tested for collision using the acceptance-rejection method. The pair is believed to undergo a collision if

$$\frac{(\sigma_T c_r)_{ij}}{(\sigma_T c_r)_{\max}} > R_f \tag{3.36}$$

where σ_T is the total collision cross-section, c_r is the relative speed, and R_f is a random number uniformly chosen between 0 and 1. The DSMC model disregards the molecular shape and treats molecules as point particles of diameter d. The total collision cross-section is given by the VHS model, which follows an inverse-power-law relationship between viscosity

$$\sigma_T = \pi d_{ref}^2 \left(\frac{c_{r,ref}}{c_r}\right)^{2\nu} \tag{3.37}$$

$$c_{r,ref}^{2\nu} = \left(\frac{2k_b T_{ref}}{m_r}\right)^{\nu} \frac{1}{\Gamma(2-\nu)} \tag{3.38}$$

where *ref* subscript denotes reference values, k_b is the Boltzmann constant, ν is the VHS viscosity index. The d_{ref}, T_{ref}, and ν are parameters chosen to ensure correct viscosity-temperature relations are followed in the gas flows. Once a particle pair has been selected for collision, the velocities of both partners in the collision pair are changed according to the collision model, while their positions are not altered.

The VHS model alone is sufficient for modelling collisions in monoatomic gases. For molecular gas species, the redistribution of internal energies during a collision is also important. The Larsen-Borgnakke (LB) model is a widely used phenomenological model for handling energy exchange between translational and internal molecular modes during collisions of molecular gas particles. While elastic collisions update only the translational velocities, the LB model probabilistically redistributes energy among translational, rotational, and vibrational degrees of freedom to represent thermal relaxation. For each collision, internal energy relaxation occurs with a probability given by

$$P_m = \frac{1}{Z_m} \tag{3.39}$$

where Z_m denotes the relaxation number associated with mode m (rotational or vibrational). Instead of modelling energy redistribution for every collision, every Z_m^{th} collision is chosen for the LB model. When relaxation is selected, the post-collision internal energies are resampled from equilibrium distributions while conserving total energy. This stochastic energy redistribution enables the modelling of non-equilibrium effects.

Several phenomenological models have been reported for handling chemical and ionisation reactions relevant to hypersonic gas-flow simulation, especially in reentry flows. The total collision energy (TCE) model is one of the chemical reaction models in the DSMC algorithm, which employs equilibrium reaction rates using the Arrhenius law. The reaction probability P_{reac} in TCE model is estimated as:

$$P_{reac} = \frac{\sqrt{\pi}}{2} \frac{A T_{ref}^{b}}{\sigma_{ref} \langle c_r \rangle_{ref}} \frac{\Gamma(\bar{\zeta} + 2 - \omega)}{\Gamma(\bar{\zeta} + b + 3/2)} \left(\frac{E_{coll} - E_a}{k_B T_{ref}} \right)^{b-\omega+1/2} \left(\frac{E_{coll} - E_a}{E_{coll}} \right)^{\bar{\zeta}+1} \tag{3.40}$$

where A, b, E_a are the Arrhenius parameters from the rate equation $k_f(T) = AT^b \exp(-E_a/k_B T)$, E_{coll} is the total energy available in the collision ($E_c + E_{int}$), E_a is the activation energy ($P_{reac} = 0$ if $E_{coll} < E_a$) and $\bar{\zeta}$ is the average number of internal degrees of freedom of the colliding particles. In the present work, a 5-species, 11-reaction air chemistry model is used to generate flow-field data at higher altitudes.

The microscopic properties of the particles are sampled over a large number of DSMC time steps. Macroscopic properties such as density, pressure, and various temperatures for the degree of freedom are computed at the geometric centre of the cell using time averaging. For example, the number density in a computational cell is calculated as

$$n = \frac{F_N \bar{N}}{V_C} \tag{3.41}$$

where $\bar{N}$ is the time-averaged number of DSMC particles in the cell during the measurement interval. If the flow is steady, a simulation is allowed to reach a steady state. Then, properties are measured on a sufficiently large sample to reduce statistical error to an acceptable level. Other macroscopic properties and surface properties are calculated using kinetic theory formulations.

3.3 Boundary Condition-based Machine Learning Algorithm

The foundational theory and basic principles of the Boundary Condition-based Machine Learning (BCML) algorithm are introduced here, followed by testing the concept using

a standard incompressible benchmark: the lid-driven cavity problem. As noted in the previous chapter, several ML models have been designed to predict flow fields for specific flow problems and are often not extendable to a broader class of flows [17, 19, 66]. In contrast to such models, the proposed BCML algorithm is a physics-assisted, general-purpose ML model that can quickly predict flow fields for steady-state compressible gas flows around arbitrarily shaped reentry geometries and further assist CFD/DSMC codes in achieving faster convergence/better efficiency. It is known that the boundary conditions are central to solving partial differential equations and directly influence the solution. In the present work, the core idea of the BCML model is to train the neural network to capture the influence of various boundary conditions on an arbitrary point in the flow domain. The influence of various boundaries is modelled by discretising the planar angle into a number of discrete vectors. This concept is inspired by the discrete ordinate method (DOM), which is used to solve the radiative transfer equation (RTE), in which spatial and angular variables influence the evolution of the radiative intensity. Since radiative heat transfer is a three-dimensional process, the solid angle 4π is divided into a number of angular intervals, and a discrete set of direction vectors replaces the direction variable. In contrast, since the BCML model is first tested on two-dimensional flows, the planar angle 2π is divided into a set of discrete vectors. This leads to the concept of generators, which are lines originating from the point that, when extended, eventually end at one of the boundaries.

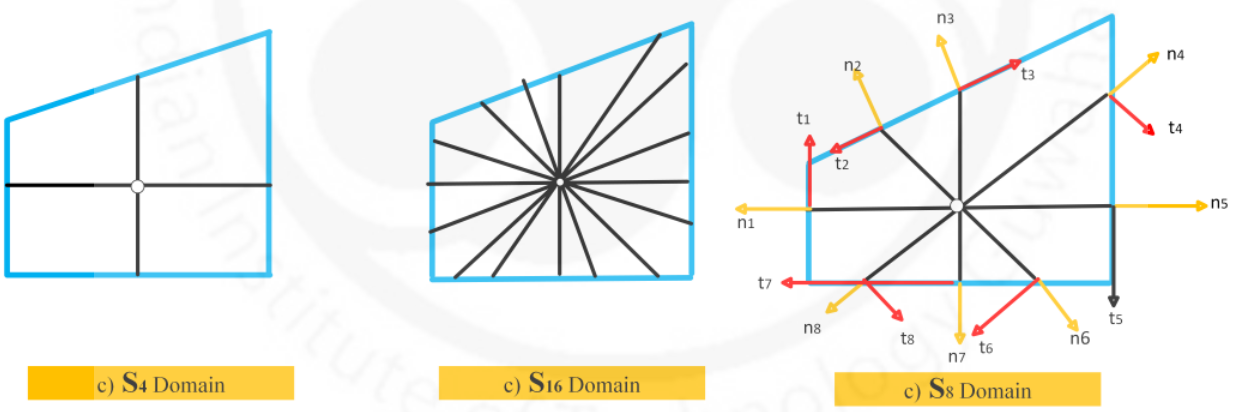

Figure 3.4: Schematic of the generators along with the boundary condition for a) 4-generator model (S_4), b) 16-generator model (S_{16}) and c) 8-generator model (S_8).

Figure 3.4 depicts the schematic of various generator models. Figure 3.4 a) shows a 4-generator or S_4 BCML model where four rays emanate from a point in a quadrilateral domain where the flow field variables are to be predicted. The four rays divide the 2π into four regions, each covering a $\pi/2$ angle. The relative lengths of the four generators from the point of origin to the final boundaries, along with the corresponding boundary conditions for pressure, temperature, and the two components of velocity (horizontal and normal components), form the input layer of the BCML neural network. As with the DOM

method, a finer discretisation of the planar angle leads to greater accuracy. Figure 3.4 b) and c) show an S_{16} and S_8 BCML model, which employs sixteen and eight generators, respectively. The effect of a single boundary on the point can now be handled by multiple generators based on its proximity. The length of a generator functions as the weight of the influence of the boundary on the point. Generally, the proximity of a boundary condition has a direct correlation with the influence it bears on the point, which means that the length of a generator has an inverse relation with the influence of the boundary condition. Hence, several distance functions, including exponential decay functions and reciprocal functions such as $1/\sqrt{s}$ and $1/s$, were initially investigated to model this behaviour. Subsequently, a systematic hyperparameter analysis was performed for each distance function, optimising the DNN architecture, batch size, and number of training epochs. The comparative results are summarised in Table 4.2. Based on the accuracy, convergence behaviour, and overall loss, the following exponential decaying function was found to provide the best overall performance and was therefore adopted in the present BCML framework:

$$f(x) = exp\Big(-2\sqrt{x}\Big) \tag{3.42}$$

where x is the length of a generator from the point to the end at the boundary. The proposed algorithm predicts the flow field at any point in the domain using the length function as the generator's weight and the corresponding boundary conditions.

The overall algorithm for finding the flow properties at any arbitrary point using the trained BCML in the domain can be summarised as follows:

1. Decide the number of generators N emanating from the arbitrary point in the domain according to the available BCML Model handling M boundary conditions.
2. For i^{th} generator, place the origin of the generator at the given point and assign the direction of the generator as $(i-1) \times 2\pi/N$. In the present work, the given point is the cell centre of an arbitrary cell of the structured or unstructured mesh.
3. Extend the i^{th} generator along the assigned angle till it reaches any boundary using a ray tracing algorithm. It is assumed that a particle resides at the cell centre and is made to move along the direction of the generator with unit velocity till it crosses all faces and reaches the boundary face.
4. Calculate the distance function from the length of the i^{th} generator and note the corresponding boundary condition.
5. Repeat the steps 2-4 for the remaining generators and get a total of $M \times N$ input values.

6. Train, optimise, and save a DNN model using $M \times N$ and P flow field properties as the number of nodes in the input and the output layers with an appropriate number of hidden layers and intermediate nodes.

7. Using the trained BCML model, estimate the flow field properties in all the cells of the mesh. This is followed by post-processing steps.

A DNN is employed in the BCML model because the relationship between the information from the boundary conditions and the corresponding flow field variables is highly nonlinear and involves a large volume of training data. DNNs are well-suited for learning such high-dimensional nonlinear mappings and therefore provide accurate predictions of the flow field. Generally, the higher the N, the number of generators, the better the resolution. However, it was found that the choice of $N = 16$ was optimum and increasing the number of generators had a small improvement in the accuracy, while the computational cost associated with the additional training data was linearly increasing. The choice of M, or the number of boundary conditions, is closely associated with the number of governing equations. In the present work, for each boundary condition, two numbers were considered to cover both the Dirichlet and Neumann-type boundary conditions. For a two-dimensional conservation law consisting of mass, momentum and energy equations, $M = 8$ and $P = 4$ for the four outputs (ρ, u, v, T) of the four governing equations. The pressure can be calculated using the closure equation.

The overall objective of the thesis is to report a new and novel ML model: the BCML model. To begin with, the feasibility of training a machine learning (ML) model using boundary conditions as input parameters for predicting flow fields is tested for incompressible viscous flows. The governing equations consist of the conservation of mass and the Navier-Stokes constitutive equations and are solved using the pressure-based solver. The present work uses CFD simulations for the lid-driven cavity using a commercially available FVM solver: ANSYS (Fluent). The cavity geometry consists of a square domain with unit width and height. A uniformly structured 129×129 nodal grid is employed in the current set of simulations. The total number of cells is 16641 with 16900 nodes. The fluid flow is considered to be steady and laminar. Dirichlet boundary conditions are applied for velocity at the top lid for a given Reynolds number. No-slip boundary conditions are considered for the remaining three walls. The output data from the CFD simulation are the u and v velocity components. The thermophysical properties, such as density and viscosity, for the hypothetical fluid used in the simulation are assumed to be unity. Only the horizontal velocity at the top wall is variable, taken as the numerical value of the Reynolds number for the case. The simulations are assumed to reach a steady state with a convergence criterion of 10^{-6}. CFD simulations are run over a wide range of Reynolds numbers from 100 to 1000, with an interval of 50. The data generated in the current CFD study are validated

against standard data available in the literature at Reynolds numbers 100, 400, and 1000. In addition, CFD simulations are performed at arbitrary Reynolds numbers (125, 372, 468, 923) to evaluate the validity of the new boundary-condition-based ML model. Hence, a total of nineteen simulations were run, and the flow field data for the velocity vector and pressure at all node points were saved for each simulation.

The 5-input data, derived from the nodal point coordinates and the Reynolds numbers of all CFD simulations, is used to train a deep neural network (DNN) for the ML model. The overall architecture of the neural network employed in the S_4 BCML model is depicted in Figure 3.6. The network consists of three main layers: the input layer, the hidden layer(s), and the output layer. There can be one or more hidden layers, depending on the problem and the need for accuracy. The architecture of the neural network is designed after performing hyperparameter analysis which includes optimizing the number of nodes in each layer, optimizing the batch size required for the training out of the total available sample data, and number of epochs, which is the number of iterations over the dataset to ensure no overfitting of the input data, the ML model has been properly trained based on the amount of CFD data available and by proper hyper parameter analysis. The testing data is set at 15-20% of the training data. The final optimised DNN architecture for the present study consists of 16 nodes in the first layer, followed by 2 hidden layers with 128 nodes each and 2 nodes in the output layer. The type of error calculated in the ML model study is mean square error. The activation function used in this ML algorithm is the Rectified Linear Unit (ReLU). Here, the input parameters for the neural network are the distances: S_1, S_2, S_3, S_4, and the Reynolds number, and the output parameters to be obtained are the u and v velocities.

For the BCML Model, the format of the velocity flow field data generated by CFD simulations is modified as follows. The flow field data, including the horizontal and vertical velocity components, are generated at each nodal point. A set of generators in N directions is initialised from each of these nodal points, and these generators are extended to finally intersect with one of the side walls, where the boundary conditions are applied. The lengths of the generators, as well as the perpendicular and normal components of velocity at the boundary, serve as the input layer for any generalised ML model. For an N-generator model, the total number of input parameters in the two-dimensional problem will be 3N (length of the generator and the two components of velocity at the boundary). However, in the current work, the number of generators is restricted to only four. The total number of nodes in the input layer should be $4 \times 3 = 12$. The classical lid-driven cavity problem is structured with only one non-homogeneous boundary condition at the top wall. The remaining seven values of the velocity boundaries are zero and can be removed from the set of inputs. Hence, the number of input parameters for the ML learning model reduces to just five (4 distances and 1 top wall horizontal velocity).

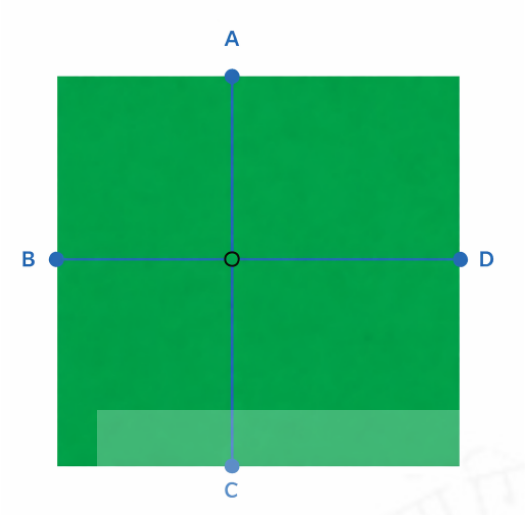

Figure 3.5: Schematic of the generators in the S_4 BCML model for the incompressible lid-driven cavity.

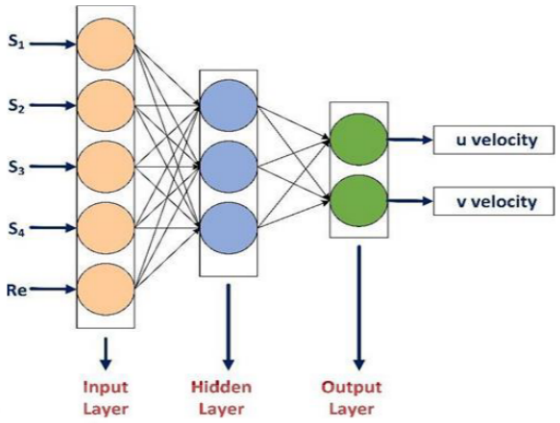

Figure 3.6: Deep neural network Architecture for the incompressible lid-driven cavity problem.

To summarise, for any arbitrary point within the cavity, say O, the distances from the four walls are input parameters. The fifth parameter is simply the simulation's Reynolds number. A schematic of the S_4-BCML data point for the lid-driven cavity is shown in Figure 3.5, where the lengths $S_1 = OA$, $S_2 = OD$, $S_3 = OC$, $S_4 = OB$ serve as four of the five inputs for the S_4 BCML model training. The BCML algorithm is used to calculate the flow field velocities for the entire domain, and the centerline data is then extracted from the flow field data. Alternatively, points on the centerline can be fed as an input to the model to directly get the u and v velocities. The generated flow field is first validated along the vertical and horizontal centerlines of the cavity against the standard literature data from Ghia et al. [97]. It is important to note that an ML model will be accurate only if the underlying data is validated and verified. Validation centerline plots of the horizontal component of velocity along the y-direction (x = 0.5) and the vertical component of velocity along the x-direction (y = 0.5) at $Re = 400$ are shown in Figure 3.7. It can be observed that the u and v plots are in excellent agreement with those reported by Ghia et al. [97]. The assessment of the new S_4 BCML model has been conducted for the following two cases. Firstly, Case 1: flow field prediction at Reynolds number 400, and secondly, Case 2: at Reynolds number 125. It should be noted that the Case 1 flow field is a part of the training data. In contrast, the flow-field data for Case 2 ($Re = 125$) are not included in the training data. In other words, the ML model has no prior information about the flow field data for $Re = 125$. The absolute value of mean square error for the flow field prediction from the S_4 BCML model is 10.3653, and the overall accuracy is 96% for Case 1. In comparison, the mean square loss is 21.7291, and the overall accuracy is 95% for Case 2. Although accuracy decreased on the untrained data, the results were encouraging. Figure 3.7 also shows the line plots of the horizontal and vertical components of the velocity obtained from the BCML model.

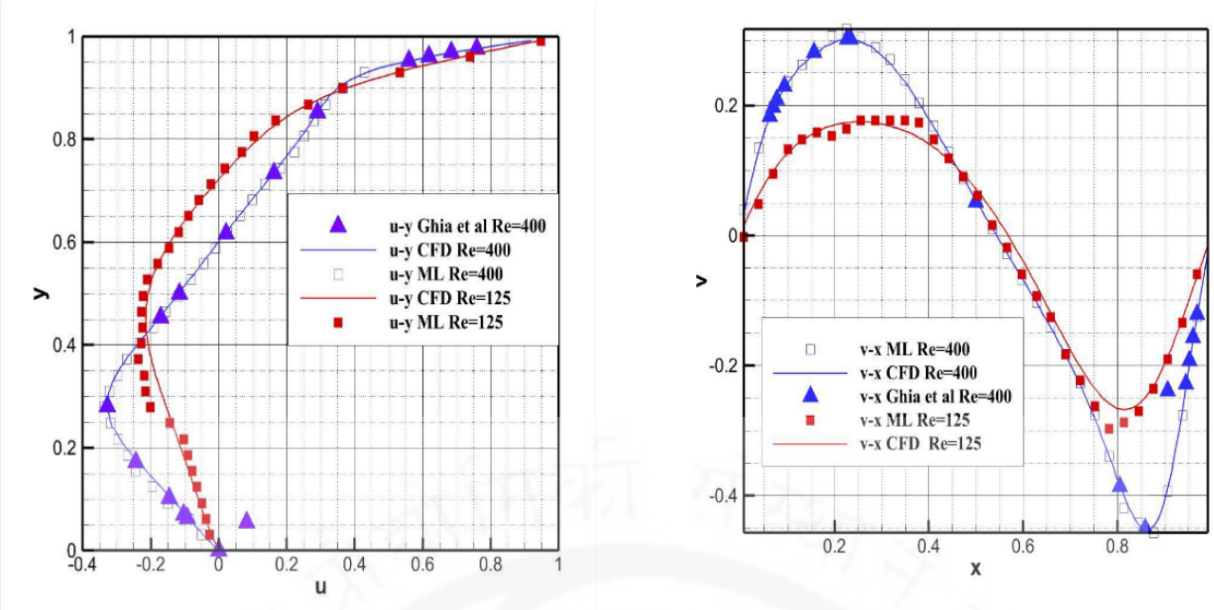

Figure 3.7: Comparison of line plots of u (left) and v-velocity (right) from CFD simulations and S_4 BCML model along the vertical and horizontal centreline of the cavity for various Reynolds numbers.

As shown in Figure 3.7, the ML model results agreed well with the CFD results for Case 1. Similarly, in Case 2, the ML model results show decent agreement, except in regions around x=0.35 and x=0.8, where there is minor deviation, as apparent from the figure. This deviation can be reduced by increasing the mesh resolution. This results in a larger training dataset, improving the model's accuracy, particularly near the walls.

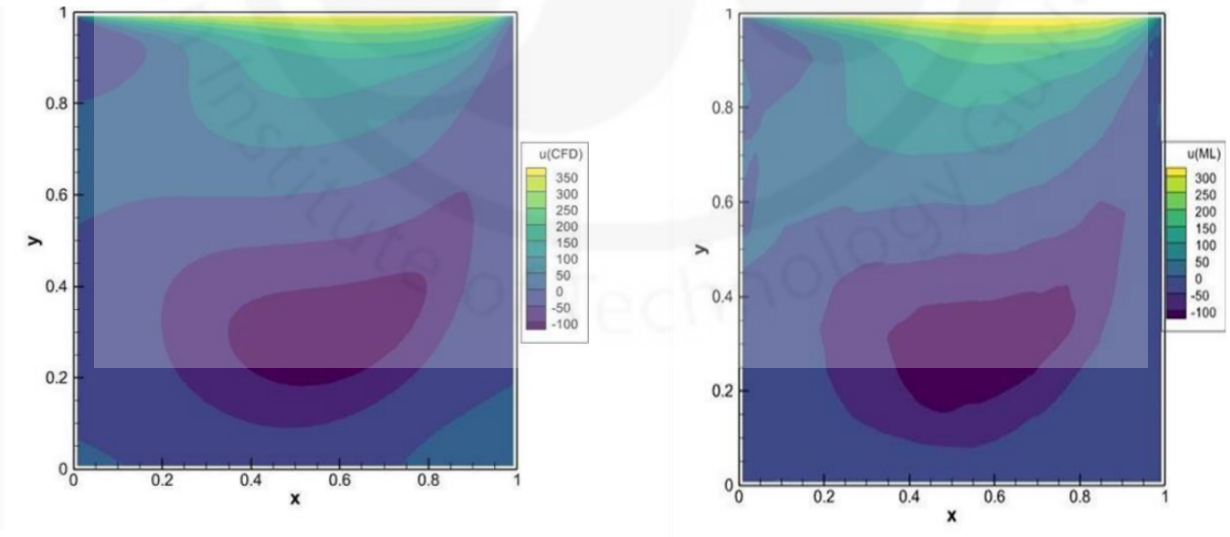

Figure 3.8: Contour plots of u-velocity obtained from CFD (left) and S_4 BCML model (right) for incompressible lid-driven cavity problem at $Re = 400$

Figure 3.8 shows the contour plots of the horizontal component of velocity (u) in the cavity generated using CFD simulation and those estimated using the S_4 BCML model,

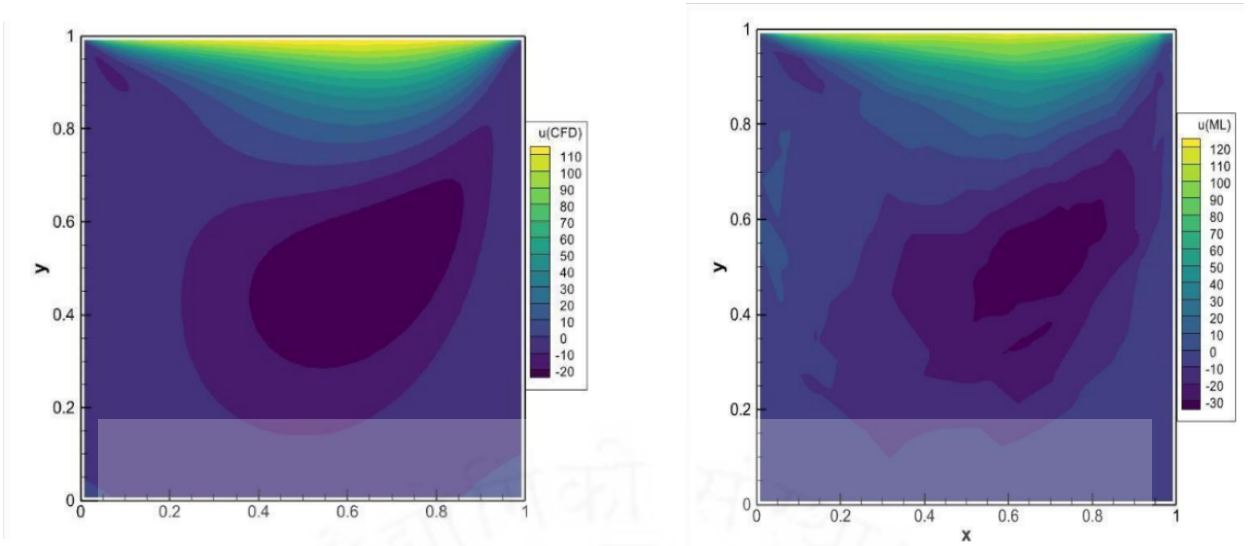

Figure 3.9: Contour plots of u-velocity obtained from CFD (left) and S_4 BCML model (right) for incompressible lid-driven cavity problem at $Re = 125$

respectively. Similarly, Figure 3.9 shows the contour plots of the horizontal component of velocity (u) taken in absolute SI units(without non-dimensional zing), which have been generated from CFD simulation and the data predicted by the S_4 BCML model at Re=125. The main advantage of the new BCML model is the significant reduction in computational resources required to obtain the flow field. The average computational time for a single CFD simulation at a given Reynolds number is about 38 minutes. In comparison, the flow field calculation using the new BCML model takes only 2.12 seconds on a 129×129 mesh.

This study was primarily undertaken to increase confidence in the approach and to assess its applicability toward achieving geometry-independent flow-field prediction. As the results were encouraging, a more difficult and practical application of reentry flows was chosen and became the central application of the current thesis. The BCML models have been reported for laminar, supersonic, and hypersonic flows under frozen and reacting conditions, and for two-dimensional external flows over arbitrarily shaped bluff bodies or reentry bodies at lower altitudes, using a continuum solver. In addition, BCML models for higher altitudes, based on data from the DSMC solvers, are reported and cover the important reentry corridor. The development of these models is described in the following chapters.

3.4 BCML vs PINN vs Image-based ML Algorithms

The proposed BCML framework is more appropriately classified as a *physics-assisted* or *physics-augmented machine learning (PANN) model*, which is fundamentally different from a physics-informed neural network (PINN). It must be noted that in the present work, neither the governing equations are directly solved by the ML model, nor are the loss functions

Table 3.1: Comparison of ML frameworks for flow prediction

Aspect	**BCML**	**PINNs**	**Image-Based ML**
Inputs	Boundary conditions only; no explicit geometry or mesh information	Spatial coordinates with boundary conditions (and time, if applicable)	Geometry encoded as images
Physics treatment	Implicitly from training data; no PDEs	Governing equations enforced via PDE residuals	Learning from data unless additional constraints are imposed
Generality	Geometry-agnostic within trained boundary-condition space	Geometry and problem-specific	Strongly geometry-dependent
Hypersonic effects	Captured indirectly through data (present work)	Difficult to handle strong shocks and finite-rate chemistry	Captured indirectly through data; physical consistency not guaranteed
Primary utility	Approximate flow field prediction, assisting CFD/DSMC (initialisation, meshing)	Potential replacement of conventional PDE solvers	Rapid flow field prediction within fixed design spaces

included during training. Instead, the physics is introduced through three components: (i) generator-based encoding of the boundary conditions, (ii) physically meaningful input features based on the distance function and boundary-condition information, and (iii) physics-aware post-processing together with a short CFD relaxation step to improve physical consistency of the predicted solution.

This distinguishes the proposed BCML framework from PINNs, where the partial differential equations, boundary conditions, and conservation laws are enforced during the optimisation through loss terms. Although PINNs have demonstrated promising results for several canonical flow problems, their application to practical hypersonic flows involving strong shock discontinuities, thermochemical non-equilibrium, and multiscale rarefied effects remains computationally challenging. Similarly, the idea of the BCML differs from the operator-learning methods, such as DeepONet and Fourier Neural Operators (FNOs), which learn mappings between input and output function spaces (operators). In contrast, BCML directly learns the nonlinear relationship between the values of the boundary condition and the flow-field variables using supervised learning on NSF or the DSMC-generated datasets. A comparison between BCML, PINNs, and image-based learning approaches is presented in Table 3.1.

Accordingly, the proposed BCML framework should be regarded as a physics-assisted surrogate model, where physical knowledge improves the representation of the learning problem without explicitly constraining the neural network through the governing equations

during training.

Chapter 4

Low-Altitude NSF-Based BCML Model

The previous chapter introduced the fundamental concept of the Boundary Condition-Based Machine Learning (BCML) framework and demonstrated its feasibility through the classical lid-driven cavity problem. Building upon the experience gained from this proof-of-concept study, the present chapter extends the BCML methodology to a practical aerospace application involving low-altitude hypersonic re-entry flows. In the continuum flow regime (40-60 km), where the Navier-Stokes-Fourier (NSF) equations remain valid, a BCML model is developed and trained using flow-field data generated from high-fidelity CFD simulations. The capability of the proposed framework is subsequently assessed for unseen flow conditions and geometries, thereby establishing the potential applicability of BCML to practical hypersonic flow prediction problems.

4.1 BCML Framework for Frozen Flows

4.1.1 Algorithm

The objective is to develop a BCML model that can predict the flow field across a broader range of flow conditions and for arbitrary geometries not included in the training data. The first BCML model was developed for laminar, supersonic flows under frozen conditions, and for two-dimensional external flows over an arbitrarily shaped bluff body or a re-entry body at lower altitudes where the continuum assumption is valid, trained on data generated using an NSF-based solver.

The BCML model predicts the flow field at any point in the domain using the generator length as a weight, and the corresponding boundary conditions for pressure, temperature, and the two velocity components (tangential and normal) form the neural network's input layer. For the low-altitude NSF-based BCML model, only Dirichlet and Neumann boundary types are used, which are sufficient to represent freestream, far-field, and constant-temperature

wall conditions in 2D supersonic re-entry flows. For each generator, eight boundary inputs are provided: pressure, temperature, and the two velocity components for both Dirichlet and Neumann types. For constant-temperature walls and freestream boundaries, the Neumann terms are set to zero, and the Dirichlet values are normalised. For far-field and zero-gradient boundaries, the Dirichlet terms are set to zero and the Neumann terms to unity. The final pressure, temperature, and velocity at a point are then obtained from the output layer of the BCML network and compared with those obtained from the in-house FVM-based NSF solver.

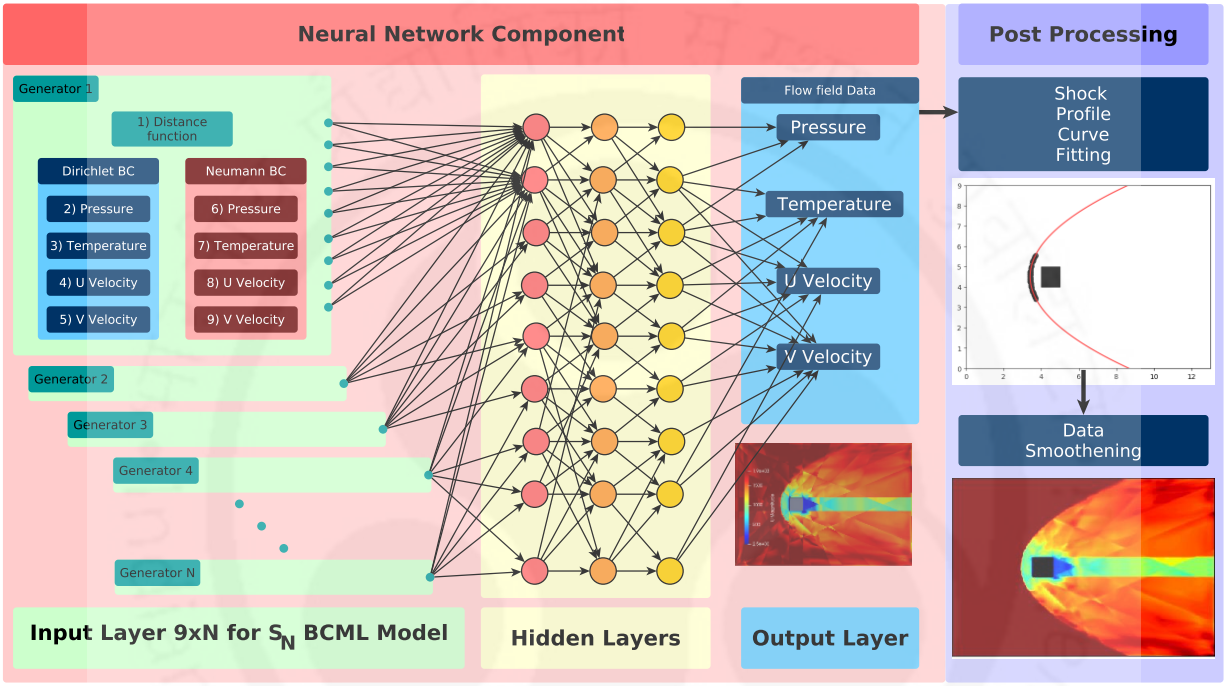

Figure 4.1: Flowchart of the Boundary Condition-based Machine Learning model for low altitude.

Figure 4.1 outlines the BCML workflow, which comprises a deep neural network (DNN) and subsequent post-processing modules. The DNN input layer contains the length function and eight boundary-condition inputs for each generator, giving an $(1+4+4)\times N = 9N$-node input layer for an S_N BCML model, followed by several hidden layers and a four-node output layer. The model can predict flow quantities at arbitrary points or at cell centres of a structured or unstructured grid; the latter is more practical since the training data come from a cell-centred FVM solver.

Although the BCML-predicted flow field closely matched the FVM results, minor unphysical gradations appeared when fewer than 32 generators were used, and increasing the angular resolution greatly enlarged the required training data. A compromise using sixteen generators with dedicated post-processing was found to be effective. In addition to point-wise flow prediction using the neural network, the BCML framework also incorporates reconstruction

and post-processing stages required for obtaining physically interpretable flow quantities. These stages include contour-field reconstruction, stagnation-line extraction, shock stand-off distance estimation, and evaluation of derived thermophysical properties for subsequent CFD/DSMC-assisted applications. The BCML model accurately captured the bow shock, which was further refined by identifying high-gradient points and fitting a polynomial shock profile, yielding excellent agreement with the FVM solution. Since the pre-shock region must match freestream conditions in supersonic flow, the upstream BCML predictions were corrected accordingly to maintain consistency with the Rankine–Hugoniot conditions. Finally, the BCML flow field was used as initial conditions for a short FVM relaxation step (∼100 iterations), which smoothened the downstream region at negligible computational cost compared to a full CFD run. Together, these steps ensure that the BCML-generated solutions are physically consistent, smooth, and compliant with conservation laws.

4.1.2 Data Generation and Training ML Model

A diverse and extensive database is the most crucial aspect for training the DNN component of the BCML algorithm.The BCML model is implemented in Python using the TensorFlow deep learning framework with the Keras high-level API. The in-house FVM code is used to simulate supersonic flow in a discretised domain containing a unit-sized bluff body, as shown in Figure 4.2. The idea is to generate a database for only two types of bluff bodies: a square body of unit dimension and a circular body of unit diameter. Figure 4.2 shows a domain of 13 m×9 m that contains a square bluff body-centred at (4.5 m, 4.5 m). The supersonic flow enters from the left boundary, and the remaining domain boundaries are Neumann-type. Instead of changing the direction of the flow at the inlet, the bluff body is rotated about its centre to include the effect of the angle of attack. The surface of the bluff body is assumed to be at a constant temperature. Further, for simplicity, it is assumed that the flow does not slip at the bluff body surface.

Firstly, the BCML model for low altitudes (40-60 km) at frozen conditions was developed and used to predict the flow field for two-dimensional, supersonic external flow over a re-entry vehicle. The training data cover a range of Mach numbers ($Ma = 2$–8) and altitudes (40–60 km). Since this altitude range lies well within the continuum regime, the Navier–Stokes (NS) equations were solved using the finite volume method (FVM) to generate the training dataset. To focus on testing the practicality and feasibility of the BCML concept, chemical reactions were neglected, and the single-temperature NSF formulation was assumed sufficient. The viscous stress tensor and heat flux vector in viscous flux vectors are handled using first-order approximations with the Stokes assumption. Inviscid terms are handled by employing the HLLC approximate Riemann solver, and the viscous terms are handled using the central difference scheme. Additionally, Sutherland's law for viscosity, along with Eucken's relation

for thermal conductivity using air as the gas medium, is employed in the FVM simulations.

The supersonic inlet boundary condition was varied across Mach numbers ($Ma = [2, 3, 4, 5, 6, 7, 8]$) at each altitude. Re-entry vehicles typically travel at an angle of attack (AoA), which was incorporated by rotating the square-shaped bluff body about its centre ($AoA = [0°, 15°, 30°, 45°]$). The wall temperature was also varied ($T_{\text{wall}} = [300\ \text{K}, 650\ \text{K}, 1000\ \text{K}]$) to account for thermal effects on the flow physics. In total, 5 altitudes × 7 Mach numbers × 4 AoA × 3 wall temperatures resulted in 420 simulations for the square bluff body, which were used to construct the database for the DNN component of the BCML model. The corresponding ambient conditions were tabulated in Table 4.1.

Table 4.1: Ambient conditions employed in the low-altitude NSF simulations

Altitude (km)	Density (kg/m^3)	Pressure (Pa)	Temperature (K)
40	3.860×10^{-3}	270.704	245.127
45	1.900×10^{-3}	138.103	256.500
50	9.960×10^{-4}	76.104	256.460
55	5.143×10^{-4}	36.558	248.400
60	2.700×10^{-4}	18.385	237.700

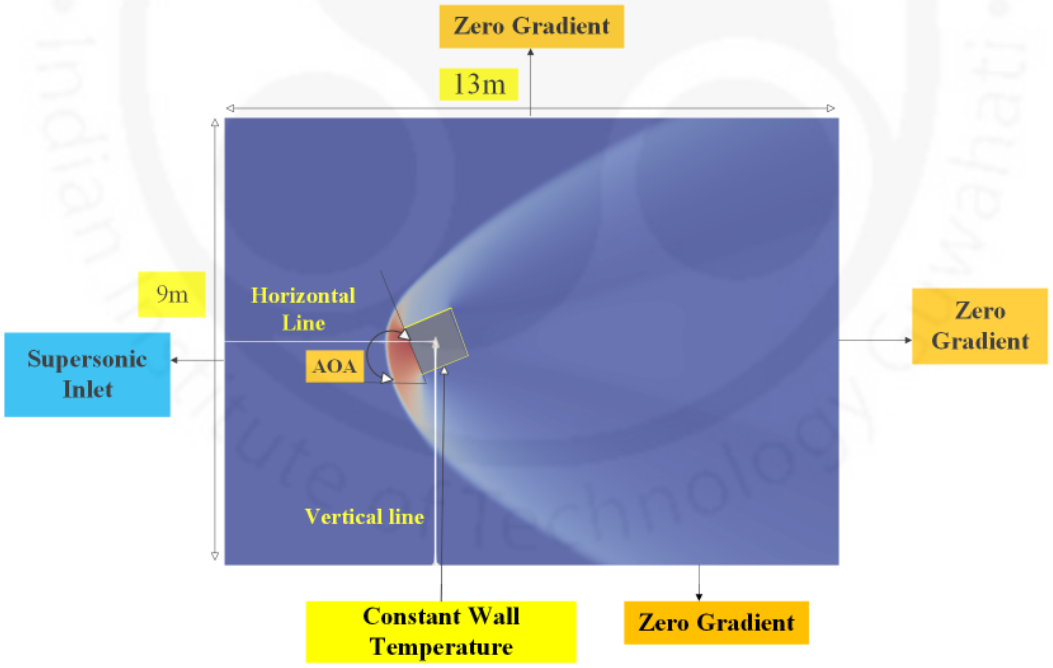

Figure 4.2: Schematic of computational domain of external flow around a square bluff body and boundary conditions.

A preliminary DNN model was trained using this database, and it was observed that the model could not handle the curvature of re-entry vehicles, evident from a large deviation between the BCML flow fields and those calculated by the FVM solver. To this end, supersonic flow past a unit-diameter circle was added to the database. The boundary

conditions (pressure, temperature, and velocity at the supersonic inlet, and wall temperature on the bluff body surface) used in such simulations were randomly sampled between the corresponding extrema. A total of 80 simulations were performed for the circular cylinder to generate a sufficiently diverse database covering the freestream conditions corresponding to the altitude range of 40–60 km. These simulations were sampled over different combinations of ambient conditions and wall boundary conditions to ensure adequate coverage of the training space while maintaining a computationally feasible database for the preliminary BCML model. The resulting database was found to be sufficient for achieving stable training and satisfactory predictive performance for simple bluff-body geometries. A total of 500 Navier–Stokes–based CFD simulations were performed to develop the low-altitude BCML model using a Fortran-based parallelised solver. All simulations were executed on 48 processors utilising the high-performance supercomputing facilities available at the Indian Institute of Technology Guwahati.

Designing a DNN model requires optimising the number of hidden layers and the number of neurons per layer, and choosing an appropriate non-linear activation function (σ_l). The current work employs the rectified linear unit function [$ReLU$ (rectified linear unit) = $max(0, x)$] as the activation function. It is known to train the network much faster, especially for regression problems. The weights and biases of the DNN are calculated using error back-propagation algorithms. A stochastic gradient descent optimiser is used to update the weights and biases. The optimisation is performed to minimise the mean square error (MSE). For the S_{16} low altitude BCML model employing $exp(-2\sqrt{s})$ distance function, the optimised DNN model consists of one hidden layer with 16 nodes, followed by three hidden layers with 256 nodes in each hidden layer and finally, the output layer with four nodes.

Table 4.2: Optimised statistics after hyper-parameter analysis for various length functions for an S_{16} BCML model

Function	BCML model	Epochs	Batch size	Accuracy (%)	Loss (MSE)
e^{-s}	$144, 16, 256, 256, 4$	500	512	96.682	0.568
e^{-2s}	$144, 16, 256, 256, 256, 4$	200	512	96.746	0.627
$e^{-0.5s^2}$	$144, 16, 256, 256, 256, 4$	500	512	96.185	0.709
e^{-s^2}	$144, 16, 256, 256, 4$	500	512	96.746	0.627
e^{-2s^2}	$144, 16, 256, 256, 256, 4$	150	512	95.265	0.741
$e^{-0.5\sqrt{s}}$	$144, 16, 256, 256, 256, 4$	500	512	95.755	0.598
$e^{-2\sqrt{s}}$	$144, 16, 256, 256, 256, 4$	500	512	96.899	0.271
$(1/\sqrt{s})$	$144, 16, 256, 256, 256, 4$	200	512	96.332	0.519
$(1/s)$	$144, 16, 256, 256, 256, 4$	200	512	95.380	0.230
$(1/s^2)$	$144, 16, 256, 256, 256, 4$	500	512	96.130	0.488

Generators are drawn from the cell centres of each cell in the domain. To find the length of the generator from the point of origin to a boundary, the ray-tracing algorithm was used to trace the motion of a fictitious particle with the cell centre as its starting point, with

velocity components at various angles, depending on the chosen number of generators. In the case of the S_{16} model, 16 points originating from the centre of a cell, with each assigned horizontal and vertical component of velocity with the unit speed at incremental angles of $2\pi/16$ starting from zero around the 2π planar angle. The generality of such a ray-tracing algorithm can easily be extended to handle simulations with unstructured meshes. The length function for all generators, along with the appropriate boundary conditions at the end of the generators, forms a single data point at the inlet layer of the DNN component of the BCML module. The normalised pressure, temperature, and velocity components from the converged FVM simulations form the data point at the final layer of the DNN. The domain for the square-shaped bluff body is discretised using a structured mesh, and each simulation is performed on 104400 cells. A single simulation will yield 104400 training data points. Therefore, the total number of data samples generated is reportedly 0.5 billion, which is a considerably large database. In the present work, an S_{16} BCML model with 144 nodes in the input layer has been developed. This necessitates training the DNN on a large dataset, as the 144-dimensional phase space must be covered for accurate prediction. In fact, choosing the combinations of ambient conditions, inlet speed, angle of attack, and wall temperatures considerably reduces the span of this high-dimensional phase space.

It is known that the output of a j^{th} neuron in the l^{th} layer is given by

$$a_j^l = \sigma_l \left(\sum_{k=1}^{N^{(l-1)}} w_{jk}^l a_k^{l-1} + b_j^l \right) \tag{4.1}$$

where σ_l denotes the nonlinear activation function, w_{jk} denotes the weight connecting k^{th} neuron of the previous layer to the j^{th} neuron in the current layer, b_j^l is the bias of the neuron. In the case of the first hidden layer, a_k^{l-1} is the input provided to the neural network. In this way, obtaining the output of each neuron from the given input is called a feed-forward operation. Designing a DNN model requires optimising the number of hidden layers, the number of neurons in each layer, and the non-linear activation function (σ_l). In the current work, the rectified linear unit function [$ReLU = max(0, x)$] is chosen as the activation function. It is known to train the network much faster, especially for regression problems [98]. The weights and biases of the DNN are calculated using error back-propagation algorithms [99, 100]. A stochastic gradient descent optimiser is used to update the weights and biases. The optimisation is performed to minimise the mean square error (MSE).

The Mean Squared Error (MSE) measures the average of the squared differences between the predicted and reference values and is given by

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 , \tag{4.2}$$

where N is the total number of samples, y_i is the reference (target) value, and $\hat{y}_i$ is the corresponding value predicted by the neural network. Owing to the squaring operation, MSE assigns a larger penalty to larger prediction errors.

The Mean Absolute Error (MAE) represents the average absolute difference between the predicted and reference values and is expressed as

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|\,. \tag{4.3}$$

Unlike MSE, the MAE provides a direct measure of the average prediction error and is less sensitive to outliers. Lower values of both MSE and MAE indicate better predictive performance and convergence of the trained neural network.

The DNN component's design was heuristically modified to obtain the optimised model. In addition to the choice of the number of hidden layers and the number of neurons per layer, selecting an appropriate length function to model the effect of each generator's boundary condition on a point is critical. It is to be noted that the length of the generator has an inverse relation with the weight a boundary condition has on a point in the domain. Additionally, if the boundary condition is far away, the weight should be negligible. This ensures that the model predicts flow features that are independent of the domain size. This effect is handled by testing a wide range of exponentially decaying and inverse functions as listed in Table 4.2. The second column indicates the number of neurons in each layer, from the input to the output layer, including intermediate hidden layers. The final DNN model design, along with the optimised number of epochs and batch sizes for each length function, is also tabulated. Finally, it was concluded that $exp(-2\sqrt{x})$ was the most accurate model, as evidenced by the loss and accuracy results in Table 4.2. With the S_{16} BCML model with $exp(-2\sqrt{s})$, the optimised DNN model consists of input layer with 144 nodes, followed by three hidden layers with 256 nodes in each hidden layer, and finally, the output layer with four nodes. The optimised number of Epochs and batch size are 500 and 512, respectively, whereas the MSE loss and accuracy are 0.27134 and 96.8994%, respectively.

4.1.3 Flow Field Prediction for Simple Geometries

The first test case considers a Mach 6 flow at ambient conditions, corresponding to an altitude of 57.5 km. The bluff body shape chosen for the first test was a simple circular cylinder geometry at a constant surface temperature of 750 K. Even though the shape of the bluff body is simple, it should be noted that the flow field data estimated using the FVM solver at the given inlet and wall boundary conditions are not included in the training data for the S_{16} BCML model. Figure 4.3 compares flow field contour plots of pressure, temperature, and the two components of velocity generated from the BCML model and those obtained from the FVM solver. It can be observed that, qualitatively, the wave features observed in

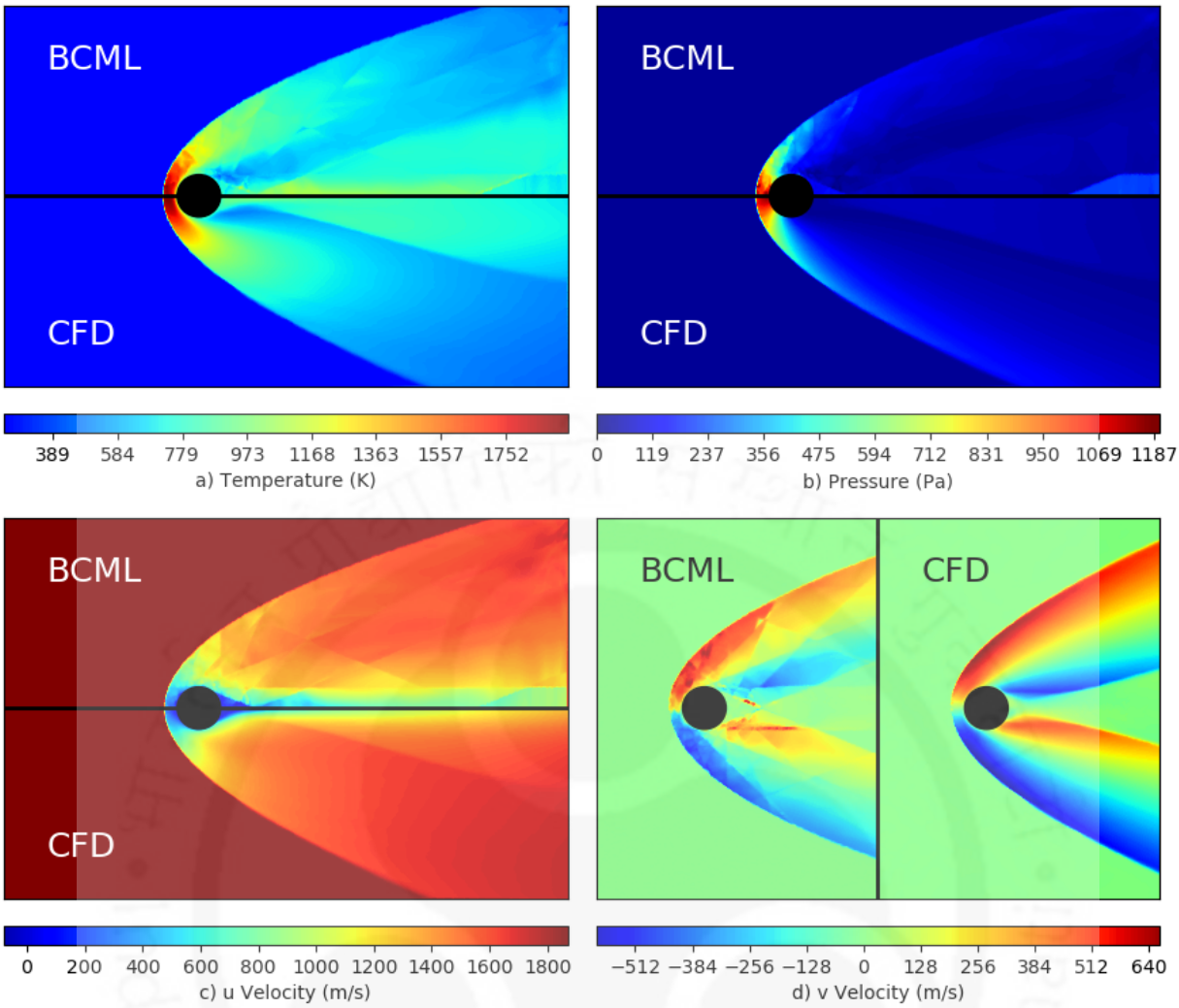

Figure 4.3: Comparison of flow field contour plots (a) translational temperature, b) pressure, c) u-velocity, and d) v-velocity predicted by the S_{16} BCML model and obtained from the FVM solver for Mach 6 flow over a circular cylinder at 57.5 km with $T_{wall} = 750$ K

the flow field generated from the BCML model are similar to those observed in the contour plots generated by the FVM solver.

The BCML model accurately captures the shock front, particularly in regions near the bluff body. However, it must be pointed out that the shock front obtained from the BCML away from the bluff body and near the far-field boundaries loses its profile and is less accurate than that predicted by the FVM solver. Additionally, the accuracy of the shock profile is better for temperature and U-velocity than for pressure and V-velocity, as shown in Figure 4.3. The contour plots of the horizontal and vertical components of the velocity also show that the flow field reconstruction by the BCML model in the recirculation zone in the wake region of the bluff body matches well with those predicted by the FVM solver. Furthermore, the BCML model accurately captures features such as the viscous boundary layer, Prandtl-Meyer expansion, and recompression shocks. However, the BCML model does not accurately capture the viscous core and free shear layer.

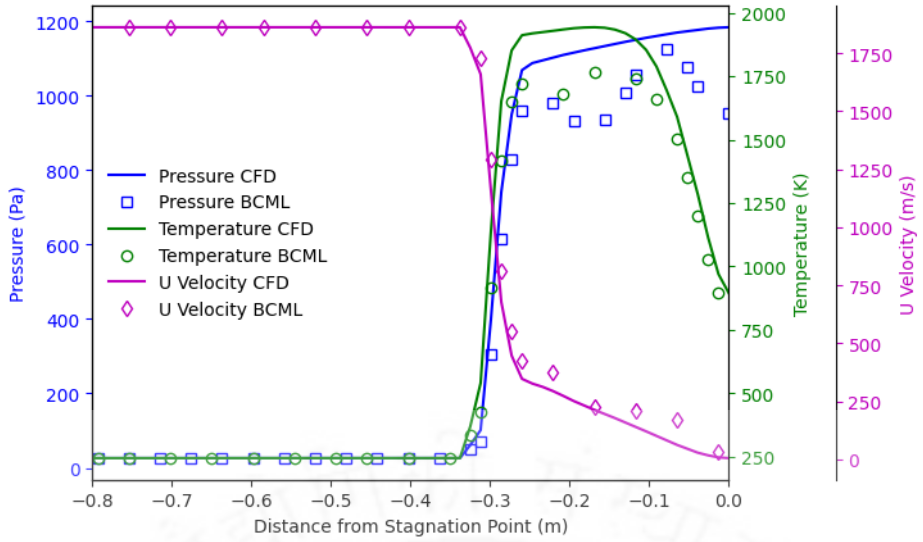

Figure 4.4: Comparison of S_{16} BCML model and FVM solver predictions for flow field properties (temperature, pressure, and u-velocity) along the stagnation line for Mach 6 flow over a circular cylinder at 57.5 km.

Generally, capturing the shock front is crucial for studying supersonic external flows. The shock stand-off distance from the BCML model was 0.354 m from the stagnation point. In comparison, the shock stand-off distance obtained from the FVM simulation was 0.351 m. The comparison of the shock front profile and the shock strength obtained from the two models is shown in the centreline plot in Figure 4.4. The horizontal axis marks the distance along the stagnation line, with zero representing the stagnation point on the bluff body. Again, the shock stand-off distances predicted from the BCML model match those obtained from the FVM simulation. The peak temperature in the post-shock region predicted by the BCML model is 1942.5 K, which is about 8.7% higher than that obtained from the FVM simulation (1786 K). In contrast to the temperature plots, the peak pressure in the post-shock region predicted by the BCML model is about 1184.22 Pa, which is about 5.25% higher than that obtained from the FVM simulation (1125.12 Pa).

Table 4.3: Initial and boundary conditions for validation cases along with L_2 norm errors.

Geometry	Pressure (Pa)	Temp (K)	AOA	Inlet Velocity (m/s)	T_{wall} (K)	P_{error}	T_{error}
Circle	26.157	245.12	0°	1862.502	750	5.737%	1.288%
Square	50.344	252.8	22°	1278.419	550	3.387%	1.001%
Fire-II	28.138	254.00	0°	1597.319	1000	7.697%	1.917%

The objective is to estimate the flow field quickly using the new BCML model. The computational time required for the MPI-parallelised FVM Fortran code to obtain converged flow-field results is around 6 CPU hours on an Intel Xeon workstation, using 24 threads, after a total of 34740 iterations. In contrast, the Python-based BCML code predicts the flow field results with reasonable accuracy in less than 5 s. This results in a practical speed-up of

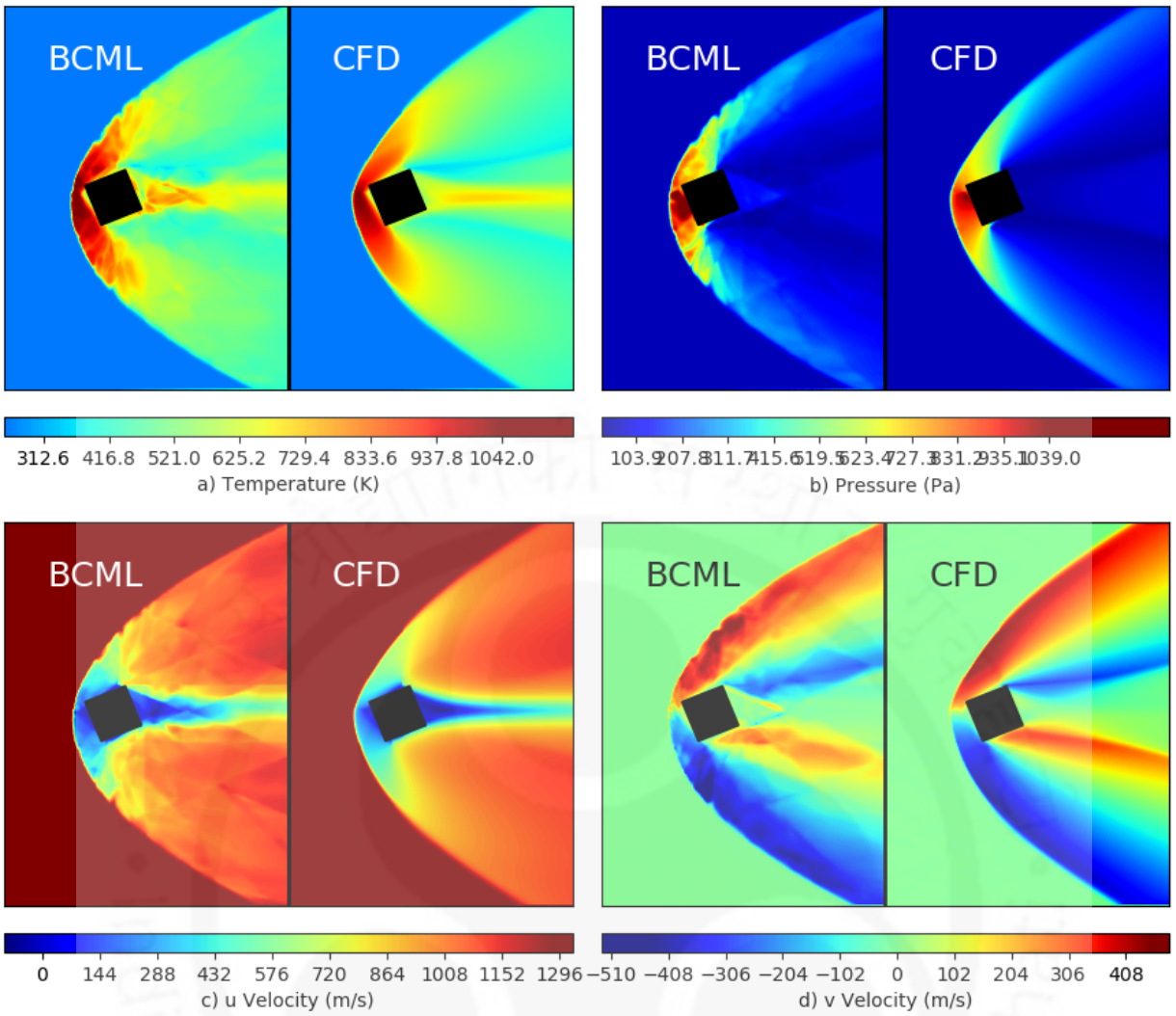

Figure 4.5: Comparison of flow field contour plots (a) translational temperature, b) pressure, c) u-velocity, and d) v-velocity predicted by the S_{16} BCML model and obtained from the FVM solver for Mach 6 flow over a square cylinder at 52.6 km with $T_{wall} = 550$ K

more than 4000 for the BCML model over the traditional FVM solver.

The second test case considers Mach 4 airflow over a square cylinder at a 22° angle of attack, with ambient conditions corresponding to an altitude of 52.6 km. The wall temperature on the square bluff body was assumed to be 550 K. Again, the inlet and wall conditions used for the test case were not used to train the BCML model. Figure 4.6 shows the centreline plots of various properties in front of the square cylinder. The corresponding contour plot has also been depicted in Figure 4.5. Again, it is clear that the shock stand-off distance predicted by the BCML model closely agrees with that estimated by the FVM solver. Additionally, the peak temperature indicated by the BCML model is within 5% accuracy compared to the peak temperature obtained from the FVM solver, which is even better than what was obtained in the first case.

Figure 4.7 shows the contour plot of the relative error for temperature between the BCML

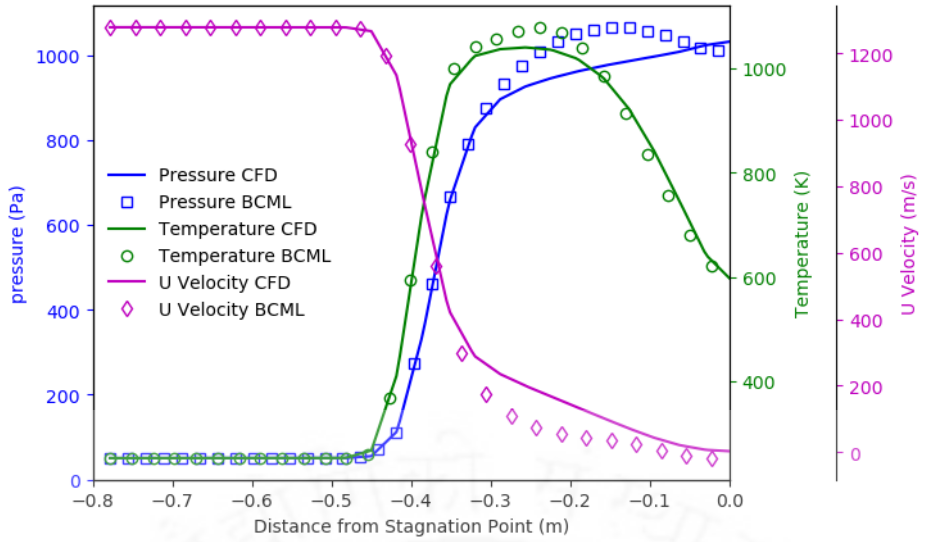

Figure 4.6: Comparison of S_{16} BCML model and FVM solver predictions for flow field properties (temperature, pressure, and u-velocity) along the stagnation line for Mach 4 flow over a square cylinder inclined at 22° at 52.6 km.

model and the FVM solver. The contour plots of the error indicate that the BCML model predicts the temperature flow field quite accurately. The error in the flow field predicted by the BCML model is higher at a few discrete points at the edge of the bow shock and in the wake region far downstream from the bluff body, and can be ignored for analysis. The L_2 norm for the entire domain was found to be less than 1% for temperature and less than 4% for pressure. The error norms for all the test and validation cases are tabulated in Table 4.3.

4.1.4 Flow Field Prediction for re-entry Vehicles

The aim is to develop a machine learning-based model to predict the flow field around an arbitrarily shaped bluff body that is not included in the training data. The previous section assesses the validity of the BCML model for gas flows around bluff bodies included in the training data under various flow conditions. The next step is to analyse the BCML model's performance in predicting the flow field around an arbitrarily shaped bluff body under arbitrary flow conditions. The next validation case considers a Mach 5 flow around a FIRE-II (Flight Investigation of re-entry Environment-II) re-entry vehicle at an altitude of 59.26 km [101]. The surface temperature of the FIRE-II re-entry vehicle is assumed to be 1000 K. The FIRE-II geometry is scaled down in such a manner that the diameter of the nose is of unit dimension.

Figures 4.9 and 4.8 show a comparison of the contour plots showing the flow field and the line plots of various properties along the stagnation line estimated by the BCML model with those predicted by the FVM solver, respectively. Again, the line and contour plots show that the shock stand-off distance predicted by the two methods is in close agreement. Although the shock profile obtained from the BCML model qualitatively matches that predicted by

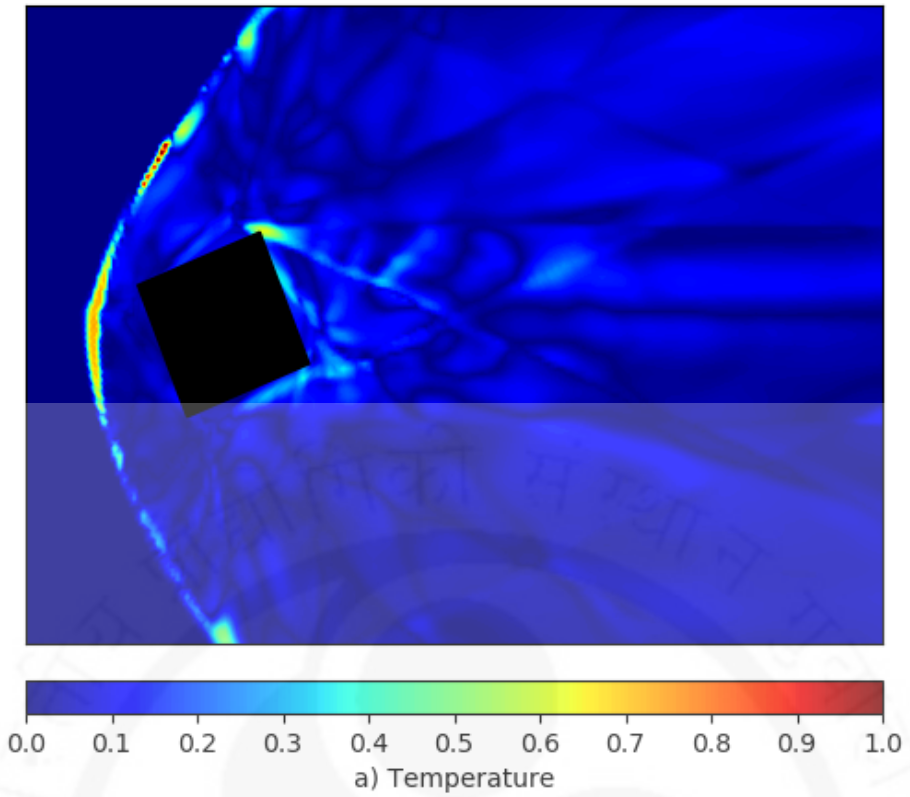

Figure 4.7: Contour plot of relative error in temperature for Mach 4 flow over a square cylinder at 52.6 km with $T_{wall} = 550$ K inclined at 22°.

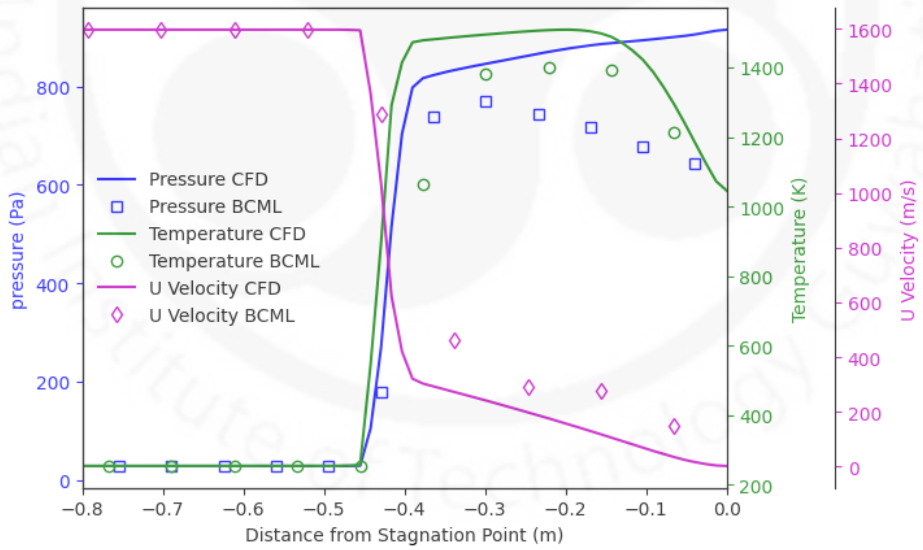

Figure 4.8: Variation of flow field properties (temperature, pressure, and u-velocity) predicted by the S_{16} BCML model and obtained from the FVM solver along the stagnation line for Mach 5 flow over the FIRE-II re-entry vehicle at 59.26 km with $T_{wall} = 1000$ K.

the FVM solver, the shock front exhibits aberrations at distances away from the re-entry geometry, suggesting shortcomings in the BCML model. However, it must be noted that the re-entry geometry is in sharp contrast to the simplistic bluff-body shapes (circular and square cylinders) considered in training the machine learning model. As more data for the hypersonic flow over bluff bodies are available, the model's accuracy will improve, and the

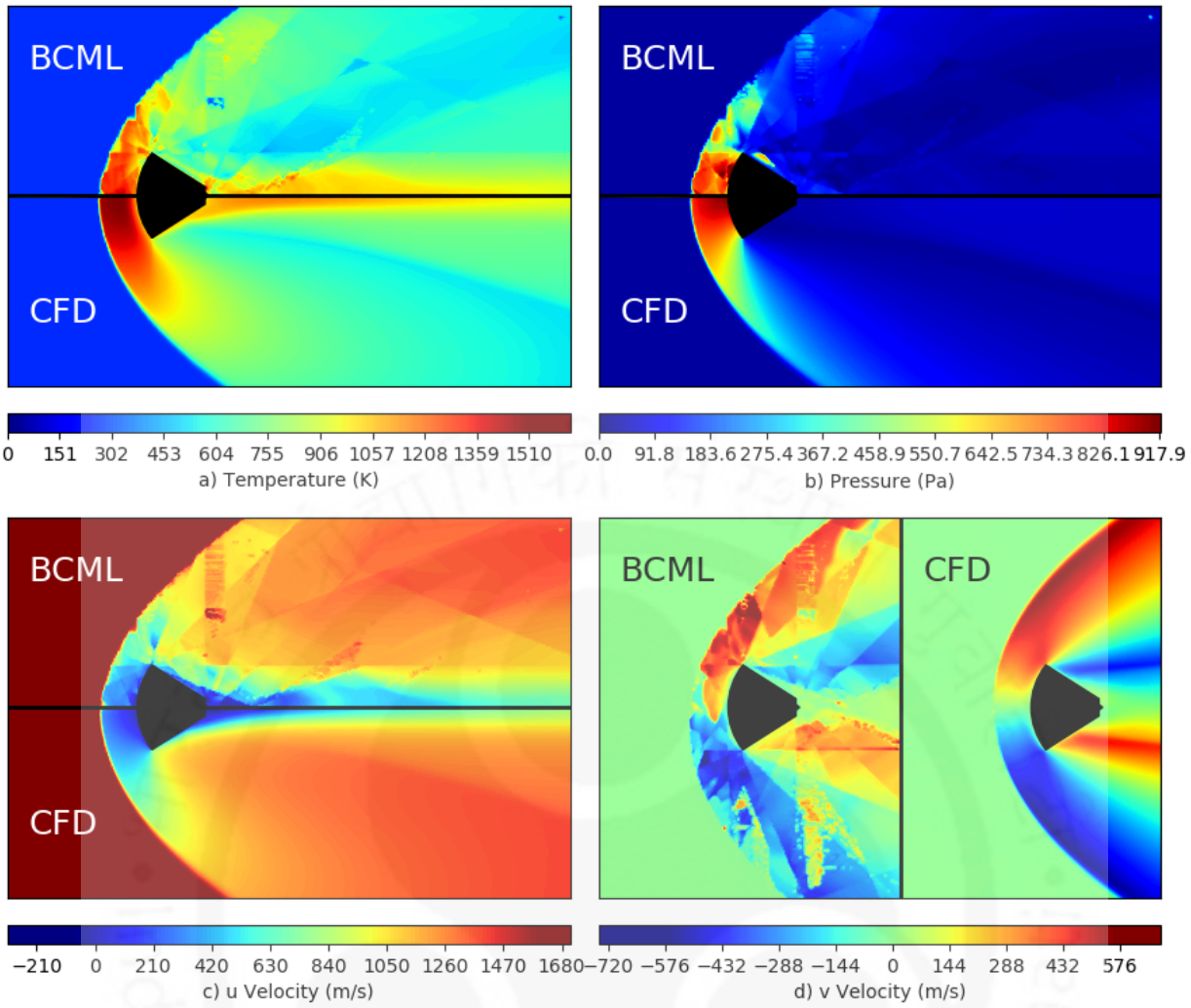

Figure 4.9: Comparison of flow field contour plots (a) translational temperature, b) pressure, c) u-velocity, and d) v-velocity predicted by the S_{16} BCML model and obtained from the FVM solver for Mach 5 flow over FIRE-II re-entry vehicle at 59.26 km with $T_{wall} = 1000$ K

flow field prediction will be closer to those obtained from the FVM solver.

4.2 BCML Framework for Chemical Reacting Flows

4.2.1 Algorithm

Once the BCML algorithm for lower altitudes was developed at frozen conditions, the natural progression was to extend the framework to account for chemical non-equilibrium effects. At higher Mach numbers, the temperatures in the shock and post-shock regions are high ($> 10^3$ K) enough for nitrogen and oxygen bonds to break. The assumption of frozen chemistry is insufficient because chemical reactions (dissociation, recombination, and exchange) significantly change the flow physics. Therefore, employing chemical reaction models is essential for accurately predicting flow fields in hypersonic conditions. Including chemical reactions requires adding species equations to the set of governing equations, expressed in

terms of concentrations or mole fractions. Unlike frozen-flow simulations, species equations are governed by reaction kinetics. The mole fractions directly influence mixture properties such as specific heats, gas constant, viscosity, thermal conductivity, and relaxation rates.

Rather than adding mole fractions directly to the input and output layers of the BCML framework, a decoupled strategy was preferred. In this approach, the BCML DNN model is first employed to predict the flow field, followed by a separate neural network, termed ChemDNN, that predicts the species mole fractions. The ChemDNN model takes the flow field predicted by the BCML DNN as input and predicts species distributions. This decoupled approach offers several advantages. First, it preserves the structure of the original BCML model. Second, it reduces the dimensionality of the BCML model. The combined BCML–ChemDNN framework applies to low-altitude hypersonic flows. It will serve as a test model for the BCML framework for higher-altitude hypersonic flows, where thermochemical non-equilibrium effects are more significant.

4.2.2 Data Generation and Training ML Models

Since the objective is to test the feasibility of the combined and decoupled BCML-ChemDNN framework, flow-field data are generated for Stardust re-entry vehicle geometry only, across a wide range of atmospheric conditions corresponding to altitudes between 40 km and 58 km (40, 45, 50, 55, and 58 km) and freestream velocities from 3.5 km/s to 12 km/s. The flow field is generated using the `OpenFOAM`-based `hy2Foam` solver [102] for an 5-species air chemistry model (N_2, O_2, N, O, NO). A total of 29 simulations are performed, corresponding to approximately 8.8 million training data samples for BCML. An unstructured mesh consisting of approximately 30,393 cells was used. The mesh quality assessment indicates a maximum non-orthogonality of 58.97°, an average non-orthogonality of 11.48°, and a maximum skewness of 0.85, confirming that the computational mesh is suitable for accurate numerical simulations. Although the physical problem is two-dimensional, the simulations employ a one-cell-thick three-dimensional computational domain, which is the standard implementation in OpenFOAM. Table 4.4 lists the `hy2Foam` models employed for the current set of simulations.

After the training data has been generated, the BCML algorithm is trained as follows for the frozen flow condition. Various DNN architectures were tested, and it was found that an input layer followed by four hidden layers with 128 nodes per layer achieved reasonable accuracy and minimal after approximately 200 epochs of training. Figure 4.10 shows the evolution of training and validation loss with epochs, in terms of mean square error (MSE) for the loss function and mean absolute error (MAE) for the S_{16} BCML model, showing smooth convergence and stable performance. The final training and validation losses stabilised at 0.086401 and 0.135345, respectively. The total training time was also in the order of a few minutes ($\sim$ 5 min). The output layer of the BCML model, which are the flow field properties,

Table 4.4: Chemical reaction parameters incorporated in the `hy2Foam` solver for the Stardust simulations

Parameter	Specification
Solver	`hy2Foam`
Gas model	5-species air chemistry model
Chemistry model	Park (1993) [103]
Thermal model	Non-equilibrium, two-temperature
Transport and thermophysical properties	Blottner-Eucken model
Mixing rule	Armaly and Sutton
Diffusion model	Modified Fick's law
Thermal energy exchange	V-T exchange: Millikan-White
Turbulence	None
Wall boundary condition	Non-catalytic wall

serves as the input properties for the ChemDNN model. Here, a total of 5 molecular species were considered as output parameters (N_2, O_2, N, O, NO). The optimised ChemDNN model is shown in the Figure 4.11, enabling smooth convergence and stable performance.

For chemically reacting hypersonic flows, the ChemDNN model employs the flow pressure, translational temperature (in Chapter 4), rotational and vibrational temperature, and the local flow speed as input parameters. The flow speed is defined as the magnitude of the velocity vector,

$$V = \sqrt{u^2 + v^2 + w^2}, \tag{4.4}$$

where u, v, and w denote the velocity components in the three Cartesian directions. The resultant speed is included as an independent input because it represents the translational kinetic energy of the incoming flow. During hypersonic atmospheric re-entry, a significant fraction of this kinetic energy is converted into internal energy across the strong bow shock, resulting in elevated gas temperatures, thermochemical non-equilibrium, and molecular dissociation. Consequently, the local flow speed plays an important role in determining the species composition and improves the predictive capability of the ChemDNN model.

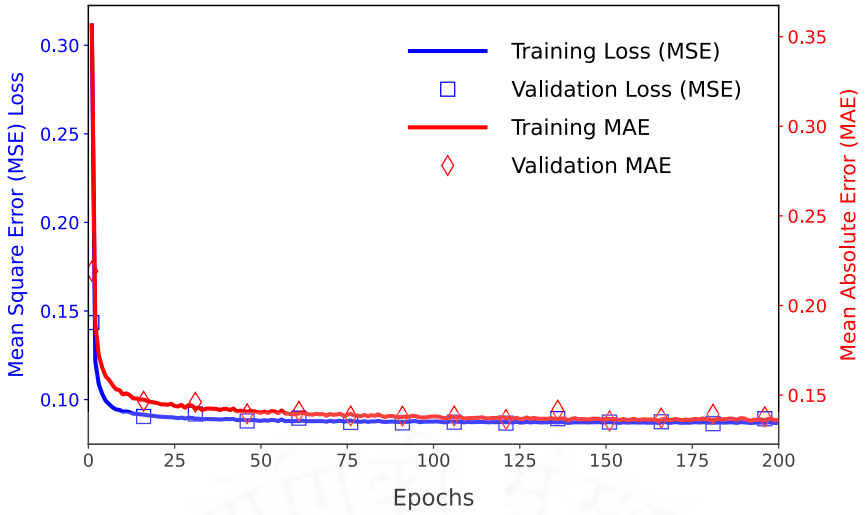

Figure 4.10: Evolution of training and validation loss and MAE with epochs for the BCML model.

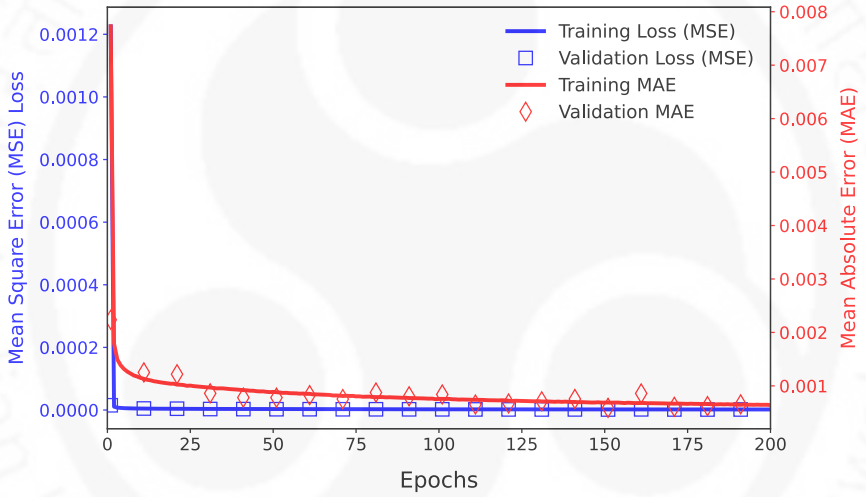

Figure 4.11: Evolution of training and validation loss and MAE with epochs for the ChemDNN model.

4.2.3 Flow Field Prediction using BCML Model

The test case examines a 10 km/s flow under ambient conditions corresponding to an altitude of 52.37 km, with a Stardust geometry. Figure 4.12 presents the variation of properties along the stagnation/baseline of Stardust geometry. The peak vibrational temperature in the post-shock region predicted by the BCML model is 5025.27 K, which is approximately 3.62% lower than the non-equilibrium NSF prediction. It should be noted that the BCML model predicts the entire flow field in less than 10 s, which is much faster than non-equilibrium NSF simulations, which take around 52 hours on a 12-core CPU workstation. The BCML model's accuracy is reasonable, especially given its ability to handle re-entry geometries. The corresponding contour plot for all properties is shown in Figure 4.13.

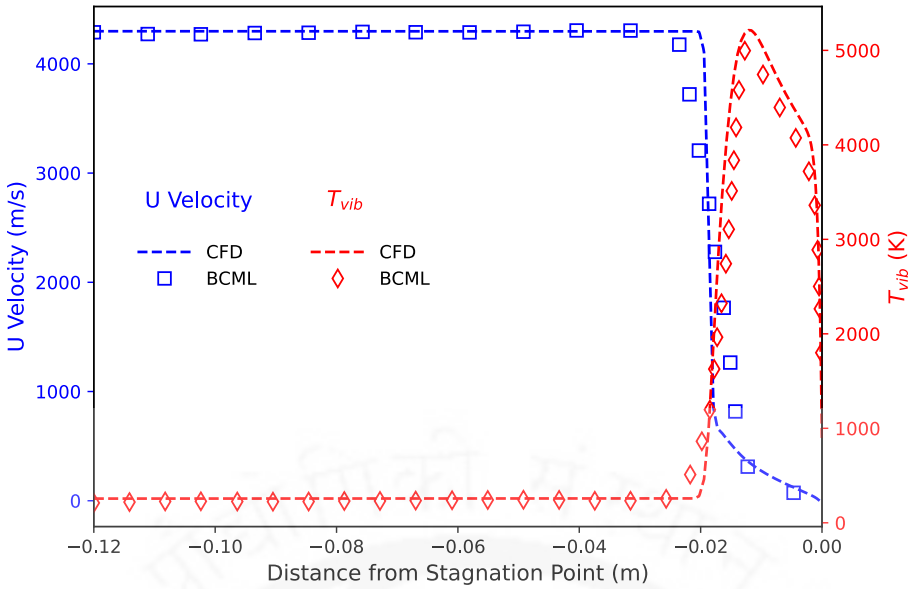

Figure 4.12: Variation of flow field properties (vibrational temperature and u-velocity) predicted by the BCML model and obtained from the `hy2Foam` solver along the stagnation line for 10 km/s flow over a Stardust Geometry at 52.37 km with $T_{cone} = 800$ K.

4.2.4 Mole Fractions Prediction using ChemDNN Model

The primary objective of this study is to assess the ChemDNN model's predictive performance independently. For this purpose, the model is evaluated using flow-field data directly obtained from NSF simulations, and the resulting predictions are systematically compared with the corresponding NSF reference solutions. This methodology enables an unbiased evaluation of the ChemDNN model by decoupling the chemistry prediction from the BCML-based flow-field estimation, thereby ensuring that the reported accuracy reflects the intrinsic capability of the ChemDNN framework.

Figure 4.14 shows the variation of mole fractions of various species along the stagnation line for a freestream velocity of 10 km/s flow over a Stardust geometry at atmospheric conditions corresponding to 52.37 km altitude. The flow consists of molecular Nitrogen (N_2) and Oxygen (O_2) in the pre-shock region, with mole fractions close to their freestream values. In the shock and post-shock regions, dissociation of molecular species becomes significant due to the high temperatures, leading to increased mole fractions of atomic nitrogen (N) and Oxygen (O). This is accompanied by the formation of small amounts of nitric oxide (NO) species through the exchange reactions. The mole fractions of the fifth species (χ_{NO}) can be determined from the unity constraint of the mole fractions. It can be observed from Fig. 4.14 that an excellent agreement is observed for all species. The corresponding mole fractions contour plot for all five species is shown in Figure 4.16.

Similarly, Figure 4.15 presents the variation of different reacting species along the stagnation line for a freestream velocity of 8.25 km/s flow over a Stardust geometry, under

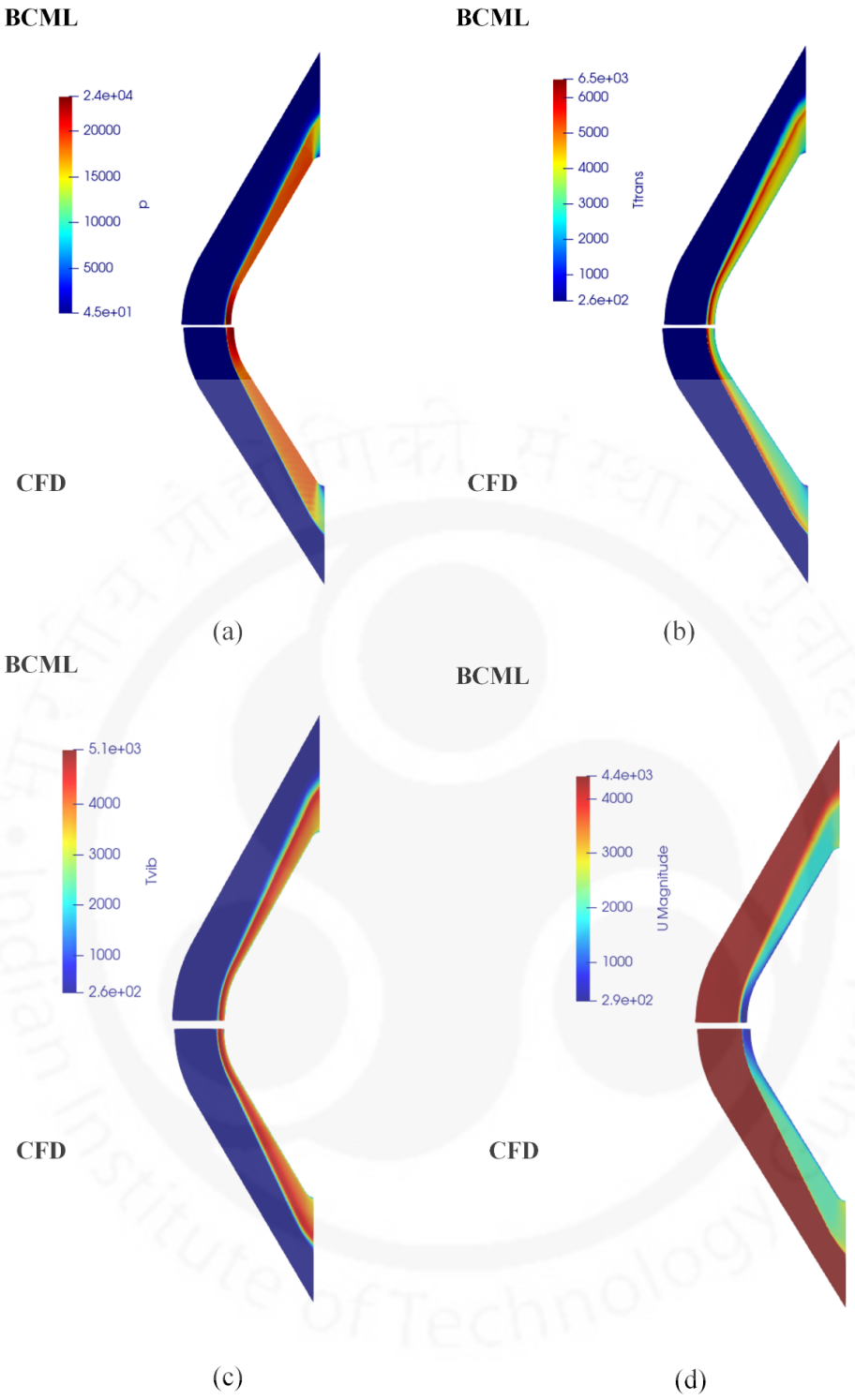

Figure 4.13: Comparison of flow field contour plots (a) pressure, b) translational temperature, c) vibrational temperature and d) velocity magnitude predicted by the S_{16} BCML model and obtained from the CFD (`hy2Foam` solver) for 10 km/s flow over Stardust geometry at 52.37 km with $T_{cone} = 800$ K.

atmospheric conditions corresponding to 56.5 km. Again, excellent agreement is obtained for all species, and the corresponding L_2 norm errors are listed in Table 4.5. Overall, this validation demonstrates that the ChemDNN model can accurately predict mole fractions

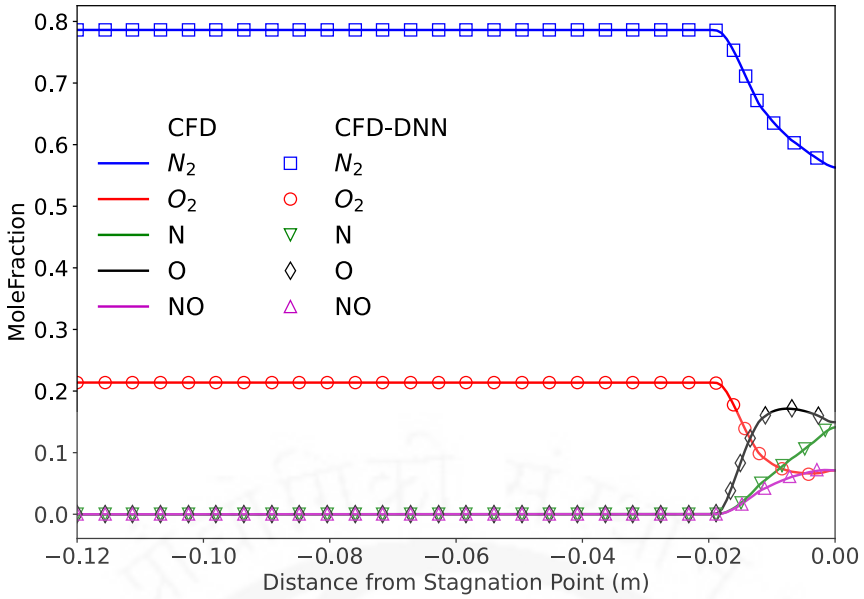

Figure 4.14: Comparison of mole fractions of various species along the stagnation line from the ChemDNN model and CFD simulation for 10 km/s flow at 52.37 km altitude with $T_{cone} = 800$ K.

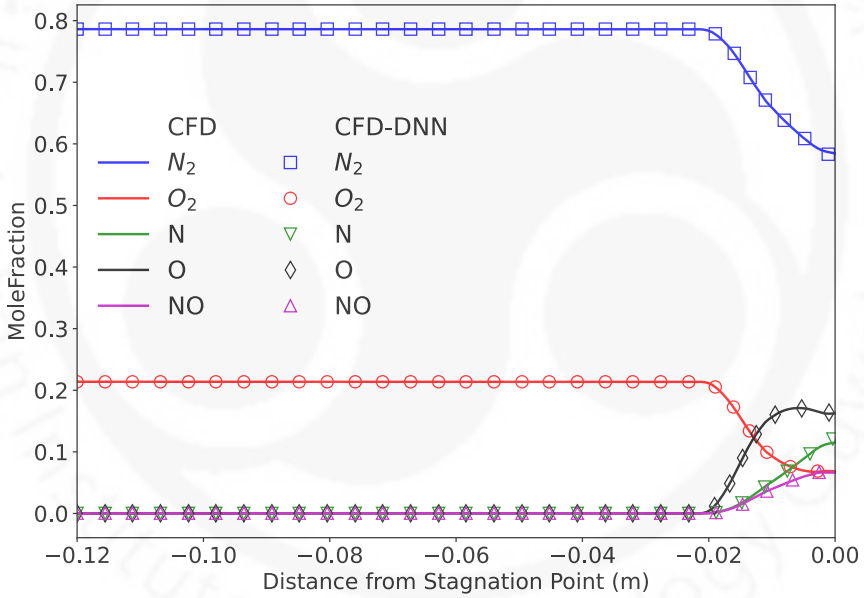

Figure 4.15: Comparison of mole fractions of various species along the stagnation line from the ChemDNN model and CFD simulation for 8.25 km/s flow at 56.5 km altitude with $T_{cone} = 800$ K.

within the domain.

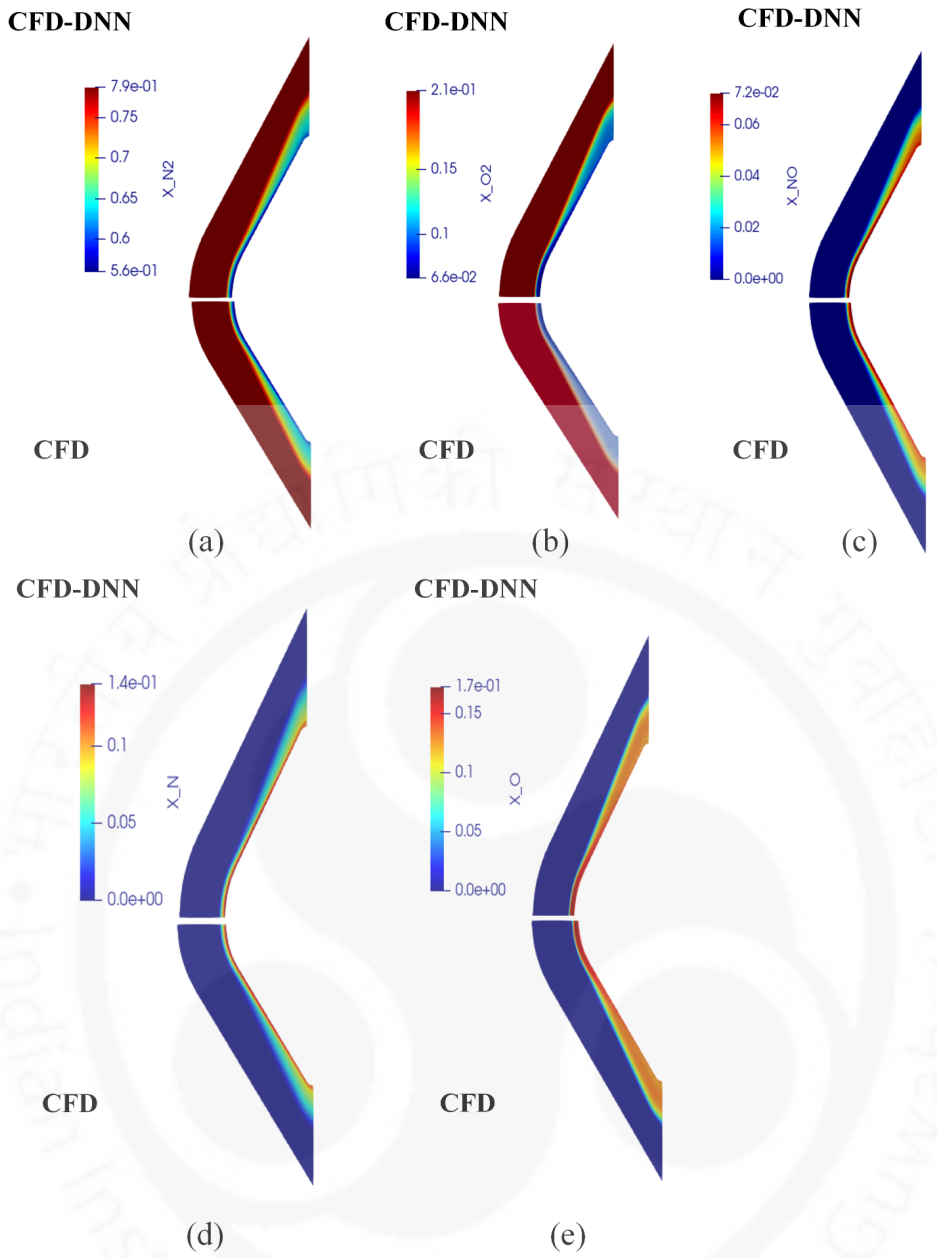

Figure 4.16: Comparison of mole fractions contour plots of species (a) N_2 (b) O_2, (c) NO, (d) N and (e) O predicted by the ChemDNN (CFD-DNN) model and the CFD simulation (`hy2Foam` solver) for 10 km/s flow over a circular cylinder at 52.37 km with T_{cone} =800 K.

Table 4.5: Initial and boundary conditions for validation cases for testing the ChemDNN model, along with error analysis.

Parameter	Altitude = 52.37 km	Altitude = 56.5 km
Pressure (Pa)	44.822	25.492
Temperature (K)	252.71	245.37
Inlet Velocity (km/s)	10.0	8.250
L_2 error (N_2)	0.074%	0.049%
L_2 error (O_2)	0.184%	0.180%
L_2 error (N)	1.016%	1.451%
L_2 error (O)	1.291%	2.045%

Chapter 5

High-Altitude DSMC-Based BCML Model

The BCML framework was initially developed for low-altitude conditions using flow-field data from the in-house `hy2Foam` NSF solver. As pointed out in previous chapters, the continuum assumption remains valid, and thermodynamic equilibrium is reasonably satisfied at lower altitudes. The applicability of the BCML framework was then extended to higher altitudes, where the flow transitions into the rarefied regime. Under these conditions, rarefaction effects and thermodynamic non-equilibrium become significant because the continuum assumption breaks down. In such cases, the direct simulation Monte Carlo (DSMC) method is used to study the flow physics.

5.1 Algorithm

The overall concept of the BCML algorithm remains unchanged. One of the challenging aspects from a data-generation perspective is the computational cost of DSMC simulations. A typical DSMC simulation with ambient conditions corresponding to an altitude of 72 km, using approximately 10^5 cells, required a runtime of the order of 10^3 CPU hours on a parallel workstation to achieve reasonably scatter-free results. It was found that the computational cost was at least 10-100 times higher than that of the NSF simulations used to generate data for the low-altitude BCML model. The increased computational cost limited the number of simulations that could be run for the high-altitude DSMC-based BCML model relative to the previous one. The high dimensionality in the low-altitude BCML model (144 inputs for the S_{16} model) was reduced to ensure that the training of the new high-altitude BCML model is sufficient. Unlike the low-altitude BCML model, the number of parameters describing boundary conditions was reduced to only 4 per generator, resulting in a total of 80 inputs ($16 \times (4 + 1)$). Instead of considering the boundaries as Dirichlet or Neumann, these were treated as either far-field or wall boundaries. The generators emanating from a point in the domain either intersect the re-entry geometry or the outer boundary of the domain. The latter are treated as far-field boundary conditions, with the ambient pressure,

ambient temperature, and flow velocities resolved into normal and tangential components serving as the boundary-condition parameters. In the case of generators intersecting the re-entry geometry, the post-shock pressure, calculated using the Rankine-Hugoniot relations for frozen conditions, the surface temperature, and zero speed in both the normal and tangential directions, constitutes the four input parameters. In this way, each generator is characterised by five parameters rather than the nine used in the development of the low-altitude NSF-based BCML framework.

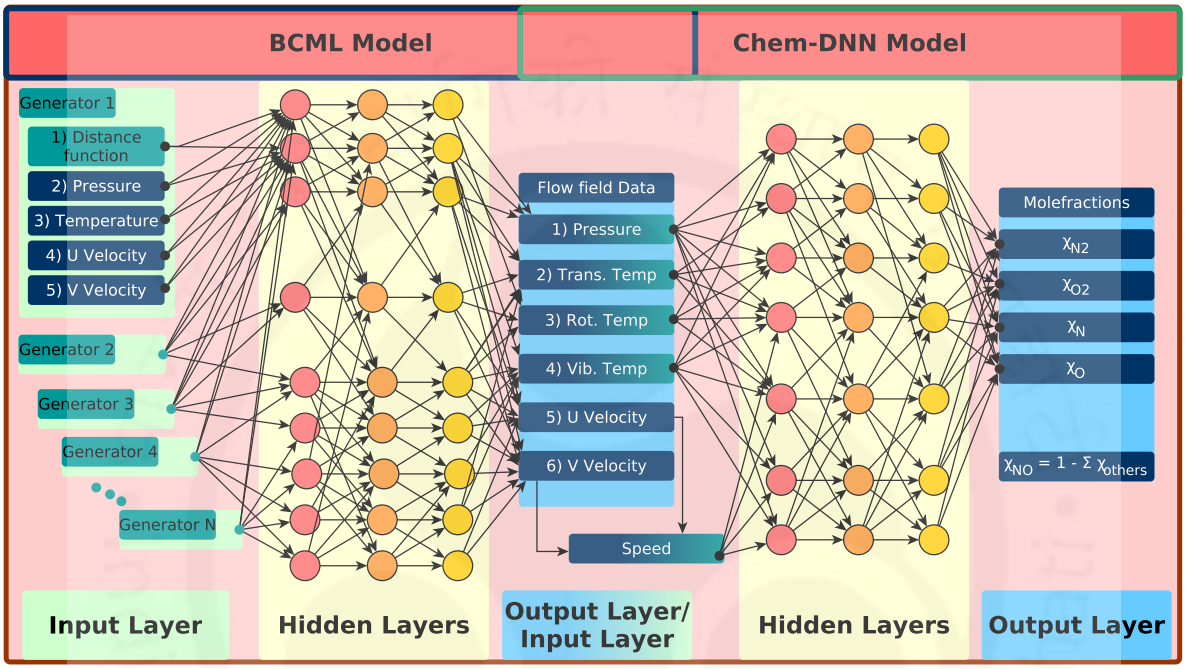

Figure 5.1: Flowchart of the high-altitude DSMC-based BCML and ChemDNN model.

The hypersonic flows at these high altitudes generally have a high degree of non-equilibrium, and it is not possible to ignore the multi-temperature effects. The DSMC simulations predict three temperatures corresponding to translational, rotational, and vibrational modes of energy. The DSMC method can capture thermodynamic non-equilibrium by including collision models for inelastic energy exchange. The new DSMC-based BCML model can predict flow fields across all three temperature modes. In total, six outputs: pressure, three temperatures, and two velocity components, are predicted by the new BCML model for high altitude simulations. The boundary conditions at the far field and re-entry surface are assumed to be in equilibrium, thereby allowing the three temperature modes to be characterised by a single temperature as an input parameter.

Another feature of the high-altitude DSMC-based BCML framework is the inclusion of chemical reactions. A gas in a realistic hypersonic re-entry flow undergoes dissociation across the shock wave. In the current model, 5-species air chemistry is employed in the DSMC simulations. For flows at very high Mach numbers ($Ma \gg 15$), accurate modelling

requires accounting for ionisation reactions and electron transport; however, these effects are not included in the current study. As discussed in the previous chapter, one option was to include mole fractions or number density as additional outputs from the BCML model. However, this would require additional parameters per generator as inputs to the BCML model for each of the species. Instead, a separate DNN model, again referred to as the ChemDNN model, was trained using the three temperatures, pressure, and speed in each cell as inputs, with four of the five species comprising the output layer. The mole fractions of the fifth species (χ_{NO}) can be determined from the unity constraint of the mole fractions. The difference between the ChemDNN models for low and high altitudes is the number of temperature inputs. In the low-altitude BCML model, trans-rotational and vibro-electronic temperatures were used, whereas in the high-altitude models, translational, rotational, and vibrational temperatures are employed. The three temperatures and the speed represent the thermal energy, internal energy (both rotational and vibrational modes), and kinetic energy, respectively. In addition to these energy terms, pressure influences the collision frequency and is therefore included as an input parameter in the ChemDNN architecture. The outputs generated by the BCML and ChemDNN frameworks are further post-processed to reconstruct thermophysical-property contours, species mole-fraction distributions, stagnation-line profiles, and surface-property variations for comparison with DSMC simulations. Figure 5.1 shows the overall DNN framework, where the BCML model predicts thermophysical properties, which are then used as inputs to the ChemDNN model to estimate mole fractions.

5.2 Data Generation and Training ML Models

The BCML model was now extended for rarefied hypersonic flows [104] for a wide range of re-entry speeds (Ma $\in [8, 35]$) and altitudes ($\in$ 60–84 km) using training data generated using the in-house DSMC code. The objective of the current work is to accelerate direct simulation Monte Carlo (DSMC) computations by generating optimised meshes using Delaunay triangulation and adaptive mesh refinement based on flow-field predictions from the BCML model at higher altitudes. Towards this, a new BCML model is reported to predict the local mean free path in the flow domain, along with other macroscopic properties around bluff bodies, under frozen rarefied hypersonic conditions at altitudes ranging from 60 to 84 km. The training data for the proposed BCML model are generated using an in-house DSMC code that simulates over a broad range of ambient conditions and Mach numbers. By integrating the newly developed high-altitude BCML model with the previously established low-altitude NSF-based BCML framework, a unified machine learning methodology is achieved for predicting the flow field around a re-entry vehicle over nearly the entire trajectory. The DSMC simulations encompass a diverse set of bluff-body geometries, including circular cylinders, square cylinders at multiple angles of attack, and several arbitrarily shaped configurations. The arbitrary geometries are constructed from random combinations of circular arcs and

straight line segments and are inscribed within a unit-radius circle to ensure geometric consistency. The inclusion of arbitrary geometries was done based on prior experience in developing the low-altitude NSF-based framework. Representative examples of the selected geometries used in the DSMC simulations are shown in Fig. 5.2.

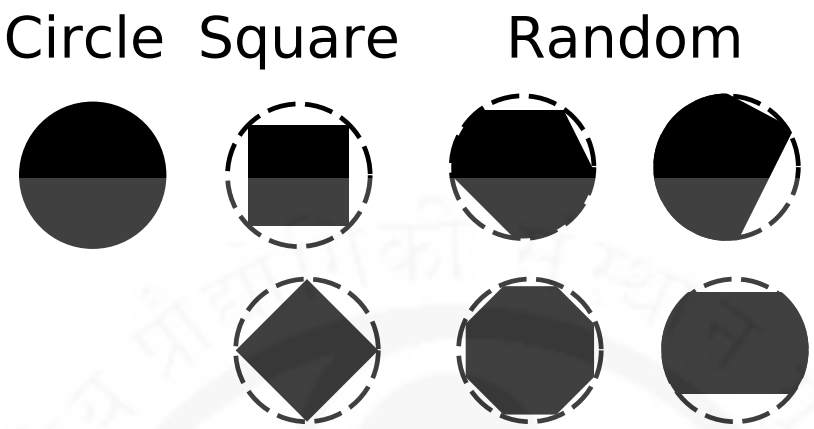

Figure 5.2: re-entry vehicle shapes used as bluff bodies in hypersonic flow simulation for training data generation.

The computational domain considered is 13 m long and 9 m wide. The surface temperature of the bluff body is assumed to be constant at 1000 K for all cases. Simulations for flow past a square cylinder are performed under various atmospheric conditions corresponding to altitudes ranging from 60 km to 84 km (i.e., 60, 66, 72, 78, and 84 km) and Mach numbers ranging from 8 to 35 (i.e., 8, 15, 25, and 35) using the DSMC method. Table 5.1 summarises the altitude-dependent ambient pressure, temperature, and composition data [65]. The present study considers four different angles of attack: 0°, 15°, 30°, and 45° for the square bluff body. For each fixed angle of attack, simulations are conducted for 20 flow conditions corresponding to four Mach numbers at five altitudes. Consequently, 80 simulations are performed for the square geometry. In simulations of circular cylinders and arbitrarily shaped geometries, atmospheric conditions and Mach numbers were randomly sampled within the specified maximum and minimum bounds. A total of 36 circular cylinder geometry simulations are performed. To ensure the model's generalizability, 38 additional simulations with arbitrarily shaped geometries are included in the training data, bringing the total to 154. The term "arbitrarily shaped geometry" refers to geometries constructed from random combinations of circular arcs and straight-line segments, inscribed within a unit-radius circle. In total, 18.724 million cell data have been generated for all three geometrical bluff bodies with a total computational cost of approximately 0.17 million CPU hours on an Intel Xeon workstation.

The physical and chemical models employed in the DSMC simulations are summarised as follows. The DSMC solver employs 5-species air chemistry model consisting of N_2, O_2, O, N, and NO. The collision dynamics for each of the species are handled using the variable hard

Table 5.1: Ambient conditions at various altitudes for High Altitude DSMC-based simulations

Altitude (km)	T_∞ (K)	ρ_∞ (kg/m^3)	P_∞ (Pa)	X_{N_2}	X_{O_2}
60	235	2.091×10^{-4}	14.109	0.794	0.206
66	283.7	1.543×10^{-4}	12.566	0.762	0.237
72	227.9	8.048×10^{-5}	5.264	0.762	0.237
78	194.3	3.273×10^{-5}	1.825	0.762	0.237
84	182.9	1.149×10^{-5}	0.603	0.762	0.237

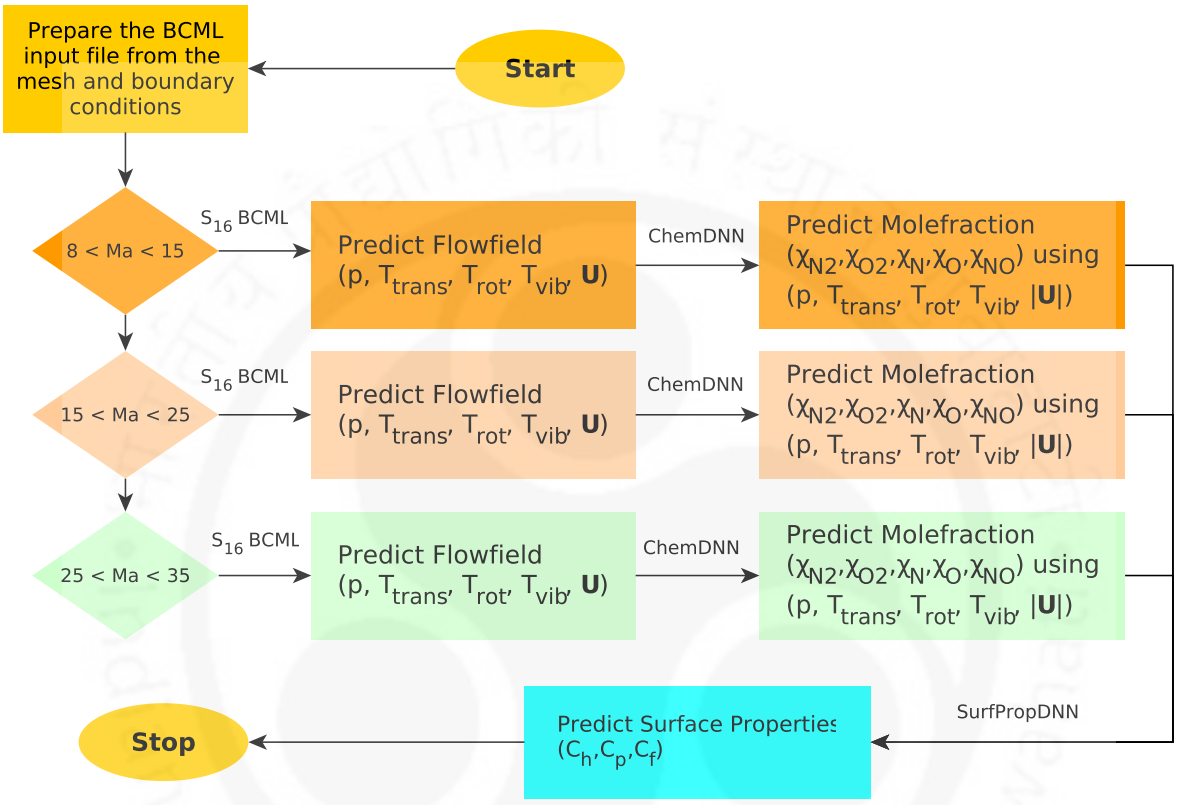

Figure 5.3: Workflow illustrating the procedure for obtaining flow/surface properties for hypersonic flows at rarefied conditions.

sphere (VHS) collision model, characterised by the standard reference diameter, reference temperature, and temperature exponent (ω) parameters [14], which capture the temperature dependence of the molecular collision cross-section. Internal energy exchange processes are modelled through constant rotational and vibrational relaxation parameters. For diatomic molecules (N_2, O_2, and NO), two rotational degrees of freedom are assigned with a fixed rotational relaxation number of $Z_{\text{rot}} = 5$, while vibrational excitation is modelled using characteristic vibrational temperatures (e.g., 3371 K for N_2, 2256 K for O_2 and NO) and a constant vibrational relaxation number of $Z_{\text{vib}} = 50$. These assumptions are consistent with routine DSMC simulations using the Larsen-Borgnakke (LB) inelastic collision model. The reactions are modelled using the total collision energy (TCE) approach using chemical reaction parameters tabulated in Table 5.2. The simulation employs a time step of 4×10^{-7} s, with approximately seven simulated particles per cell. Overall, the chosen set of collision, reaction, and internal energy models provides a physically representative yet computationally

efficient framework for simulating chemically reacting, non-equilibrium air flows relevant to high-enthalpy gas dynamics and atmospheric re-entry environments. Figure 5.3 illustrates the overall workflow for estimating the flow field and surface properties of an arbitrarily shaped re-entry geometry.

Table 5.2: Chemical reactions and corresponding reaction rate coefficients following the modified Arrhenius equation, $k = AT^{\eta}\exp(-E_a/(k_bT))$.

Reaction	E_a (J)	A	η
$N_2 + N_2 \rightarrow 2N + N_2$	15.67×10^{-19}	4.1×10^{-12}	-0.62
$N_2 + O_2 \rightarrow 2N + O_2$	15.67×10^{-19}	1.5×10^{-11}	-0.68
$N_2 + NO \rightarrow 2N + NO$	15.67×10^{-19}	1.5×10^{-11}	-0.68
$N_2 + N \rightarrow 3N$	15.67×10^{-19}	1.0×10^{-11}	-0.68
$N_2 + O \rightarrow 2N + O$	15.67×10^{-19}	4.0×10^{-12}	-0.54
$O_2 + N_2 \rightarrow 2O + N_2$	8.197×10^{-19}	1.3×10^{-10}	-1.00
$O_2 + O_2 \rightarrow 2O + O_2$	8.197×10^{-19}	5.3×10^{-11}	-1.00
$O_2 + NO \rightarrow 2O + NO$	8.197×10^{-19}	1.1×10^{-10}	-1.00
$O_2 + N \rightarrow 2O + N$	8.197×10^{-19}	1.1×10^{-10}	-1.00
$O_2 + O \rightarrow 3O$	8.197×10^{-19}	1.5×10^{-10}	-1.05
$NO + N_2 \rightarrow N + O + N_2$	10.43×10^{-19}	2.1×10^{-10}	-1.00
$NO + O_2 \rightarrow N + O + O_2$	10.43×10^{-19}	2.0×10^{-10}	-1.00
$NO + NO \rightarrow N + O + NO$	10.43×10^{-19}	1.0×10^{-10}	-1.00
$NO + N \rightarrow N + O + N$	10.43×10^{-19}	4.0×10^{-10}	-1.10
$NO + O \rightarrow N + O + O$	10.43×10^{-19}	4.0×10^{-10}	-1.10
$N_2 + O \rightarrow NO + N$	5.18×10^{-19}	8.0×10^{-17}	0.00
$O_2 + N \rightarrow NO + O$	0.2×10^{-19}	4.0×10^{-15}	-0.39
$NO + N \rightarrow N_2 + O$	0.2×10^{-19}	5.0×10^{-16}	-0.35
$NO + O \rightarrow O_2 + N$	2.18×10^{-19}	2.3×10^{-19}	-0.50

The flow-field data, along with surface data from the geometries, were generated using the in-house DSMC code and used to train the ML models. Three ML models have been trained and tested for complete hypersonic flow analysis at high-altitude conditions. These include the BCML model, which uses the generator length from each DSMC cell as input, along with parameters for the boundary conditions of pressure, temperature, and the two components of velocity. Additionally, the BCML model was divided into three overlapping DNN models based on Mach number ranges (S_{16} Mach 8-15, S_{16} Mach 15-25, and S_{16} Mach 25-35 BCML models) to enhance prediction accuracy. This approach improved accuracy and reduced overall loss across each model. Various DNN architectures were tested, and it was found that a four-layer DNN with 256 nodes per layer achieved reasonable accuracy and minimal loss across all BCML models after approximately 500 epochs of training. Figure 5.4 shows the evolution of training and validation loss with epochs, in terms of mean square

error (MSE) for the loss function and mean absolute error (MAE) for the S_{16} Mach 25-35 BCML model, showing smooth convergence and stable performance. The final training and validation losses stabilised at 0.00263 and 0.00297, respectively.

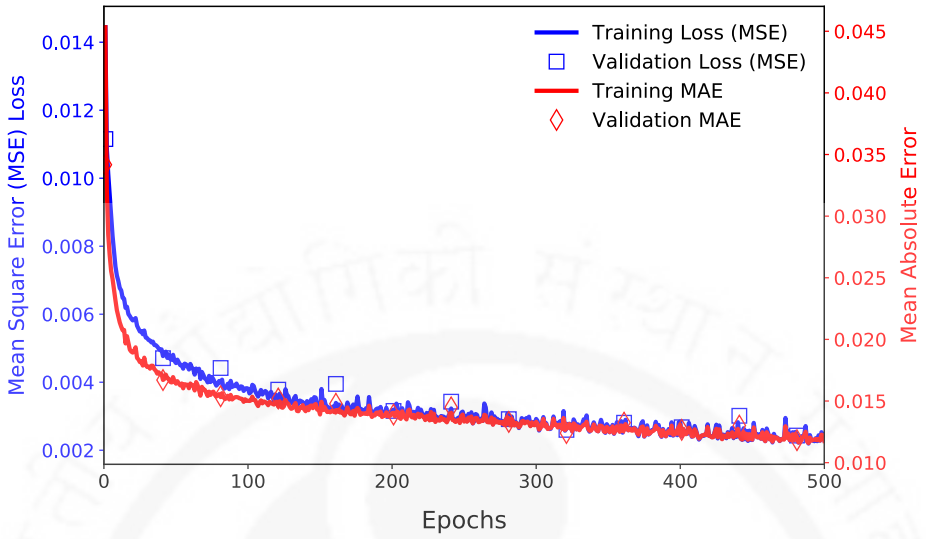

Figure 5.4: Evolution of training and validation loss with epochs for the S_{16} Mach 25-35 BCML model.

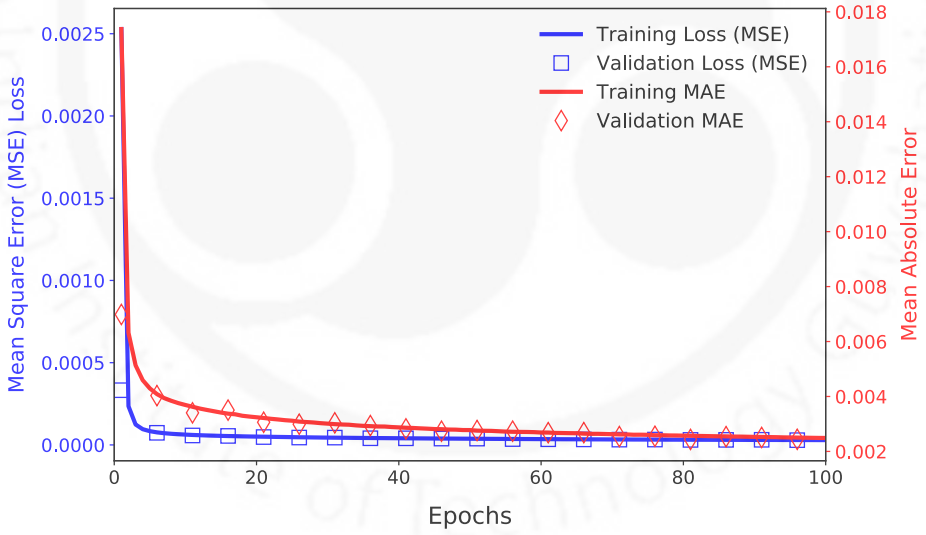

Figure 5.5: Evolution of training and validation loss with epochs for the Mach 25-35 ChemDNN model.

The second model is also related to flow-field prediction in terms of the mole fractions of various species. The flow properties predicted by the BCML model serve as inputs to the second model, the ChemDNN model. The ChemDNN architecture comprises an input layer with five nodes (pressure, three modes of temperature, and speed), three hidden layers of 128 nodes each, and a four-node output layer for the mole fractions of molecular and atomic Nitrogen and Oxygen. The mole fraction of NO can be calculated from the summation law of mole fractions. Similar to the BCML model training, three separate DNN models were

trained for Mach intervals of 8-15, 15-25, and 25-35.

As shown in Fig. 5.5, the training loss values progressively decrease, indicating effective training. It is worth noting that, as a model, ChemDNN achieves higher accuracy (approximately 99%) than the BCML model, which is trained on data for arbitrary geometric shapes. In this way, ChemDNN is a simple DNN that maps one set of flow-field properties obtained from DSMC simulations to another. A minimal difference between training and validation loss (below 0.000011) indicates that the model is neither under-fitting nor over-fitting.

In addition to predicting the flow-field properties and species concentrations, a separate deep neural network, referred to as the SurfPropDNN model, has been developed to estimate the aerodynamic surface properties of re-entry vehicles. The objective of this model is to directly predict engineering quantities of interest, such as the surface pressure coefficient and heat-flux coefficient, without requiring additional post-processing of the DSMC solution. A detailed discussion of this is presented later in this chapter, as illustrated in Figure 5.20.

Finally, a third model, SurfPropDNN, is trained to predict surface properties on the re-entry geometry as a function of angle from the stagnation point and the distance of the point on the surface from the centre of the circumscribed circle. The convergence plots in Fig. 5.6 confirm that the SurfPropDNN achieves high accuracy with negligible overfitting across training epochs.

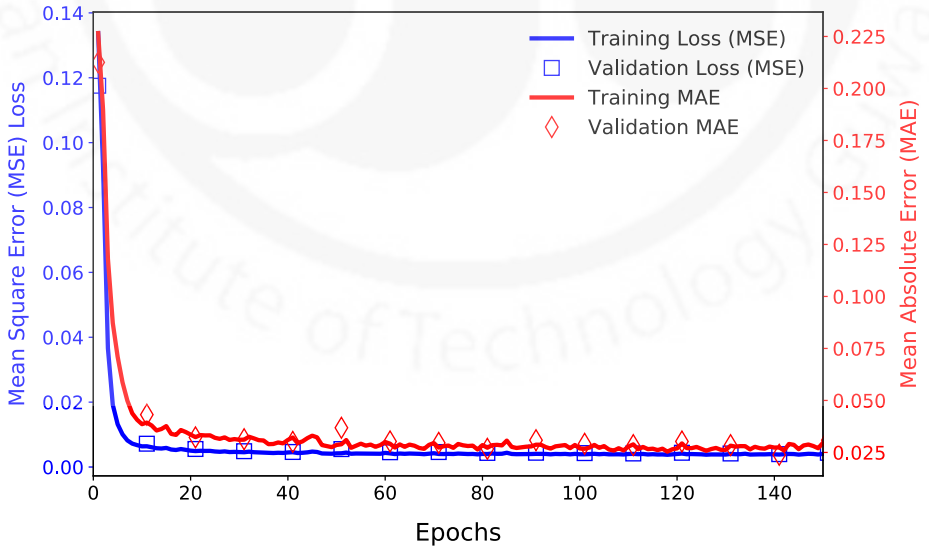

Figure 5.6: Evolution of training and validation loss with epochs for the Mach 25-35 SurfPropDNN model

5.3 Flow Field Prediction using BCML Models

The optimised ML models were employed to predict the flow field and surface properties for a wide range of rarefied hypersonic flows past geometries of different shapes and at various ambient conditions. The inlet conditions of the gas are calculated by selecting ambient conditions at a randomly sampled altitude, along with a randomly chosen Mach number for the hypersonic flows. The choice of the BCML, ChemDNN, and SurfPropDNN models is directly related to the flow's Mach number. The testing of these ML models was conducted in a sequential manner: first, validating the flow field for general thermophysical properties using the S_{16} BCML models; and second, validating the ChemDNN model using the DSMC flow field as input data. Finally, a complete analysis of hypersonic flow over an arbitrarily shaped geometry is discussed.

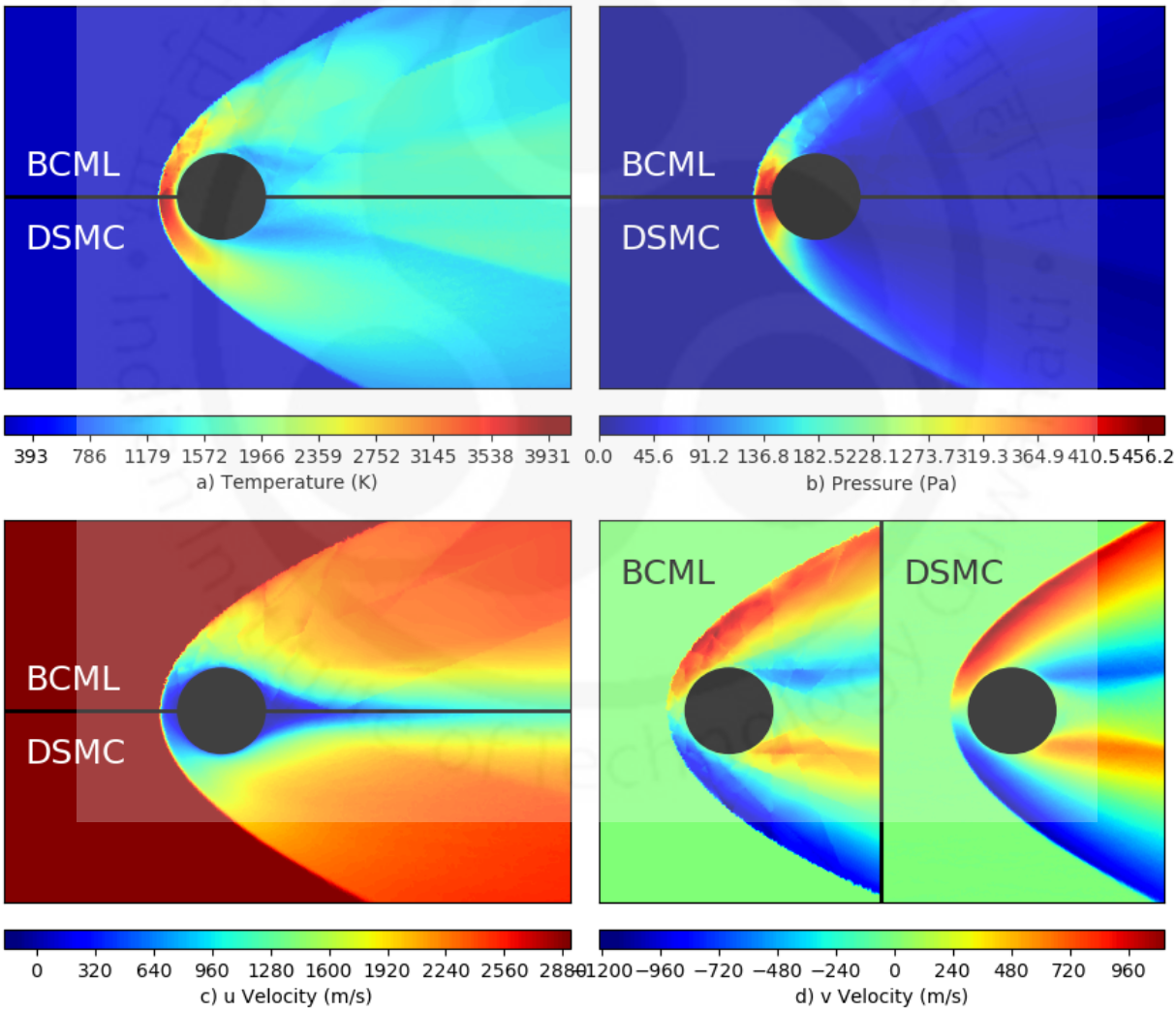

Figure 5.7: Comparison of flow field contour plots of (a) translational temperature, (b) pressure, (c) u-velocity, and (d) v-velocity from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with T_{wall} =1000 K.

The first test case considers a Mach 10 flow at ambient conditions corresponding to

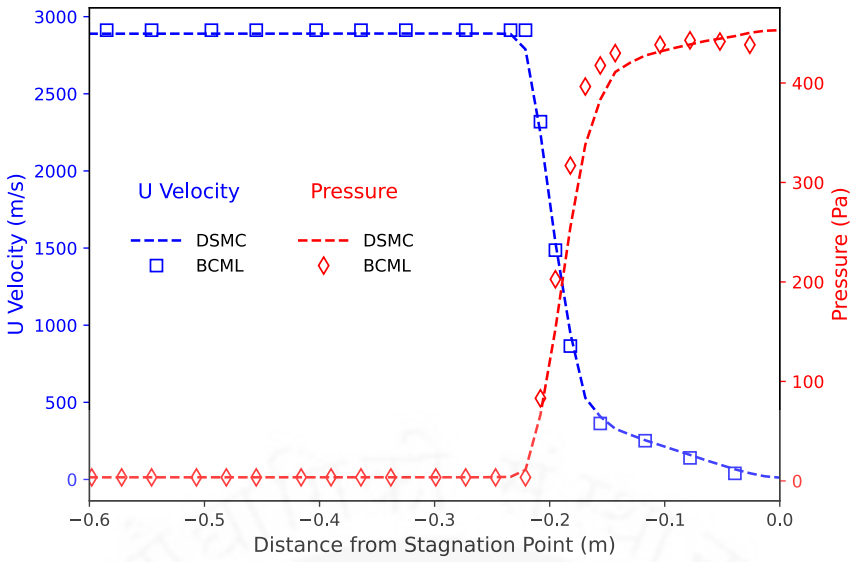

Figure 5.8: Variation of flow field properties (temperature, pressure, and u-velocity) predicted by the BCML model and obtained from the DSMC solver along the stagnation line for Mach 11 flow over a circular cylinder at 75 km with $T_{wall} = 750$ K.

an altitude of 75 km. The bluff body shape chosen for the first test was a simple circular cylinder geometry with a constant surface temperature of 1000 K. Even though the shape of the bluff body is simple, it should be noted that the flow field data estimated using the DSMC code at the given inlet conditions are not included in the training data for the S_{16} Mach 8-15 BCML model. Figure 5.7 compares flow field contour plots of pressure, temperature (T_{trans}), and the two components of velocity generated from the S_{16} Mach 8-15 BCML model and those obtained from the DSMC simulation. The corresponding line plots have been depicted in Fig. 5.8 and 5.9, respectively. It can be observed that, qualitatively, the wave features observed in the flow field generated from the S_{16} Mach 8-15 BCML model are similar to those observed in the contour plots generated by the DSMC code.

Generally, capturing the shock front is crucial for studying hypersonic external flows. The shock stand-off distance from the BCML model was 0.247 m from the stagnation point. In comparison, the shock stand-off distance obtained from the DSMC simulation was 0.221 m. The shock stand-off distances predicted from the BCML model match those obtained from the DSMC simulation. The comparison of the shock front profile and the shock strength obtained from the two models is shown in the centreline plot in Fig. 5.8 and 5.9. The horizontal axis marks the distance along the stagnation line, with 0 corresponding to the stagnation point on the circular cylinder. The peak translational temperature in the post-shock region predicted by the BCML model is 3824.2 K, which is about 0.673% higher than that obtained from the DSMC simulation (3798.69 K). Additionally, the peak pressure in the post-shock region predicted by the BCML model is about 446.85 Pa, which is about 1.359% lower than that obtained from the DSMC simulation (453.01 Pa).

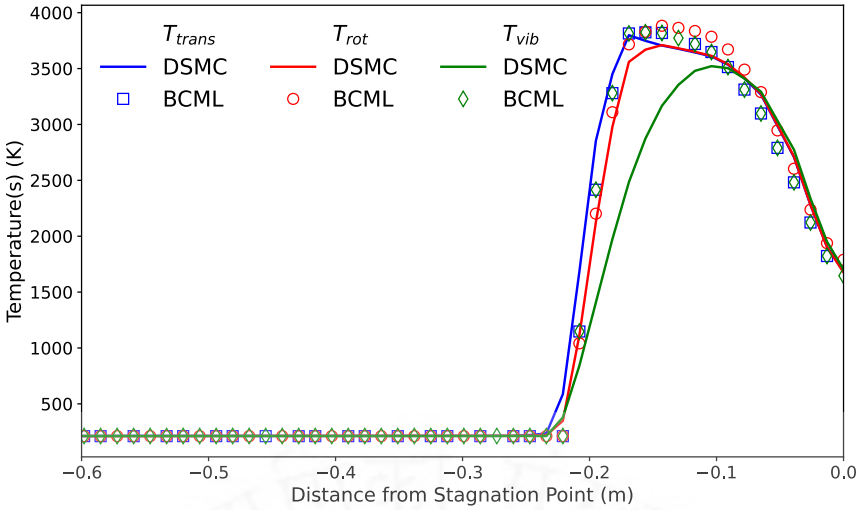

Figure 5.9: Comparison of various temperatures (translational, rotational, and vibrational) along the stagnation line from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 1000$ K.

The BCML model accurately captures the shock front, particularly in regions near the bluff body. However, it must be noted that the shock front obtained from the BCML, away from the bluff body and near the far-field boundaries, loses its profile and is less accurate compared to the DSMC results. The contour plots of the horizontal and vertical components of the velocity also show that the flow field reconstruction by the BCML model in the recirculation zone in the immediate wake region of the bluff body matches well with those predicted by the DSMC code. Furthermore, the BCML model could capture features such as the viscous boundary layer, Prandtl-Meyer expansion, and recompression shocks. However, the BCML model does not accurately capture the viscous core and free shear layer. This conclusion is based on a direct comparison of the BCML predictions with the corresponding DSMC contour plots. While the BCML model accurately captures the bow shock and the overall flow field, relatively larger deviations are observed in the wake region corresponding to the viscous core and free shear layer, where strong gradients and non-equilibrium effects are dominant.

The second test case considers a Mach 9 flow at ambient conditions corresponding to an altitude of 67.5 km. The bluff body chosen here was an arbitrarily shaped geometry. Again, the shape of the geometry is not included in the training data for the S_{16} Mach 8-15 BCML model. Figures 5.11 and 5.12 show the centreline plots of various properties in front of the arbitrarily shaped bluff body. Again, the shock stand-off distance predicted by the BCML model closely agrees with that predicted by the DSMC code. Additionally, the peak temperature indicated by the BCML model is within 5.2% accuracy compared to the peak temperature obtained from the DSMC code. The corresponding contour plot is shown in Figure 5.10. The error in the flow field predicted by the BCML model is higher at a few discrete points at the edge of the bow shock and in the wake region far downstream from the

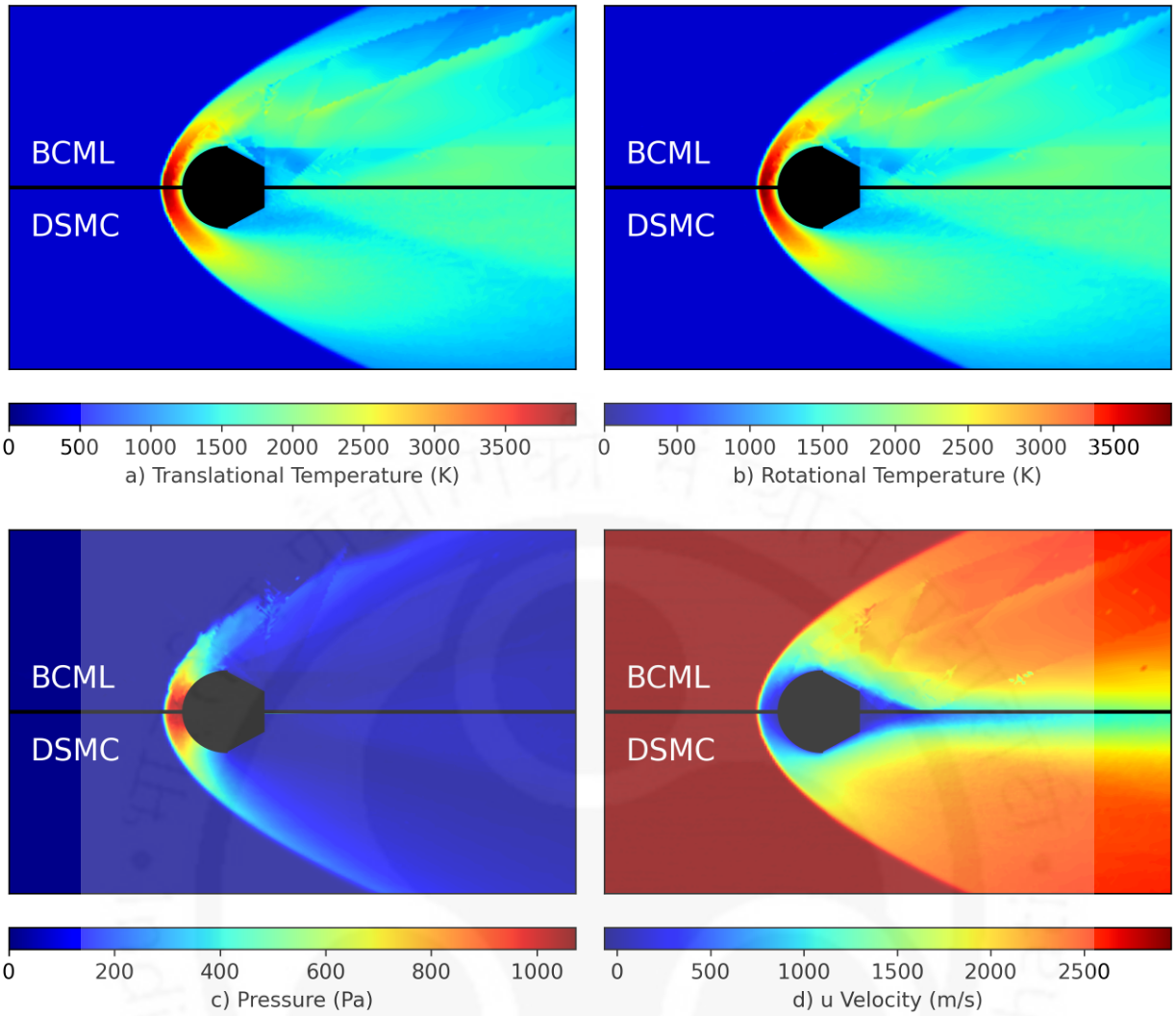

Figure 5.10: Comparison of flow field contour plots of (a) translational temperature, (b) rotational temperature, (c) pressure and (d) u-velocity from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 9 flow over an arbitrarily shaped geometry at 67.5 km with $T_{wall} = 1000$ K.

bluff body. The L_2 norm for the entire domain was found to be less than 2% for temperature and less than 5% for pressure. The error norms for all the test cases are tabulated in Table 5.3.

5.4 Mole Fractions Prediction using ChemDNN Models

The primary objective of this study is to evaluate the ChemDNN model's performance independently. The model is tested using the flow field directly obtained from DSMC simulations, and its predictions are compared with the corresponding DSMC results. This approach enables an independent assessment of ChemDNN's predictive capability, ensuring that the model's performance is evaluated without coupling the flow-field prediction from the BCML model.

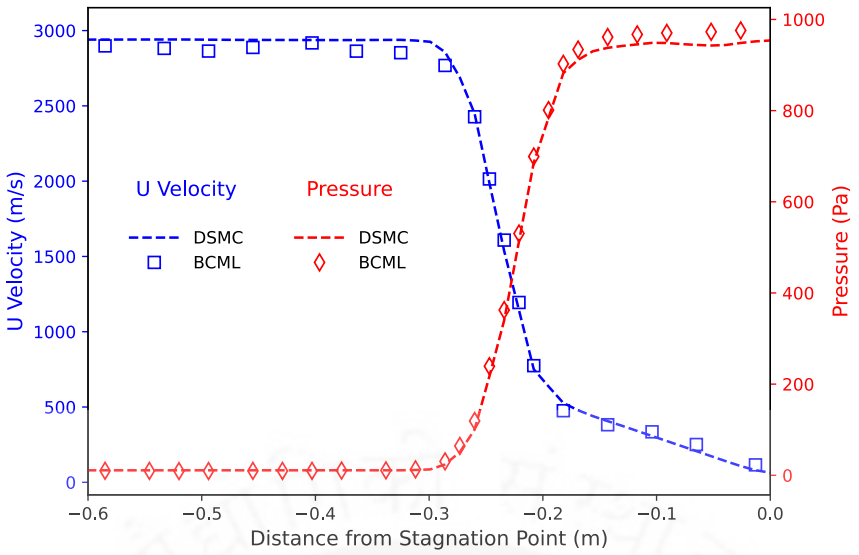

Figure 5.11: Comparison of pressure and u-velocity along the stagnation line from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 9 flow over an arbitrarily shaped geometry at 67.5 km with $T_{wall} = 1000$ K.

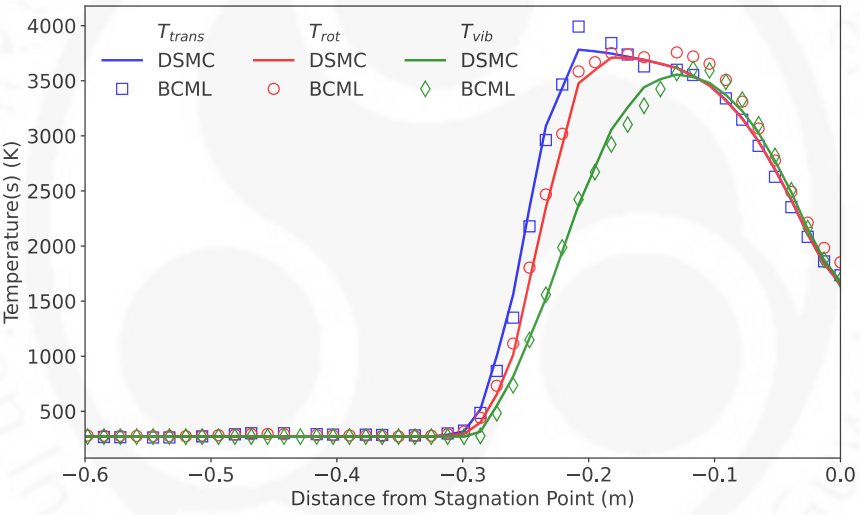

Figure 5.12: Comparison of various temperatures (translational, rotational, and vibrational) along the stagnation line, from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 9 flow over an arbitrarily shaped geometry at 67.5 km with $T_{wall} = 1000$ K.

Figure 5.13 shows the variation of mole fractions of various species along the stagnation line for Mach 32 flow over a circular cylinder geometry at atmospheric conditions corresponding to 75 km altitude. It is obvious that the flow consists of molecular nitrogen (N_2) and Oxygen (O_2) in the pre-shock region, with mole fractions close to their freestream values. In the shock and post-shock regions, dissociation of molecular species becomes significant due to the high temperatures, leading to increased mole fractions of atomic nitrogen (N) and Oxygen (O). This is accompanied by the formation of small amounts of nitric oxide (NO) species through the exchange reactions. It can be observed from Fig. 5.13 that an excellent agreement is observed for all species.

Table 5.3: Initial and boundary conditions for validation cases for testing the BCML model, along with error analysis.

Geometry	Pressure (Pa)	Temp (K)	AoA	Inlet Velocity (m/s)	P_{error}	$T_{\text{trans-error}}$	$U_{velocity\text{-error}}$
Circle	3.544	211.1	0°	2912.3	5.240%	2.961%	9.207%
Arbitrary	10.743	269.75	0°	2962.97	4.525%	1.703%	9.191%

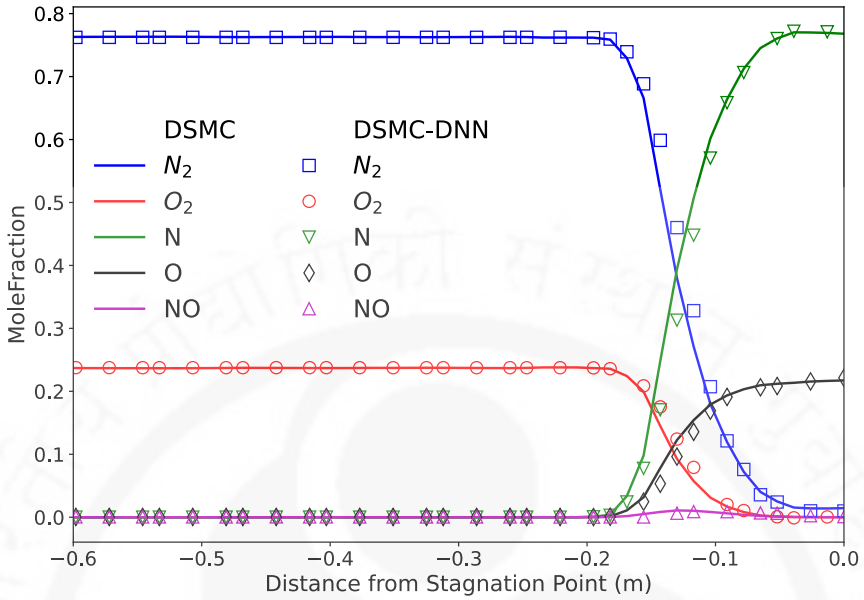

Figure 5.13: Comparison of mole fractions of various species along the stagnation line from the Mach 25-35 ChemDNN model and the DSMC simulation for Mach 32 flow over a circular cylinder geometry at 75 km with $T_{wall} = 1000$ K.

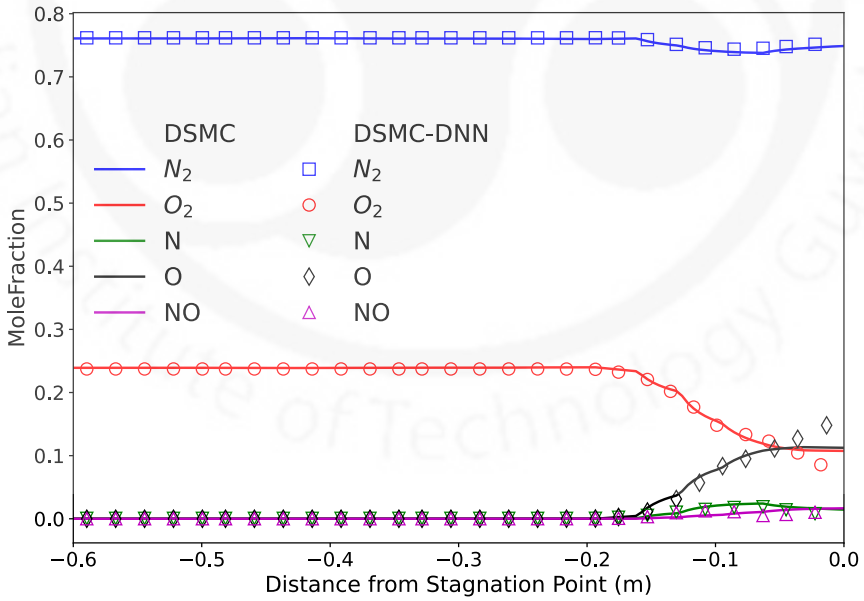

Figure 5.14: Comparison of mole fractions of various species along the stagnation line, from the Mach 15-25 ChemDNN model and the DSMC simulation for Mach 22 flow over a square-shaped geometry at AoA 36° at 80 km with $T_{wall} = 1000$ K.

Similarly, Fig. 5.14 presents the variation of different reacting species along the stagnation line for a Mach 22 flow over square-shaped geometry at an angle of attack of 36°, under atmospheric conditions corresponding to 80 km. Again, excellent agreement is obtained

for all species, and the corresponding L_2 norm errors are listed in Table 5.4. Overall, this validation demonstrates that the ChemDNN model can accurately predict mole fractions within the domain.

Table 5.4: Initial and boundary conditions for validation cases for testing the ChemDNN model, along with L_2 error analysis.

Geometry	P_∞ (Pa)	T_∞ (K)	AoA	U_∞ (m/s)	Error (N_2)	Error (O_2)	Error (N)	Error (O)
Circle	3.544	211.1	0°	9319.64	0.49%	0.60%	0.963%	0.589%
Square	1.417	190.5	36°	6086.586	0.346%	3.877%	7.652%	2.470%

5.5 Complete Hypersonic Flow Analysis for re-entry Vehicles

After independently validating the individual models, a comprehensive prediction of all thermophysical and species-specific flow fields was performed for an arbitrarily shaped geometry and compared with DSMC results. For this purpose, an arbitrarily shaped bluff body consisting of three straight edges and an arc was considered. It should be noted that the geometry and test condition of Mach 27 at an altitude of 82 km were not included in any of the training datasets corresponding to the BCML and ChemDNN models. Mach 25-35 BCML and ChemDNN models are used to predict the flow fields.

Figure 5.15 shows a comparison of contour plots for translational, rotational, and vibrational temperatures, as well as pressure and velocity, generated using the BCML model with those obtained from the DSMC simulations. The BCML-predicted contours exhibit reasonable agreement with the DSMC results in capturing the overall flow structure, including the shock stand-off distance and post-shock relaxation regions. There are deviations in the wake region away from the geometry. Such deviations were also observed in the flow field prediction from the low-altitude NSF-based BCML model [105]. This difference can primarily be attributed to the approximation of the influence of the boundary conditions using only 16 generators. It is expected that accuracy will improve as more generators are added. However, this would require additional training data and computational effort. As the objective of the present work is to develop a BCML model that provides a rough estimate of the flow field around a bluff body under hypersonic conditions, the number of generators has been limited to 16.

Figures 5.16 and 5.17 show the comparison of thermophysical flow field properties along the stagnation line from the S_{16} Mach 25-35 BCML model and the DSMC simulation. The BCML flow field predictions exhibit acceptable agreement with the DSMC results for both pressure and u-velocity, accurately capturing the sharp gradients across the shock layer and

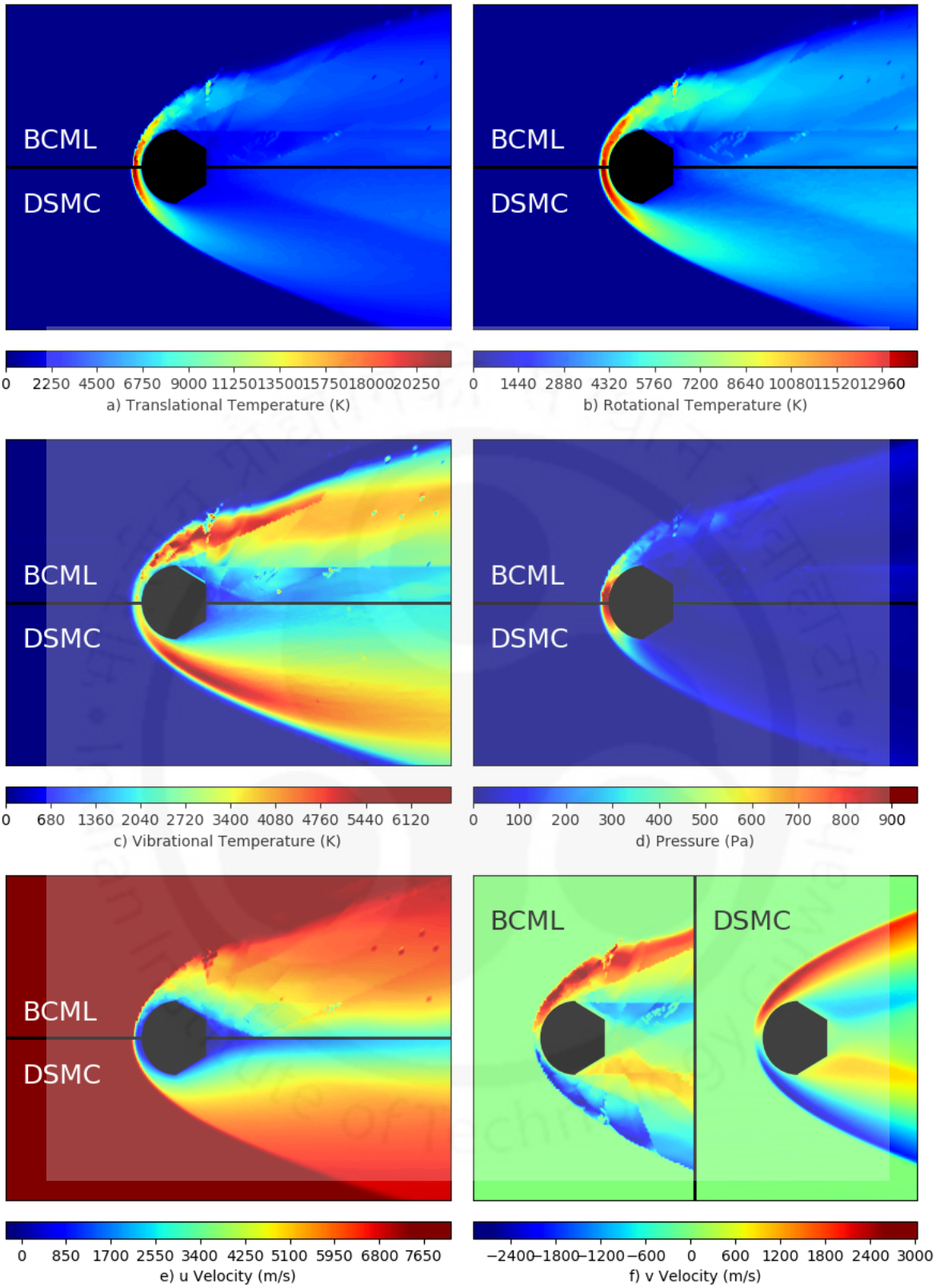

Figure 5.15: Comparison of flow field contour plots of (a) translational temperature, (b) rotational temperature, (c) vibrational temperature, (d) pressure, (e) u-velocity, and (f) v-velocity, from the S_{16} Mach 25-35 BCML model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with T_{wall} =1000 K.

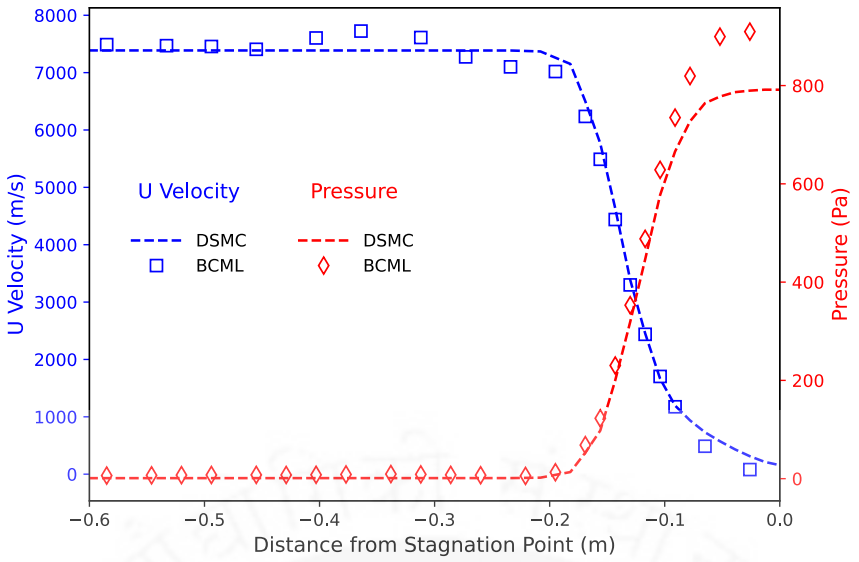

Figure 5.16: Comparison of pressure and u-velocity along the stagnation line from the S_{16} Mach 25-35 BCML model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with $T_{wall} = 1000$ K.

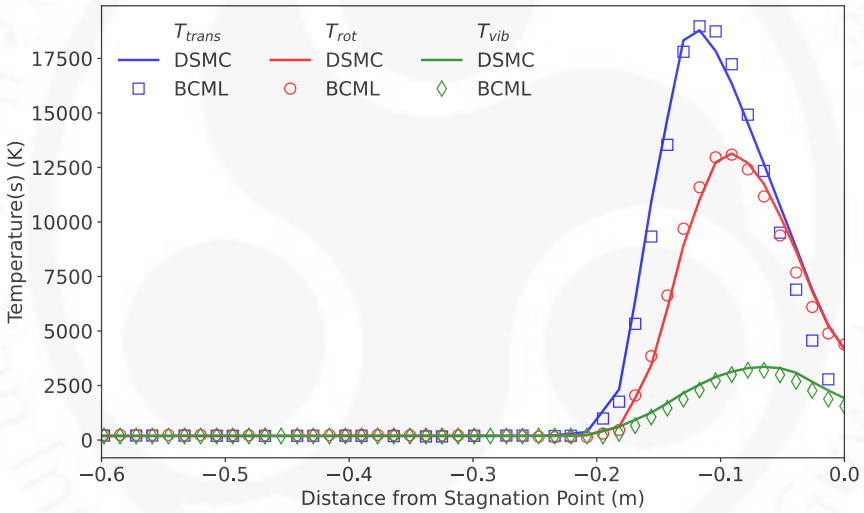

Figure 5.17: Comparison of various temperatures (translational, rotational, and vibrational) along the stagnation line from the S_{16} Mach 25-35 BCML model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with $T_{wall} = 1000$ K.

the overall post-shock behaviour. This consistency demonstrates the BCML approach's ability to reproduce key flow features in the non-equilibrium post-shock regime. However, it must be noted that the stagnation pressure predicted by the BCML model is less accurate than the other properties. Additionally, Fig. 5.17 shows that the translational, rotational, and vibrational temperatures exhibit non-equilibrium behaviour across the shock, with the translational temperature peaking first, followed by an equilibration of the rotational and vibrational modes in a decreasing manner in the downstream region of the shock structure. Again, the three temperature variations predicted by the BCML model agree well with the DSMC results. The ML model accurately captures both the shock structure and the relaxation trends of internal energy modes.

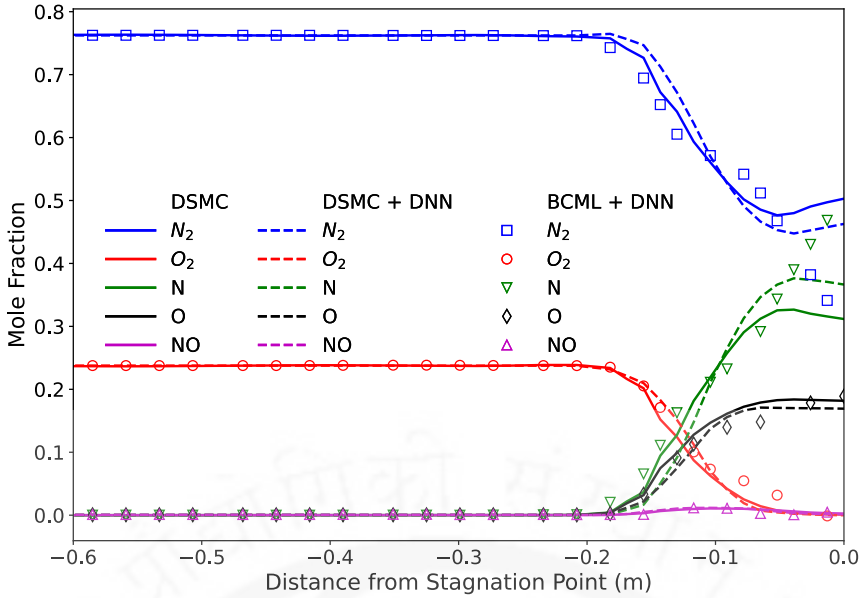

Figure 5.18: Comparison of mole fractions of various species along the stagnation line from the Mach 25-35 ChemDNN model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with $T_{wall} = 1000$ K.

Figure 5.18 shows the variation of mole fractions of various species along the stagnation line obtained from the DSMC simulations and the ChemDNN model. Additionally, Fig. 5.18 shows the results from the Mach 25-35 ChemDNN model, which utilises both the DSMC flow field data (labelled as DSMC+DNN, represented by dashed lines in the plot) and those predicted by the S_{16} Mach 25-35 BCML model (labelled as BCML+DNN, represented by symbols in the plot). Since the flow field data predicted by the BCML model is reasonably in agreement with the DSMC flow field, the difference between the results from the two ChemDNN solutions, as well as the difference with the mole fraction obtained directly from the DSMC simulation, is within satisfactory agreement.The lower accuracy for the N_2 and N mole fractions is primarily attributed to the errors in the pressure field predicted by the BCML model near the stagnation region. Since the dissociation of nitrogen species is highly sensitive to the local thermodynamic state, particularly pressure and temperature, these deviations are propagated through the ChemDNN model, resulting in relatively larger errors in the predicted N_2 and N mole fractions. Nevertheless, the overall agreement between the ChemDNN predictions and the DSMC results remains satisfactory for all species. In particular, the predictions for the mole fractions of molecular and atomic oxygen agree very closely. The L_2 norm errors for the ChemDNN approach, calculated with respect to the DSMC simulation, were found to be 2.2732%, 0.256%, 1.08037%, and 0.3408% for the mole fractions of species N_2, O_2, N, and O, respectively. In addition to this, the mole fractions of various species predicted by the Mach 25-35 ChemDNN model using the DSMC-generated flow field data as input are in commendable agreement with those obtained from the DSMC simulation, as shown in Figure 5.19. Overall, the contour and line plots indicate that the

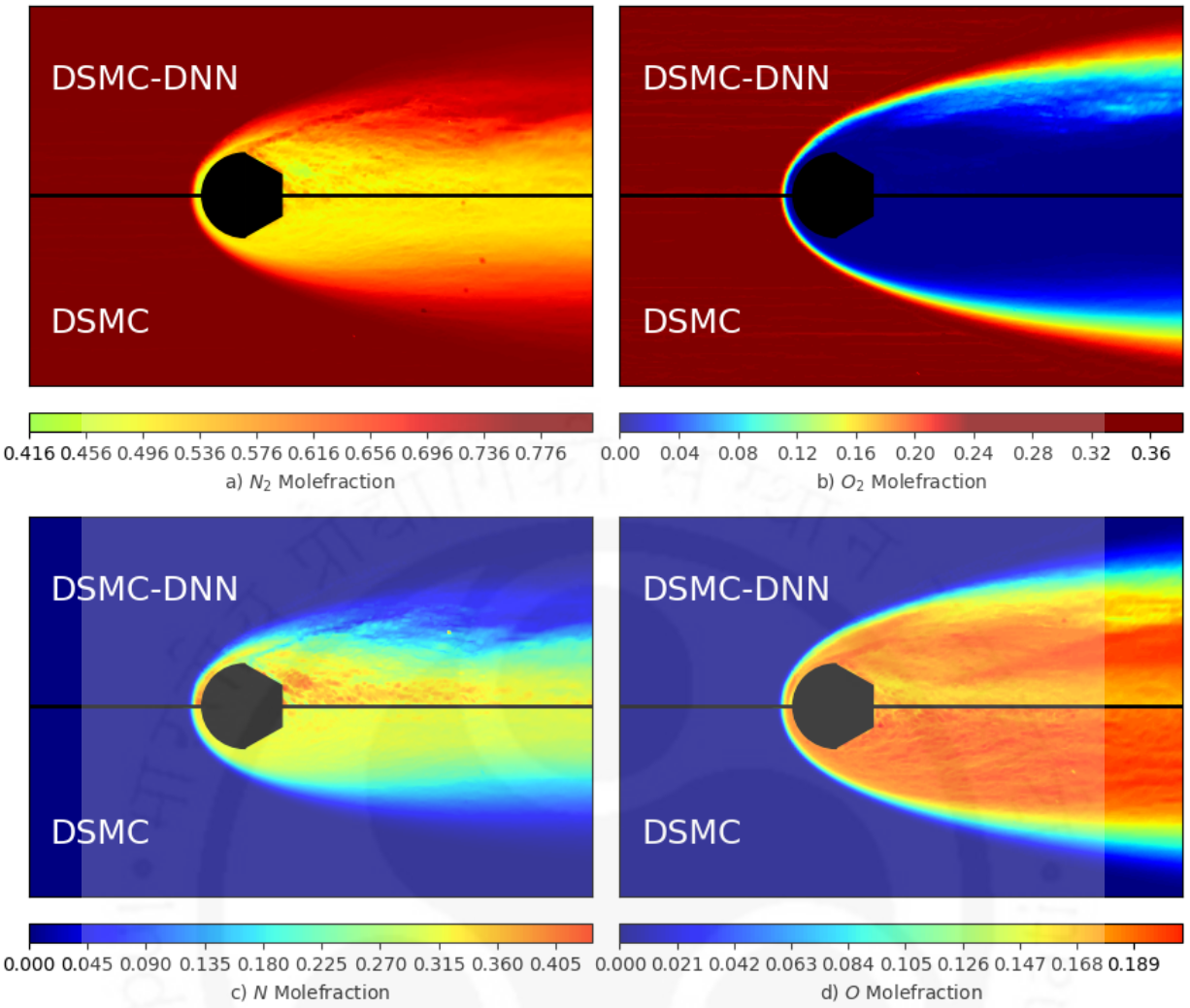

Figure 5.19: Comparison of mole fractions contour plots of (a) N_2, (b) O_2, (c) N, and (d) O predicted by Mach 25-35 ChemDNN model and the DSMC simulation for Mach 27 flow over a circular cylinder at 82 km with T_{wall} =1000 K.

combination of the S_{16} BCML model and the ChemDNN model predicts both macroscopic flow properties and thermochemical properties with a decent degree of accuracy.

Estimation of the peak heat flux is essential for selecting suitable materials for the TPS design. For a comprehensive analysis of re-entry flows at hypersonic conditions, a separate DNN model, the SurfPropDNN model (Fig. 5.20), is also reported and can be used to estimate surface properties of the re-entry vehicle. The input layer comprises four inputs: the distance of the point on the surface from the centre of the geometry, the angle from the stagnation point, the freestream Mach number, and the Knudsen number, calculated from ambient conditions. The output layer consists of the friction coefficient (C_f), heat flux coefficient (C_h), and pressure coefficient (C_p). These quantities are calculated directly from the DSMC simulations and, therefore, do not require the calculation of gradients of flow properties.

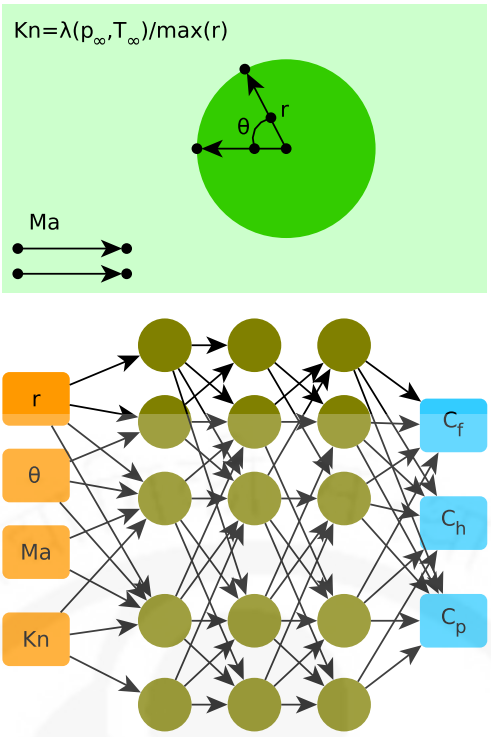

Figure 5.20: DNN architecture of the SurfPropDNN model for estimating surface properties on the re-entry geometry.

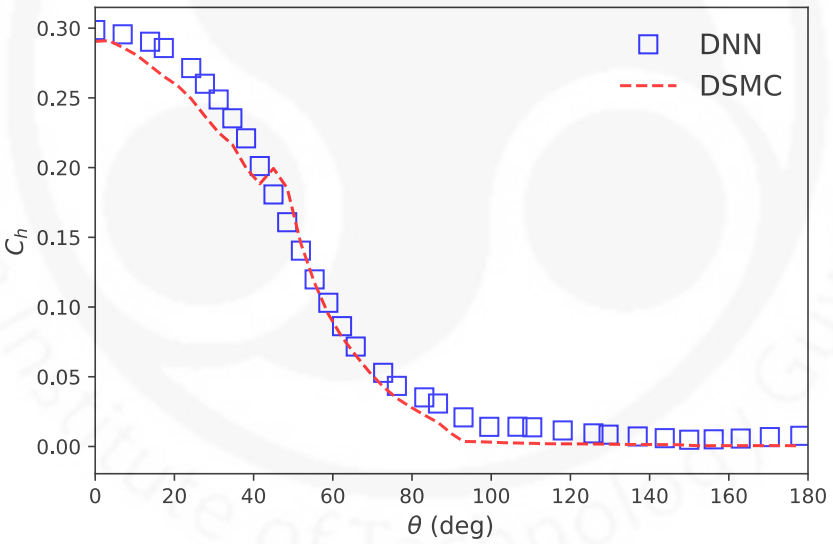

Figure 5.21: Comparison of coefficient of heat flux along the re-entry vehicle surface, from the Mach 25-35 SurfPropDNN model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with $T_{wall} = 1000$ K.

Figures 5.21 and 5.22 show the comparison of the coefficient of heat flux and the coefficient of pressure from the SurfPropDNN and the DSMC simulations for the Mach 27 flow over the arbitrarily shaped geometry. The peak value of C_h from DSMC is 0.29140, whereas the SurfPropDNN model predicts 0.29968, corresponding to an error of 2.968%. Similarly, the DSMC peak value of C_p is 1.81576, while the DNN prediction is 1.72313, yielding an error of 5.101%. It should be noted that the SurfPropDNN model can accurately predict the

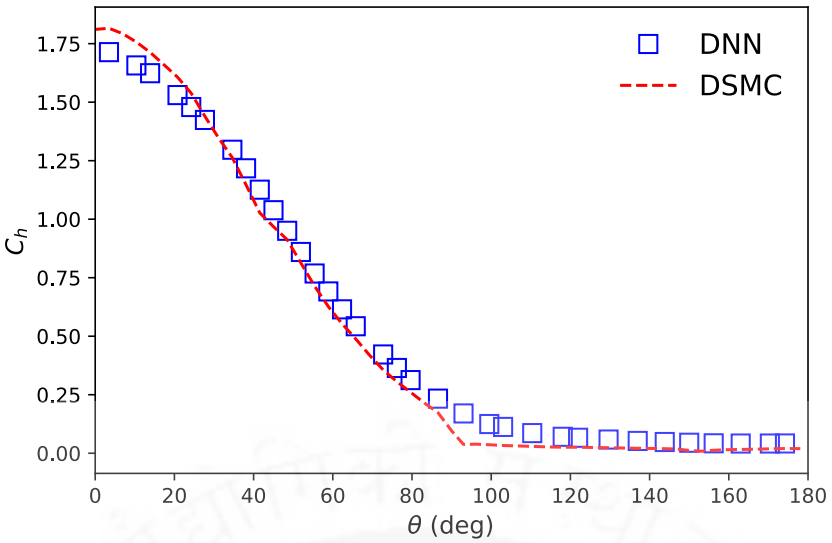

Figure 5.22: Comparison of coefficient of pressure along the re-entry vehicle surface, from the Mach 25-35 SurfPropDNN model and the DSMC simulation for Mach 27 flow over an arbitrarily shaped geometry at 82 km with $T_{wall} = 1000$ K.

overall trend and, in particular, the properties at the stagnation point. However, the model still suffers from inconsistencies at certain points, particularly at the surface's sharp corners. This will be improved in the future by incorporating curvature effects in arbitrarily shaped geometries.

It should be noted that mole fraction and surface properties are more sensitive to the local flow-field accuracy compared to the bulk flow quantities such as pressure and velocity. Small deviations in the predicted heat-flux coefficient may influence the estimated surface heat-transfer rate, thereby affecting thermal protection system (TPS) design margins. Similarly, an error in the pressure coefficient distribution may affect the estimation of aerodynamic forces and drag characteristics. In reacting hypersonic flows, errors in temperatures can also affect dissociation rates and the mole fractions of the various species, particularly in strong-shock and thermochemical nonequilibrium regions. Consequently, the present ChemDNN and SurfPropDNN frameworks should primarily be interpreted as rapid, preliminary-estimation tools within the trained operating regime, rather than as standalone replacements for detailed, high-fidelity reacting-flow simulations.

5.6 Generalisation of the BCML Framework to Untrained Ambient Conditions

In practical hypersonic applications, surface thermal conditions often vary significantly due to aerodynamic heating, material properties, and mission-specific constraints. Consequently, models should be robust for the wall-temperature regimes included in the training. The

high-altitude DSMC-based BCML model was trained using flow-field data corresponding to a fixed prescribed wall temperature of 1000 K and then tested at other wall temperatures: 500 K and 1500 K. The resulting predictions are compared with the DSMC flow fields to examine the model's ability to capture shock structure and flow physics. This analysis provides critical insight into the generalisation strength and physical consistency of the BCML approach for realistic hypersonic flow scenarios.

The first test case considers a Mach 10 flow at ambient conditions corresponding to an altitude of 75 km. The bluff body shape chosen for the first test was a simple circular cylinder with a constant surface temperature of 500 K. Figures 5.23 and 5.24 show the centreline line plots for the three temperatures, and other flow properties (pressure and velocity), respectively. The peak pressure predicted by the BCML model is 458.342 Pa, which is 2.082% higher than the DSMC simulation (449.02 Pa). Similarly, the peak Translational Temperature predicted by the BCML model is 3711.11 K, which is 1.582% lower than the DSMC simulation (3770.77 K). The corresponding mole fractions predicted by the ChemDNN model are shown in Figure 5.25. The model could predict nitrogen species relatively well with some minor deviations in oxygen species.

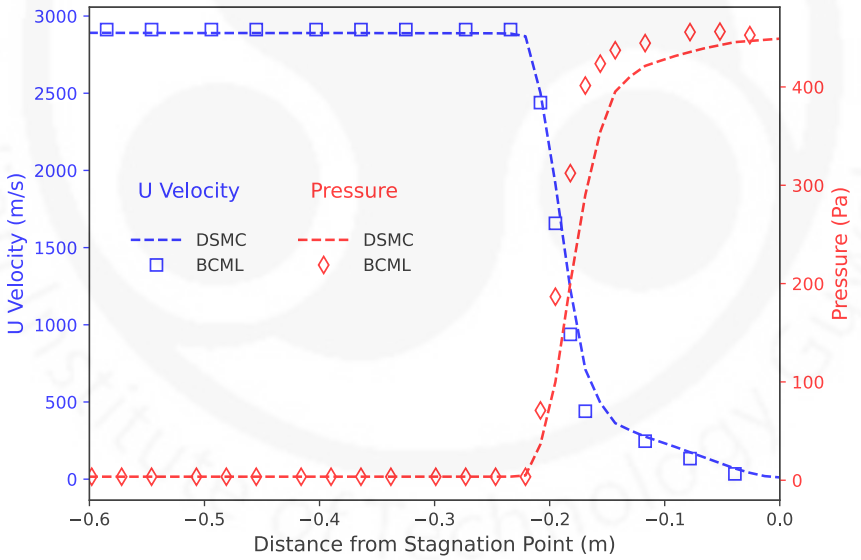

Figure 5.23: Comparison of pressure and u-velocity along the stagnation line, from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 500$ K.

A similar analysis has been performed for a constant wall temperature of 1500 K and on the same Mach 10 flow over a circular cylinder at ambient conditions, corresponding to an altitude of 75 km. The centreline line plots are shown in Figures 5.26 and 5.27, respectively. The peak pressure predicted by the BCML model is 429.681 Pa, which is 5.10% lower than the DSMC simulation (452.8 Pa). Similarly, the peak translational temperature predicted by

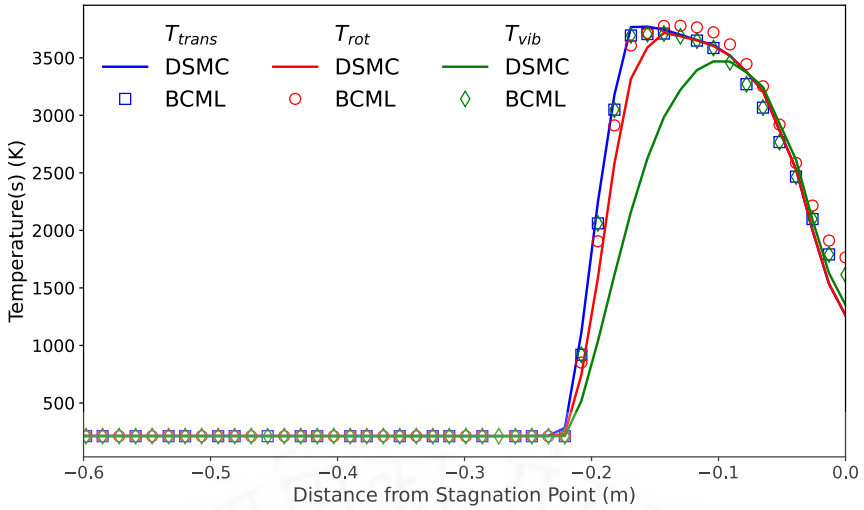

Figure 5.24: Comparison of various temperatures (translational, rotational, and vibrational) along the stagnation line, from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 500$ K.

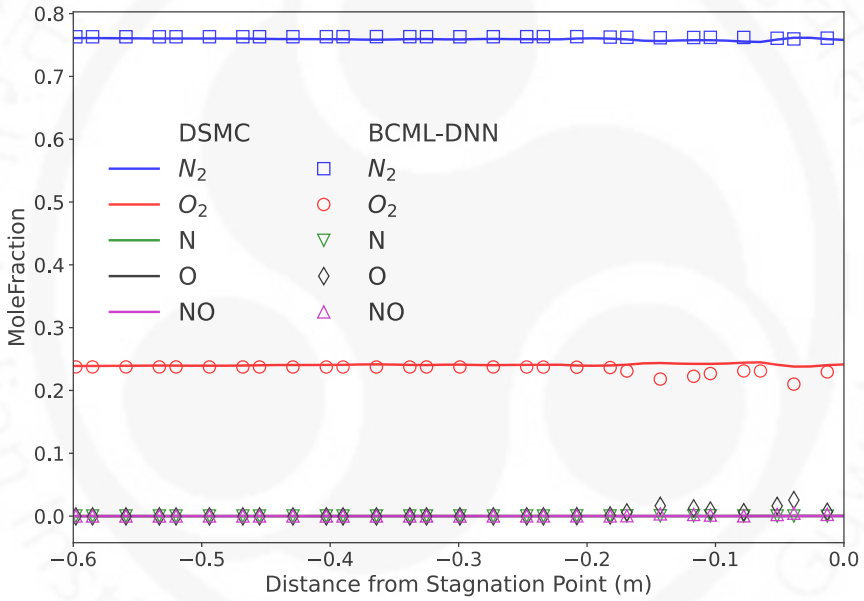

Figure 5.25: Comparison of mole fractions of various species along the stagnation line, from the Mach 8-15 ChemDNN model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 500$ K.

the BCML model is 3951.51 K, which is 4.05% higher than the DSMC simulation (3797.58 K).

In addition to the above cases, a wide range of DSMC simulations was conducted at ambient conditions, beyond the training data for the BCML/ChemDNN framework. The flow fields predicted by one of the three BCML models were compared with those from DSMC simulations at 90 km and 55 km, each at Mach 5, 20, and 40. It was found that the BCML model's accuracy deteriorates significantly outside the range of training conditions. This is mainly due to the high variation in the ambient number densities at altitudes above 84 km and below 60 km. Another set of simulations was run at 75 km at Mach 5 and Mach

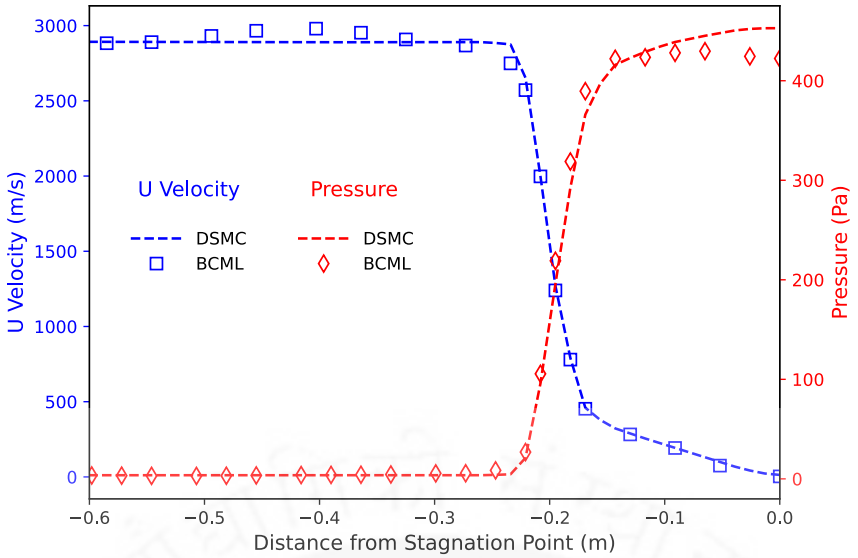

Figure 5.26: Comparison of pressure and u-velocity along the stagnation line, from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 1500$ K.

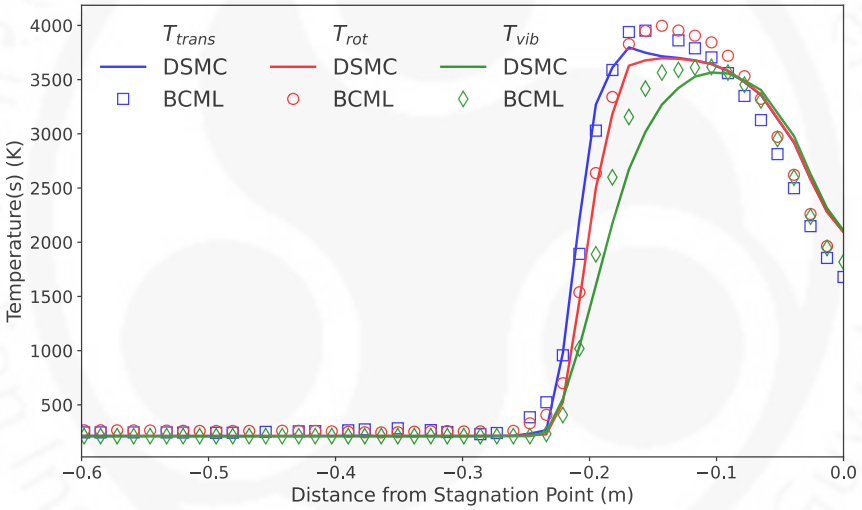

Figure 5.27: Comparison of various temperatures (translational, rotational, and vibrational) along the stagnation line, from the S_{16} Mach 8-15 BCML model and the DSMC simulation for Mach 10 flow over a circular cylinder at 75 km with $T_{wall} = 1500$ K.

40. It was found that the translational temperature, pressure, and velocity from the BCML model were in reasonable agreement with the DSMC results. However, the BCML model was unable to capture the degree of non-equilibrium. Figure 5.28 shows a pictorial representation of the various cases included in the present study: both within and outside the range of the training conditions. Overall, the model's extrapolation capability is limited, especially if conditions are far from the training range.

Overall, the ML-based models demonstrated a reasonable correlation in predicting surface and flow properties for hypersonic rarefied flows. The DNN-based models captured key flow

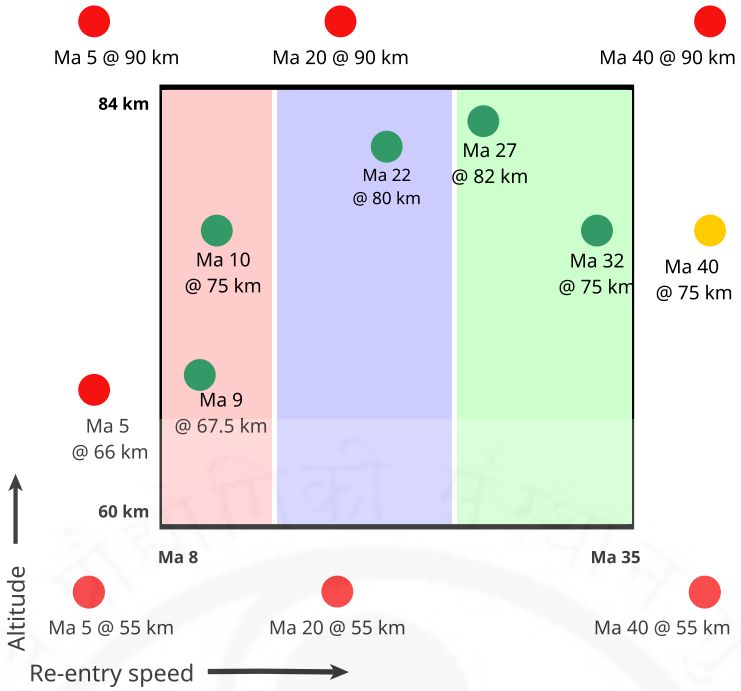

Figure 5.28: Graphical representation of various simulation cases depicting the altitude (in km) and Mach number (Ma) (Key: Green-Good agreement, Yellow: Reasonable agreement for limited variables, Red: highly inaccurate)

features and chemical non-equilibrium behaviour with reasonable accuracy compared to DSMC simulations. Although minor deviations were observed near the stagnation point and in the wake region, the overall trends and peak values of various thermophysical properties agreed well with the DSMC results. These results indicate that the proposed ML approach has the potential to serve as an efficient surrogate for DSMC-based simulations, offering a substantial reduction in computational cost while maintaining acceptable predictive accuracy for high-enthalpy flow conditions.

To provide a more systematic assessment of the generalisation capability of the proposed BCML framework, the validation cases have been classified into four categories, namely: (i) seen geometry/seen flow condition, (ii) unseen geometry/seen flow condition, (iii) seen geometry/unseen flow condition, and (iv) unseen geometry/unseen flow condition. For each category, quantitative validation has been performed by comparing the stagnation-line distributions of pressure (P), translational temperature (T_{trans}), and velocity magnitude (U) with the corresponding reference DSMC solutions. The errors are tabulated in Table 5.5, thereby providing a clearer distinction between the interpolation, geometry-generalisation, and unseen operating-condition capabilities of the proposed BCML framework.

The overall reliability and operational constraints of the proposed BCML framework can be characterised by several key factors:

- The BCML framework is relatively more accurate for simulation cases that are within

Table 5.5: Comparison of performance of the BCML model for various classes of geometries along the stagnation line.

Geometry	**Flow Condition**	**Test Case**	**Altitude (km)**	**Mach**	$T_{\text{trans,error}}$ (%)	P_{error} (%)	U_{error} (%)
Seen	Seen	Circular Cylinder	72	35	8.582	1.401	4.642
Unseen	Seen	Arbitrary Bluff Body	72	35	9.561	8.254	4.632
Seen	Unseen	Circular Cylinder	75	27.5	9.840	2.348	2.590
Unseen	Unseen	Arbitrary Bluff Body	75	27.5	9.791	7.394	6.697

the range of the training database. The extrapolation study presented in Figure 5.28 indicates that the model generalises better to variations in wall temperature, whereas comparatively larger deviations are observed for extrapolated free stream conditions because of significant deviation in the flow characteristics.

- Greater errors are observed in regions containing strong shock waves, free shear layers, and wake structures, where steep flow gradients and thermochemical non-equilibrium effects make the flow physics more challenging to approximate using a surrogate model.
- Although the generator-based BCML formulation is aimed to be geometry-agnostic, its capability improves when the training database adequately represents the geometric features of practical configurations. Consequently, representative training geometries are important for achieving reliable predictions on previously unseen configurations.
- The ChemDNN model utilises the BCML-predicted flow field variables as inputs. Therefore, any error in the predicted thermodynamic state may affect the accuracy of mole fraction concentrations predicted by the ChemDNN, particularly in regions exhibiting strong thermochemical non-equilibrium.

It must be noted that the proposed BCML framework is intended as a rapid surrogate model within its validated operating envelope. For extrapolative conditions, highly novel geometries, or safety-critical applications, the parent CFD/DSMC solver is recommended to verify the predictions and provide the highest level of solution fidelity.

5.7 Consolidated Simulation Summary and Error Validation

The validation of the proposed BCML frameworks was performed using a combination of contour-field comparisons, stagnation-line profiles, peak-property deviations, shock stand-off distance estimation, surface-property validation, and domain-wide error norms. Since different metrics capture different aspects of the flow physics, both global and local error measures were analysed throughout the thesis. In general, the BCML framework demonstrated reasonable accuracy within the trained and interpolative parameter space, while larger local

deviations were observed near bow-shock edges, wake regions, and strongly extrapolative operating conditions.

Table 5.6: Consolidated summary of validation cases and error analysis.

Case	**Regime/Model**	**Geometry**	**Validation Type**	**Domain-wide Error**	**Local/Peak Error**	**Remarks**
Mach 6, 57.5 km	Low-altitude NSF-BCML	Circular Cylinder	Interpolation	$L_2(T) = 1.288\%$	Peak T: 8.7%	Accurate shock stand-off prediction; minor far-field shock-profile deviation
Mach 4, 52.6 km	Low-altitude NSF-BCML	Square Cylinder	Interpolation	$L_2(T) = 1.001\%$	Peak $T < 5\%$	Local wake-region deviations observed
Mach 9, 67.5 km	High-altitude DSMC-BCML	Arbitrary Geometry	Interpolation	$L_2(T) < 2\%$	Peak T: 5.2%	Local errors near bow-shock edge and far wake
Mach 12, 69 km	High-altitude DSMC-BCML	Arbitrary Geometry	Interpolation	$L_2(T) = 1.691\%$ $L_2(P) = 5.929\%$	Peak T: 8.95% Peak P: 6.303%	Local discrepancies near shock edges and downstream wake
Mach 22, 80 km	ChemDNN	Square Geometry	Interpolation	Species L_2 error: N_2: 0.346% O_2: 3.877% N: 7.652% O: 2.470%	–	Reasonable agreement for reacting-species prediction
Mach 27, 82 km	SurfPropDNN	Arbitrary Geometry	Interpolation	C_h: 2.968%	–	Surface heat-flux and pressure coefficients predicted reasonably
Mach 5/20/40	High-altitude DSMC-BCML	Circular Cylinder	Extrapolation	Higher deviations observed	Non-equilibrium mismatch	Predictive accuracy deteriorates outside the trained altitude/Mach range

Although the domain-wide L_2 error norms were generally low for the validation cases considered, local deviations were observed in regions containing strong gradients, particularly near bow-shock edges, stagnation regions, and far-wake structures. These local errors become more pronounced for unseen geometries and operating conditions beyond the training data. Therefore, the reported global error norms should be interpreted together with local peak-error analysis and contour-based comparisons.

Most validation cases presented in Chapters 4 and 5 are of an interpolative nature, in which the operating conditions and geometries differ from those in the training cases but remain within the range of the trained parameter space. In contrast, the extrapolation cases discussed in Section 5.6 involve operating conditions outside the nominal training range, where a noticeable deterioration in predictive capability was observed, particularly for thermochemical nonequilibrium quantities. Table 5.6 shows a consolidated summary of all validation cases included in the present thesis.

In addition to flow-field validation, surface quantities, such as the heat-flux coefficient (C_h) and pressure coefficient (C_p), were also examined using the SurfPropDNN framework. Representative validation cases demonstrated reasonable agreement with DSMC-derived surface quantities.

Chapter 6

BCML-assisted CFD Framework

A rapid prediction of hypersonic flow fields across both low-altitude and high-altitude regimes has been shown to be attainable with reasonable accuracy through the BCML framework. However, it must be noted that the true contribution of the BCML framework is not just the rapid flow field prediction, but rather its role as an assisting tool to accelerate conventional NSF and DSMC simulation workflows. In particular, two such applications have been examined in detail: one focusing on accelerating FVM-based NSF solver, and the other targeting the speed-up of DSMC simulations using improved meshing algorithms, which are described in detail in this Chapter.

6.1 Accelerating Convergence for NSF Simulations

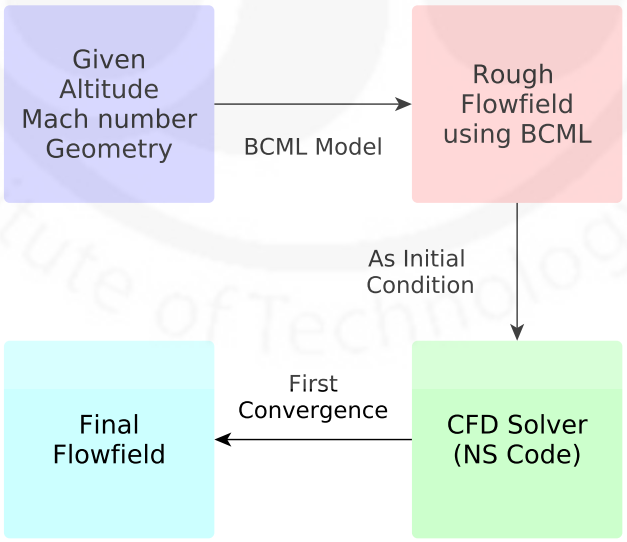

Figure 6.1: Flowchart of BCML-assisted CFD simulations

The primary motivation for developing the BCML model was to reconstruct the flow field around a bluff body in supersonic and hypersonic flows, providing a rough estimate of the bluff body's performance. In addition, the BCML model can be used in conjunction with a CFD solver, serving as a tool to improve efficiency and reduce computational cost. The flow

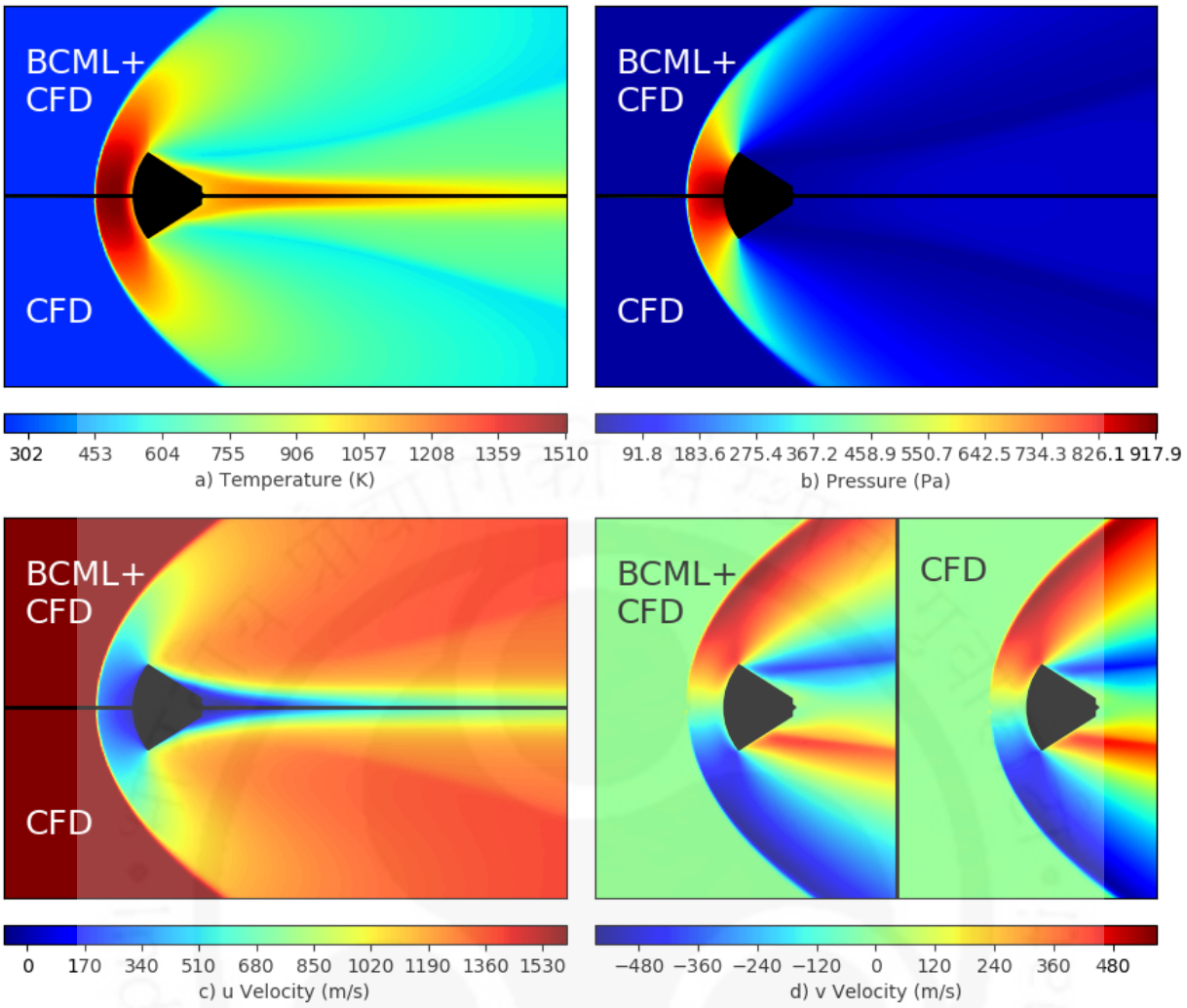

Figure 6.2: Comparison of flow field contour plots (a) temperature, b) pressure, c) u-velocity, and d) v-velocity predicted by the S_{16} BCML-assisted FVM solver and the FVM solver for Mach 5 flow over FIRE-II re-entry vehicle at 59.26 km with $T_{wall} = 1000$ K.

field generated by the BCML model can serve as an initial condition for the FVM solver. The FVM solver then continues to solve the NSF equations in the domain till convergence.

It was found that simulations with the BCML-assisted FVM solver converged faster than those with the traditional FVM solver, in which the internal cells are initialised uniformly in the domain to the inlet conditions. Figure 6.1 summarises the overall process of using the BCML model for improving the convergence rates for the CFD simulations.

Hypersonic CFD simulations often suffer from CFL limitations during the initial stages due to the strong nonlinearity of the governing equations. This nonlinearity is mainly due to steep pressure and temperature gradients in the strong bow shocks. In addition, high-temperature effects such as thermochemical non-equilibrium vibrational excitation and possible dissociation make the simulation more complex. At the start of the simulation, large residuals and abrupt adjustments in the flow variables can trigger Gibbs oscillations,

which eventually may lead to divergence. Using BCML-estimated flow fields instead of the freestream condition as initial conditions can improve the simulation's convergence.

Table 6.1: Computational cost for various computational methodologies (BCML-assisted FVM solver and traditional FVM solver).

	FVM Solver			BCML-assisted FVM Solver		
Geometry	Iterations	Runtime (CPU hr)	CPU time/Iteration	Iterations	Runtime (CPU hr)	CPU time/Iteration
Cylinder	36,931	22.72	2.215 sec	28,910	17.78	2.214 sec
Square	34,776	19.09	1.976 sec	26,326	14.87	2.033 sec
FIRE II	63,588	41.61	2.356 sec	49,515	31.57	2.295 sec

Table 6.1 compares the number of iterations required for convergence (mass residual $< 10^{-6}$) using the conventional FVM solver and the BCML-assisted FVM solver. All simulations were executed in parallel on a 24-core Intel Xeon workstation using MPI with uniform domain decomposition. The table also lists the corresponding CPU-hour costs, showing a consistent 22–25% reduction when the BCML-assisted solver is used. Figure 6.2 presents the contour plots of key flow properties from both solvers [(BCML-assisted FVM solver (shown as BCML+CFD) and FVM solver (shown as CFD)], which match closely since the same governing equations are solved to convergence in each case.

In addition to reducing computational cost, the BCML model can also generate an optimised mesh based on the flow field. The grid resolution can be increased in the shock region to better capture the shock. Also, the mesh resolution can be reduced in the pre-shock region, significantly reducing the simulation's computational cost.

As discussed in previous chapters, the BCML framework consists of a DNN model, followed by post-processing steps. To distinguish the contributions of each component of the proposed BCML framework, an additional study was conducted using this FIRE II validation case. The overall procedure is decomposed into four stages: (i) flow field predicted by the BCML DNN (Raw BCML), (ii) applying the shock-profile fitting and upstream consistency corrections (BCML + SC), (iii) flow field after a few FVM relaxation steps (BCML + SC + Rel), and (iv) the final BCML-assisted CFD solution (BCML + CFD). Figure 6.3 shows a comparison of the contour plot for translational temperature for each stage, along with the reference CFD solution. The results demonstrate that the post-processing procedures improve the physical consistency of the predicted flow field.

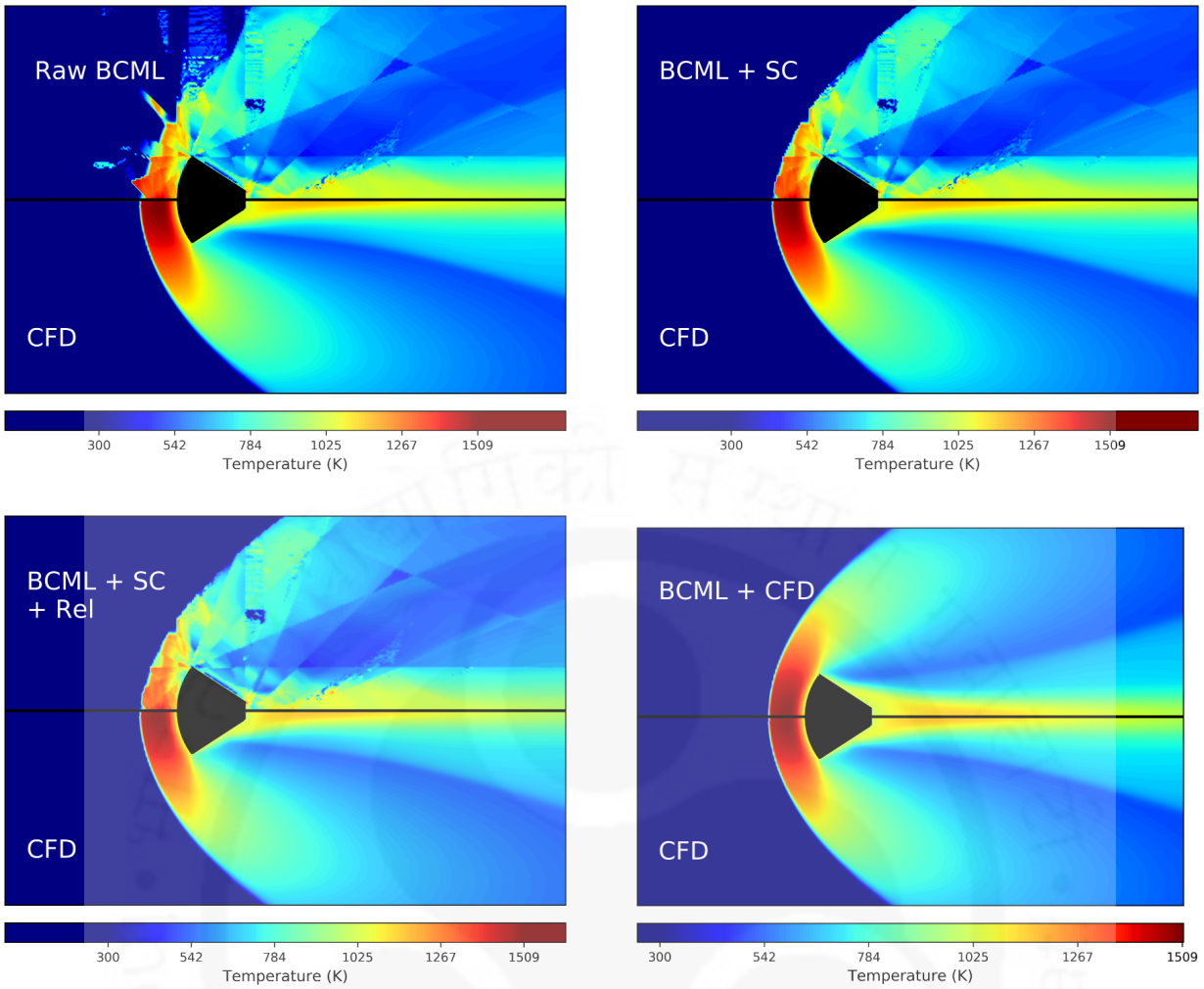

Figure 6.3: Comparison of the translational temperature contour plots predicted by the raw BCML prediction, BCML with shock correction (BCML + SC), BCML with shock correction and FVM relaxation (BCML + SC + Rel), and the final BCML-assisted CFD solution for Mach 5 flow over FIRE-II re-entry vehicle at 59.26 km with $T_{wall} = 1000$ K.

6.2 Optimised Mesh Generation for DSMC Simulations

6.2.1 Algorithm

In physical problems where flow features such as shocks, wakes, boundary layers, or concentration gradients are spatially localised, a highly resolved uniform mesh incurs a high computational cost, while inadequate grid resolution in regions of interest can lead to poor accuracy. Additionally, the quality of the computational mesh is closely coupled to the numerical stability of continuum-based solvers in hypersonic flow simulations. It is mentioned in Chapter 3.3 that many ML studies focus on generating optimised meshes for traditional CFD methods, such as finite-volume and finite-element methods. These models employ geometric parameters such as mesh quality.

Accurately capturing transport phenomena in the DSMC algorithm also critically depends on the computational mesh resolution. However, unlike traditional CFD algorithms,

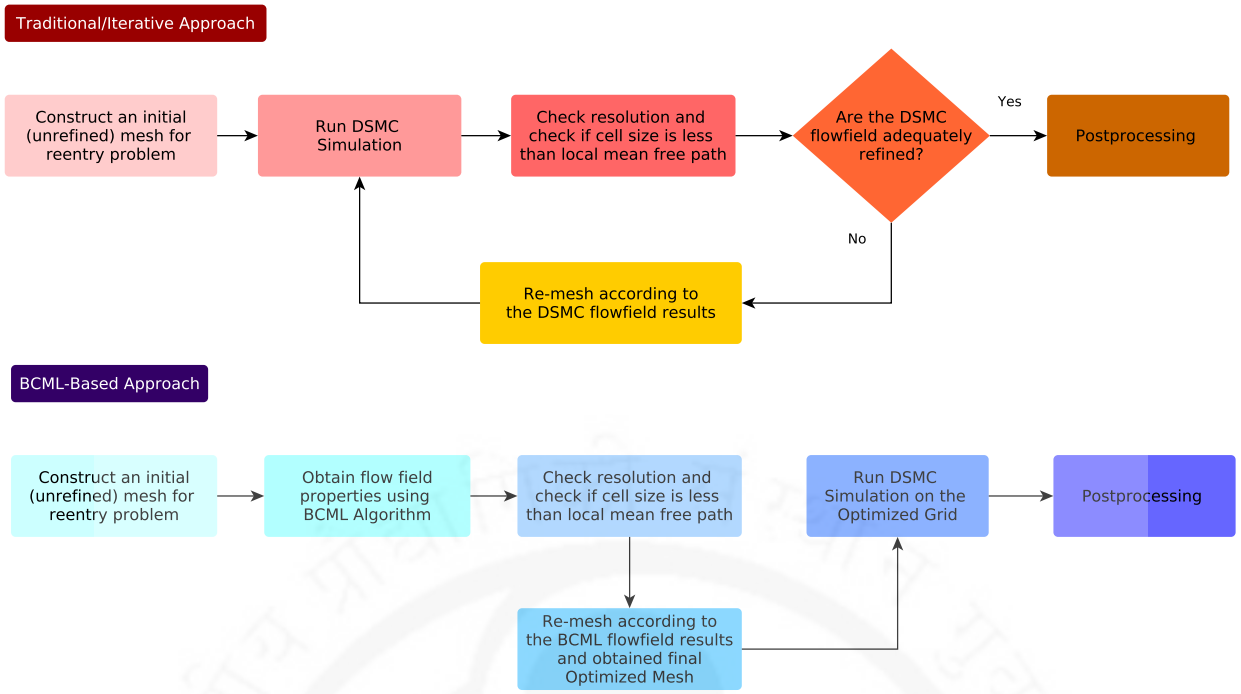

Figure 6.4: Flowchart highlighting the contrast between traditional and BCML-based Meshing Design philosophy.

DSMC requires additional constraints on the mesh generation, based on flow physics. For DSMC, mesh quality influences accuracy and computational efficiency rather than numerical stability. An important criterion for mesh generation is that the cell sizes should be a fraction of the local mean free path. This is unique to the DSMC algorithm; hence, meshes generated by engineers using traditional CFD codes or commercial software are often not suitable for DSMC simulations. A new data-driven mesh generation algorithm is designed to improve the computational efficiency of DSMC simulations, extending the BCML framework. The mesh generation algorithm is based on flow physics predicted by the machine learning model a priori, thereby avoiding iterative design of meshes. Figure 6.4 illustrates the overall BCML-based design methodology in comparison with the traditional mesh design approach.

The methodology adopted for the development of the present BCML model closely follows that described in Chapter 5 for the high-altitude DSMC-based BCML framework. Accordingly, the data generation procedure, network architecture, training strategy, and validation methodology remain essentially unchanged. The only difference is that the present model is trained using DSMC flow fields obtained for frozen conditions and includes the local mean free path as an additional output parameter. This additional output enables the BCML predictions to be directly utilised for adaptive DSMC mesh generation, which forms the focus of the present chapter.

6.2.2 Data Generation and Training ML Model

The high-altitude DSMC-based BCML framework can accelerate direct simulation Monte Carlo (DSMC) computations by generating optimised meshes via Delaunay triangulation and adaptive mesh refinement based on fast flow-field predictions. Towards this, a new high-altitude DSMC-based BCML model is trained under frozen rarefied hypersonic conditions at altitudes ranging from 60 to 84 km. The new BCML model is used to predict the local mean free path in the flow domain, along with other macroscopic properties around bluff bodies.

The training data for the new high-altitude BCML model for frozen conditions is generated using the in-house DSMC solver over a wide range of freestream conditions and Mach numbers. The computational domain considered in the present study extends 13 m in the streamwise direction and 9 m in the transverse direction. The wall temperature is held constant at 1000 K for all cases. One set of DSMC simulations was performed for a square-cylinder bluff body. The atmospheric conditions include those at altitudes between 60 and 84 km (60, 66, 72, 78, and 84 km) and freestream Mach numbers range from 8 to 35 (8, 15, 25, and 35). The ambient conditions employed in the simulations are the same as those used for the chemically reactive, realistic DSMC simulations. Four angles of attack (0°, 15°, 30°, and 45°) are considered for the square cylinder, resulting in a total of 80 cases. In addition to the square-cylinder, flow fields were obtained for circular-cylinder and arbitrarily shaped re-entry geometries constructed from random combinations of circular arcs and straight-line segments. The arbitrarily shaped geometries are constructed in such a manner that they are inscribed in a unit-radius circle or circular cylinders and arbitrary geometries, the freestream conditions are randomly sampled within the same altitude and Mach number limits to provide broader coverage of the relevant flow regime. A total of 36 simulations are conducted for the circular cylinder, and 38 simulations involving randomly generated bluff-body shapes composed of straight-line segments and circular arcs are included to increase the dataset's diversity. In total, 154 DSMC simulations are performed across the three geometry categories. These simulations yield approximately 18.724 million cell-based data samples and require a total computational cost of about 0.17 million CPU hours on an Intel Xeon workstation.

A mixture of nitrogen and oxygen is the working gas for the simulations. Elastic collisions in the DSMC simulations were handled using the standard VHS model. The reference diameter, reference temperature, and temperature exponent (ω) parameters needed for the VHS model are obtained from Bird [14]. Inelastic energy-exchange processes were modelled using a constant rotational $Z_{\text{rot}} = 5$ and vibrational $Z_{\text{vib}} = 50$ relaxation parameter-based Larsen-Borgnakke model. A time step of 4×10^{-7} s is used, with approximately seven simulated particles per cell. Overall, the adopted physical models provide a physically

reasonable and computationally efficient representation of non-equilibrium effects in high-enthalpy re-entry flows.

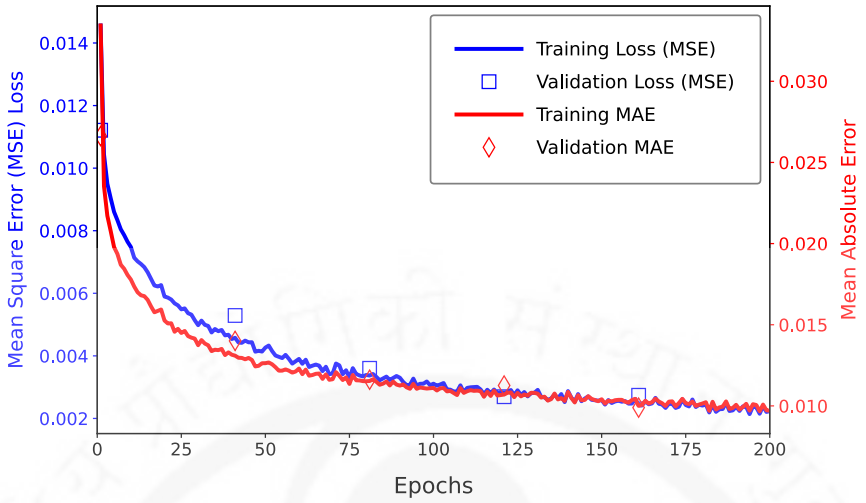

Figure 6.5: Evolution of training and validation loss with epochs for the S_{16} BCML model.

The structure of the new BCML model for the frozen condition is similar to that of the high-altitude DSMC-based BCML model discussed in Chapter 5. For each generator, the distance function and the four boundary conditions form the set of input parameters. In the present case, a 16-generator model S_{16} is developed, similar to the previous BCML models, yielding a total of 80 nodes in the input layer. These inputs capture the influence of surrounding boundary conditions on the local state of the cell centre and are passed through multiple fully connected hidden layers to capture the nonlinear coupling inherent in rarefied, non-equilibrium flows. The output layer comprises seven macroscopic flow quantities, namely the pressure, translational, rotational, and vibrational temperatures, the two Cartesian velocity components (u and v), and the local mean free path. The additional inclusion of the local mean free path as an output is a distinguishing feature of the new model. Figure 6.5 illustrates the convergence behaviour of the S_{16} BCML model, where the mean squared error (MSE) and the mean absolute error (MAE) for the training and validation datasets decrease with increasing epochs and asymptotically approach stable values.

6.2.3 Flow Field Prediction using BCML Model

After training the BCML model, it was validated for a wide range of ambient conditions and re-entry geometries under frozen conditions. Two of the results have been discussed here. The point of the validation is to demonstrate that the BCML model can generate sufficiently accurate flow-field results at a fraction of the computational cost, which can then be used to generate optimised meshes for the same conditions using the DSMC method.

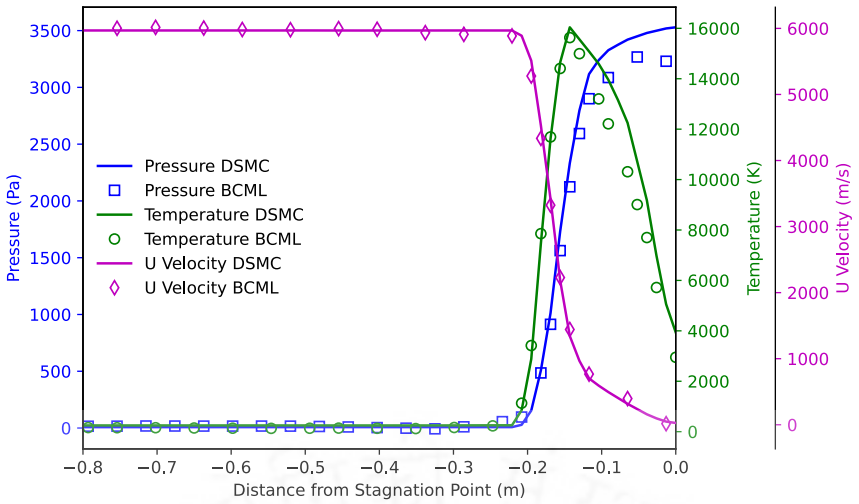

Figure 6.6: Variation of flow properties along the stagnation line for Mach 19 flow over a circular cylinder at 70 km with $T_{wall} = 1000$ K

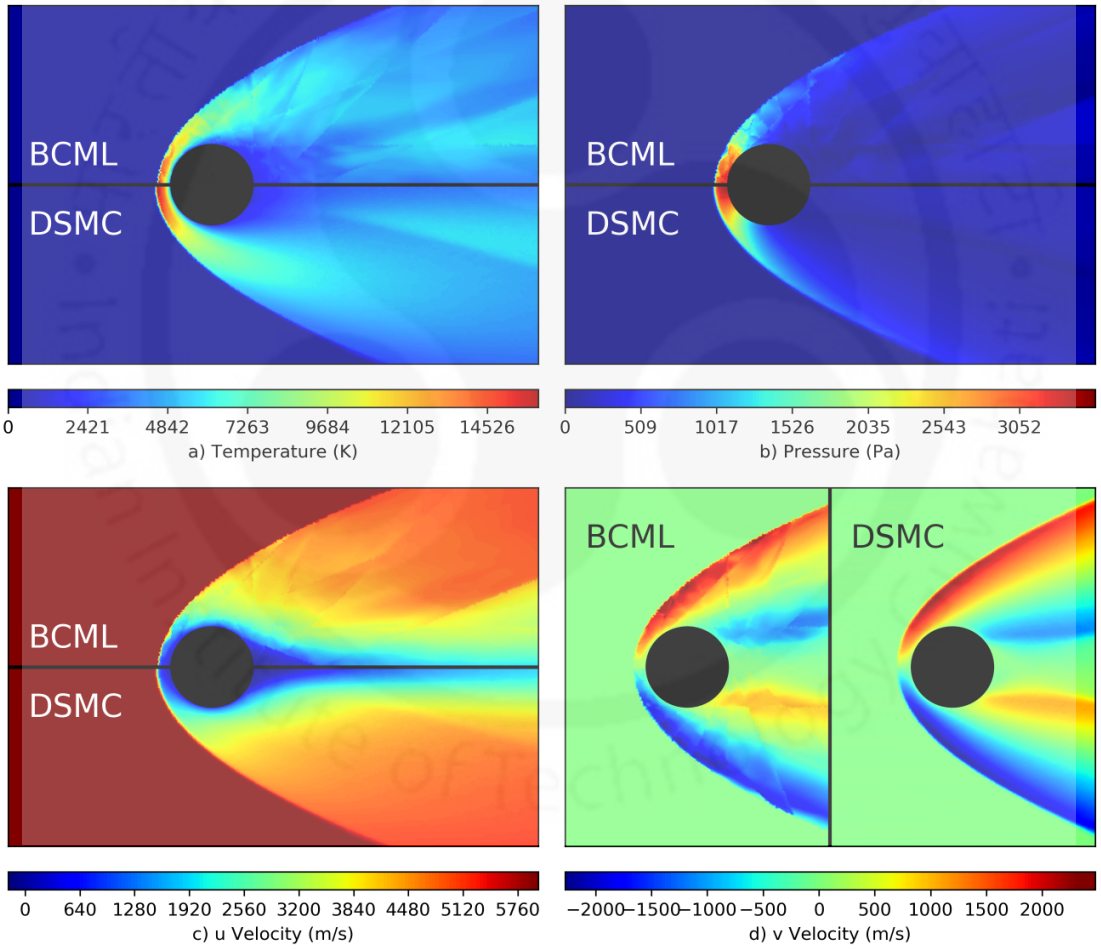

Figure 6.7: Comparison of flow field contour plots of (a) translational temperature, (b) pressure, (c) u-velocity, and (d) v-velocity from the S_{16} BCML model and the DSMC simulation for Mach 19 flow over a circular cylinder at 70 km with $T_{wall} = 1000$ K.

The first test case considers a Mach 19 flow at ambient conditions corresponding to an altitude of 70 km. The re-entry geometry selected for the first validation case is a circular cylinder maintained at the same uniform surface temperature of 1000 K as the training

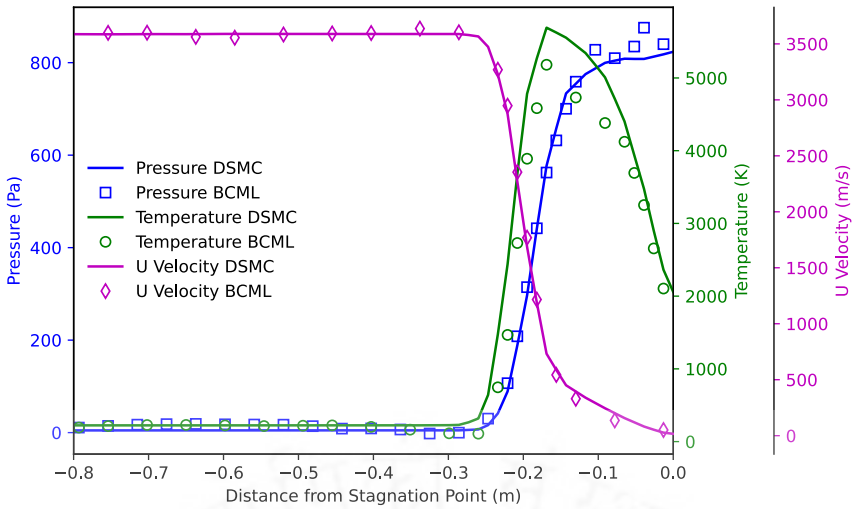

Figure 6.8: Variation of flow properties along the stagnation line for Mach 12 flow over an arbitrary bluff body at 69 km with $T_{wall} = 1000$ K

data. The inlet conditions for this problem were not part of the training dataset for the S_{16} Mach BCML model. Figure 6.7 presents a comparison of flow-field contours of pressure, translational temperature (T_{trans}), and the two velocity components obtained from the S_{16} BCML model and the corresponding DSMC simulations. The associated line profile is shown in Figure 6.6. The peak translational temperature in the post-shock region predicted by the BCML model is 15,631.7 K, which is approximately 2.549% lower than the DSMC prediction (16,040.4 K). The peak post-shock pressure predicted by the BCML model is 3287.61 Pa, corresponding to a deviation of about 6.839% from the DSMC value of 3529.06 Pa. A qualitative comparison indicates that the wave structures predicted by the S_{16} BCML model closely resemble those captured by the DSMC solution. Similar to previous BCML tests, the current BCML model for frozen hypersonic flows at higher altitudes accurately captures the shock structure and other wave patterns. The shock stand-off distance estimated by the BCML model closely matches that obtained from the DSMC simulations.

The second test case examines a Mach 12 flow under ambient conditions corresponding to an altitude of 69 km, with an arbitrarily shaped bluff body as the test geometry. In contrast to the previous validation case, the geometry was not included in the training dataset for the S_{16} BCML model. Figure 6.8 presents the centerline plot of selected flow properties upstream of the arbitrarily shaped bluff body. The peak translational temperature in the post-shock region predicted by the BCML model is 5,181.3 K, which is approximately 8.95% lower than the DSMC prediction (5,691.2 K). Compared with temperatures, the BCML model's peak pressure matches the DSMC peak pressure relatively better. The peak post-shock pressure predicted by the BCML model is 875.641 Pa, corresponding to a deviation of about 6.303% from the DSMC value of 823.57 Pa. Relatively larger discrepancies are observed at a few localised regions near the edges of the bow-shock away from the centerline and in the far-wake region downstream of the bluff body. The L_2 norm of the error over the entire

computational domain is found to be less than 1.7% for temperature and less than 5.93% for pressure. The error norms for the two test cases are summarised in Table 6.2. It should be noted that the BCML model predicts the entire flow field in less than 30 s, which is much faster than DSMC simulations that take around 500 hours. The BCML model's accuracy is reasonable, especially given its ability to handle arbitrary geometries.

Table 6.2: Initial and boundary conditions for the validation cases used to test the BCML model, along with the corresponding error analysis.

Parameter	Circular Cylinder	Arbitrary Geometry
Pressure (Pa)	7.698	4.604
Temperature (K)	246.5	224.863
Inlet Velocity (m/s)	5979.529	3606.992
L_2 error (Pressure)	4.095%	5.929%
L_2 error (T_{trans})	1.605%	1.691%
L_2 error ($U_{magnitude}$)	8.372%	2.152%

6.2.4 BCML-based Meshing Algorithm for DSMC

As discussed earlier, an optimised mesh is crucial for running the DSMC simulation efficiently. Bird emphasises that the mesh size must be based on the local mean free path to maintain DSMC's validity. It is noteworthy that sharp gradients in shocks and boundary layers require fine resolution. In addition to capturing shock structures, an optimised grid reduces statistical noise by ensuring sufficient particles per cell. It captures critical flow features such as heat transfer and shear stress near walls. Proper meshing minimises computational costs, ensures solution convergence through grid independence studies, and accurately represents the physics of hypersonic flow. Techniques such as adaptive mesh refinement (AMR) have been explored to dynamically adjust grid resolution based on local flow conditions, allowing for an optimal balance between computational cost and accuracy.

A new adaptive mesh-generation methodology specifically for DSMC is developed in this thesis. Generally, a spatially varying scalar field is used to control the local mesh resolution through a log-compressed length-field formulation. The proposed approach combines neighbourhood-based minimum propagation, logarithmic smoothing, and iterative Delaunay refinement to generate meshes that are both resolution-efficient and numerically robust. Geometry-aware boundary discretisation and automated boundary classification are incorporated to ensure compatibility with the in-house DSMC code. The resulting framework enables fully automated generation of solver-ready meshes while maintaining strict control over element quality and spatial grading.

One of the key outputs of the BCML model is the mean free path, which is the average distance a particle travels between successive collisions with other particles or obstacles. It

is inversely proportional to the density of the medium in which the particle is moving. The expression of mean free path (λ) for the power-law/VHS gas is:

$$\lambda = \frac{1}{\sqrt{2}\, n\, \sigma_{\text{ref}}} \left(\frac{T_{\text{ref}}}{T} \right)^{\omega} \tag{6.1}$$

where n is the number density, T is the local temperature and σ_{ref} is the reference total collision cross-section estimated at T_{ref} reference temperature. In the present work, adaptivity is driven by BCML-generated mean free path in the flow field. The ratio of the characteristic cell size (Δx) to the local mean free path, $\Delta x/\lambda$, plays a critical role in determining the accuracy and reliability of simulation results. For a mesh to be considered suitable, this ratio should ideally be less than or equal to 1 throughout the computational domain. If Δx is significantly larger than λ, important molecular collisions may be missed or misrepresented, leading to non-physical results and loss of accuracy in capturing gradients of flow properties. In contrast, an excessively small Δx relative to λ leads to unnecessary computational cost without a proportional gain in accuracy. This is especially true for DSMC codes following the ray tracing algorithm. Therefore, maintaining an optimal $\Delta x/\lambda$ ratio ensures that the mesh is fine enough to accurately resolve the physics of molecular motion and collisions while also being computationally efficient.

The refinement criteria chosen in the present work for DSMC simulations are based on the fraction $\Delta x/\lambda$. The Python-based new meshing code, based on algorithms by Chew and Paul [81], can generate a grid based on the required refinement using a multi-level threshold. The mesh generation code is written in Python. It employs `NumPy` (numeric Python) for array operations and linear algebra, and `SciPy` (scientific Python) for spatial indexing using the `cKDTree` module and triangulation using the `delaunay` module. In addition, the code uses the `Shapely` library for geometry-related operations, such as point-in-polygon tests and the exclusion of solid regions. This combination provides a robust and computationally efficient platform for adaptive mesh refinement.

The present algorithm constructs an unstructured mesh whose local resolution is governed by a spatially varying physical length scale derived from flow-field information. Such an approach is particularly suited for rarefied and transitional flow simulations, where the local mean free path varies strongly across the domain and dictates the required mesh resolution. The computational domain $\Omega \subset \mathbb{R}^2$ is assumed to be populated by a discrete set of sampling locations $\{\mathbf{x}_i\}_{i=1}^{N}$, at which a scalar physical length scale $\lambda(\mathbf{x}_i)$ is prescribed. The objective is to generate a mesh in which the characteristic edge length $h(\mathbf{x})$ is proportional to $\lambda(\mathbf{x})$, while maintaining smoothness, numerical robustness, and geometric fidelity. To avoid numerical pathologies associated with extremely small values of the physical length scale, the input field is first regularised by enforcing a lower bound,

$$\lambda(\mathbf{x}) \leftarrow \max\left(\lambda(\mathbf{x}), \lambda_{\min}\right),$$

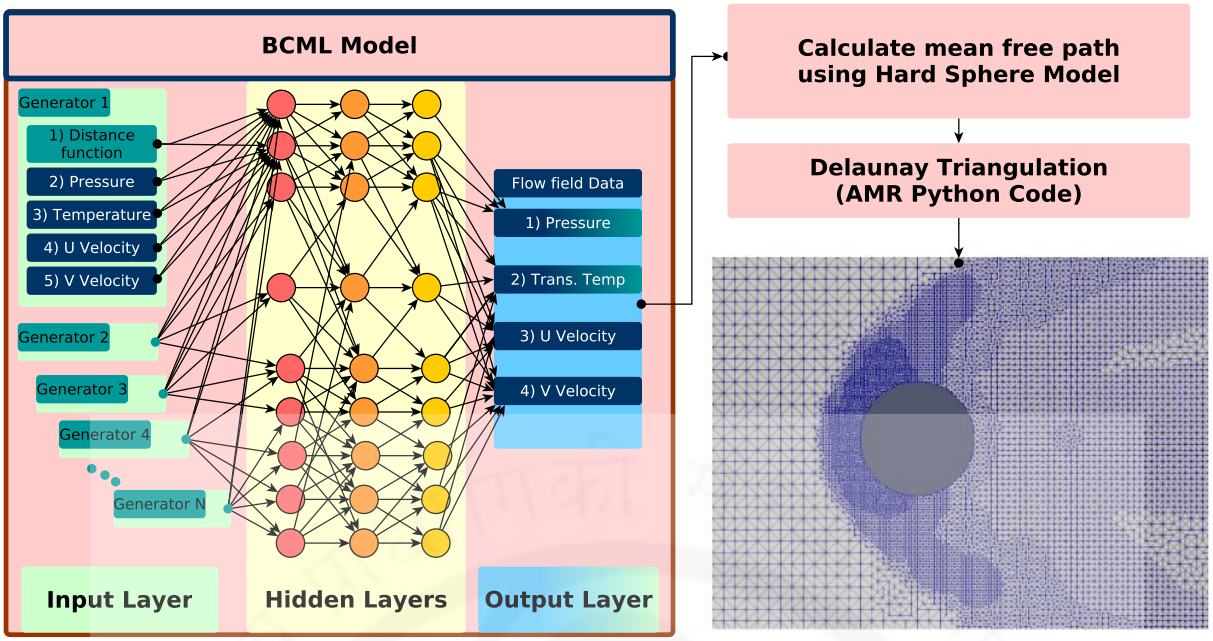

Figure 6.9: Schematic of the BCML model for frozen hypersonic flows over re-entry geometries at rarefied ambient conditions.

where $\lambda_{\min}$ is a user-defined cutoff. This ensures that the resulting mesh spacing remains finite throughout the domain and prevents excessive node clustering in regions where the physical model predicts vanishing length scales.

A direct linear mapping between the physical length scale and the mesh size can lead to excessive disparity in element sizes when λ spans several orders of magnitude. To mitigate this issue, a logarithmically compressed power-law mapping is employed. A reference value λ_{ref}, chosen as the median of the sampled field, is introduced to normalise the distribution. The preliminary target mesh size is then defined as

$$h_{\text{raw}}(\mathbf{x}) = C\,\lambda_{\text{ref}} \left(\frac{\lambda(\mathbf{x})}{\lambda_{\text{ref}}} \right)^{\alpha},$$

where C is a prescribed resolution ratio and $0 < \alpha < 1$ controls the degree of compression. This formulation preserves the relative ordering of refinement while preventing excessive refinement in regions of extremely small λ. The resulting target size is further constrained by global bounds,

$$h_{\min} \leq h_{\text{raw}}(\mathbf{x}) \leq h_{\max},$$

which impose absolute limits on the finest and coarsest admissible mesh resolution.

To prevent abrupt changes in the mesh size field, a local minimum propagation is applied. For each sampling location $\mathbf{x}_i$, all neighbouring points within a prescribed buffer radius r_b are identified, and the target size is modified according to

$$h_i \leftarrow \min \left(h_i, \min_{j \in \mathcal{N}(i)} h_j \right),$$

where $\mathcal{N}(i)$ denotes the set of neighbours within the buffer region. This operation ensures that coarse elements do not appear immediately adjacent to regions requiring fine resolution, thereby enforcing a minimum refinement influence radius. To further improve metric smoothness and mesh quality, the target size field is smoothed in logarithmic space. Defining $\ell_i = \log h_i$, a relaxed neighbour-averaging procedure is applied iteratively,

$$\ell_i^{(k+1)} = (1-\omega)\,\ell_i^{(k)} + \omega \frac{1}{|\mathcal{K}(i)|} \sum_{j \in \mathcal{K}(i)} \ell_j^{(k)},$$

where $\mathcal{K}(i)$ denotes a fixed-size local neighbour set and ω is a relaxation parameter. Performing the smoothing in logarithmic space ensures that relative, rather than absolute, variations in mesh size are attenuated. After a small number of iterations, the smoothed target size is recovered via $h_i = \exp(\ell_i)$.

During mesh refinement, the target size must be evaluated at arbitrary spatial locations. This is accomplished using an inverse-distance-weighted interpolation. For a query point $\mathbf{x}$, the local target size is computed as

$$h(\mathbf{x}) = \sum_{j=1}^{k} w_j(\mathbf{x})\, h_j, \qquad w_j = \frac{(d_j^2 + \varepsilon^2)^{-1}}{\sum_m (d_m^2 + \varepsilon^2)^{-1}},$$

where $d_j = \|\mathbf{x} - \mathbf{x}_j\|$ and ε is a small blending parameter that prevents singular behavior.

Starting from an initial coarse distribution of points covering the computational domain, an iterative Delaunay-based refinement procedure is applied. At each iteration, a triangulation is constructed, and all unique edges are examined. For an edge connecting vertices $\mathbf{v}_1$ and $\mathbf{v}_2$, the edge length is

$$L_e = \|\mathbf{v}_2 - \mathbf{v}_1\|,$$

and the local target size is evaluated at the midpoint $\mathbf{x}_m = (\mathbf{v}_1 + \mathbf{v}_2)/2$. If the condition

$$L_e > \gamma\, h(\mathbf{x}_m)$$

is satisfied, where $\gamma > 1$ is a safety factor, the midpoint is inserted as a new mesh vertex. The triangulation is then recomputed, and the process is repeated until no edges violate the metric constraint or a maximum number of refinement cycles is reached. This procedure ensures that all edges conform to the prescribed size field.

Solid boundaries are represented using a combination of straight segments and circular arcs. Each boundary is initially sampled at high resolution and subsequently resampled adaptively using an arc-length accumulation strategy. Along the boundary, a new point is retained whenever the accumulated distance exceeds the local target size evaluated slightly offset from the surface along the outward normal direction. This approach ensures consistency between boundary and interior resolution while avoiding unnecessary clustering of boundary

nodes. After assembling the refined interior points, boundary points, and outer domain boundaries into a single point set, a global Delaunay triangulation is constructed. Triangular elements whose centroids lie within the solid region are identified using point-in-polygon tests and removed, thereby excising the bluff body from the computational domain. Boundary edges are identified as those belonging to a single triangle and are classified by proximity to known boundary segments, enabling the assignment of appropriate boundary conditions.

Finally, the two-dimensional mesh is extruded uniformly in the third dimension to generate prismatic elements of prescribed height. Each triangular element is mapped to a six-node prism, and each boundary edge is mapped to a quadrilateral face. The resulting mesh is suitable for three-dimensional simulations under the spanwise-homogeneity assumption. Overall, the algorithm combines physics-informed metric construction, logarithmic smoothing, strict Delaunay refinement, and adaptive boundary discretisation to produce meshes that efficiently resolve rarefied flow features while avoiding unnecessary over-refinement. The overall algorithm of the code involves:

Algorithm 1 Density-driven unstructured mesh generation

```
Read sampling locations x_i and physical length scale λ(x_i)
Enforce lower bound on λ
Compute target mesh size using compressed power-law mapping
Apply global bounds to mesh size
Propagate local minimum within buffer radius
Smooth mesh size in logarithmic space
Construct continuous size evaluator
repeat
    Construct Delaunay triangulation
    for each unique edge do
        Evaluate edge length and local target size
        if edge violates size constraint then
            Insert midpoint
        end if
    end for
until no new vertices inserted or iteration limit reached
Adaptively discretise solid boundaries
Remove elements inside the solid region
Classify boundary edges
Extrude mesh to prismatic elements
```

The combination of BCML-generated flow fields and a BCML-assisted optimised DSMC mesh has been examined extensively. An initial mesh is first constructed to simulate hypersonic flow over a circular cylinder at freestream conditions corresponding to an altitude of 75 km and a Mach number of 30. These ambient conditions are not included in the BCML model's training dataset, thereby enabling an assessment of the generalisation capability of the proposed BCML-assisted meshing strategy. Based on the BCML prediction of the

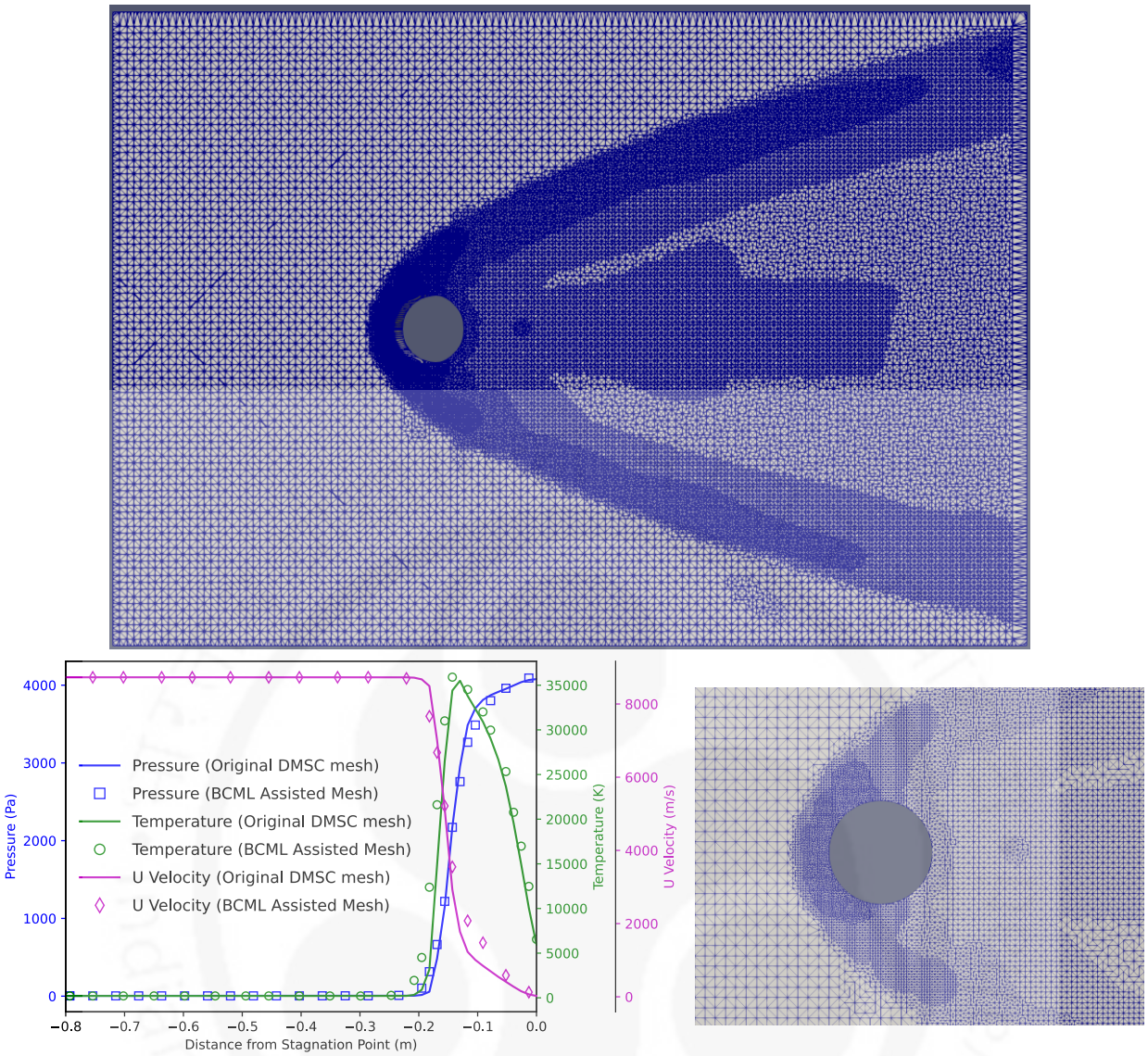

Figure 6.10: Final optimised BCML Assisted DSMC mesh for Mach 30 flow over Circular cylinder at altitude 75 km: full view (top) and zoomed in view (bottom right), along with comparison of properties along the stagnation line for the BCML Assisted mesh (60,634 cells) with a refined DSMC mesh (397,795 cells).

local mean free path, an optimised mesh is generated, as shown in Figure 6.10. An initial unstructured mesh containing approximately 1.0×10^5 cells of approximately uniform size was first generated, followed by application of the proposed meshing algorithm, resulting in a reduced mesh comprising approximately 6.0×10^4 cells. Hypersonic flow over a circular cylinder is then simulated using the DSMC solver on the BCML-assisted, optimised mesh to obtain the final flow field. The DSMC solver was executed on a workstation utilising 12 CPU cores, and the corresponding computational time was recorded.

To evaluate the performance of the DSMC solution obtained with the BCML-assisted optimised mesh, an additional set of DSMC simulations is performed on a highly refined

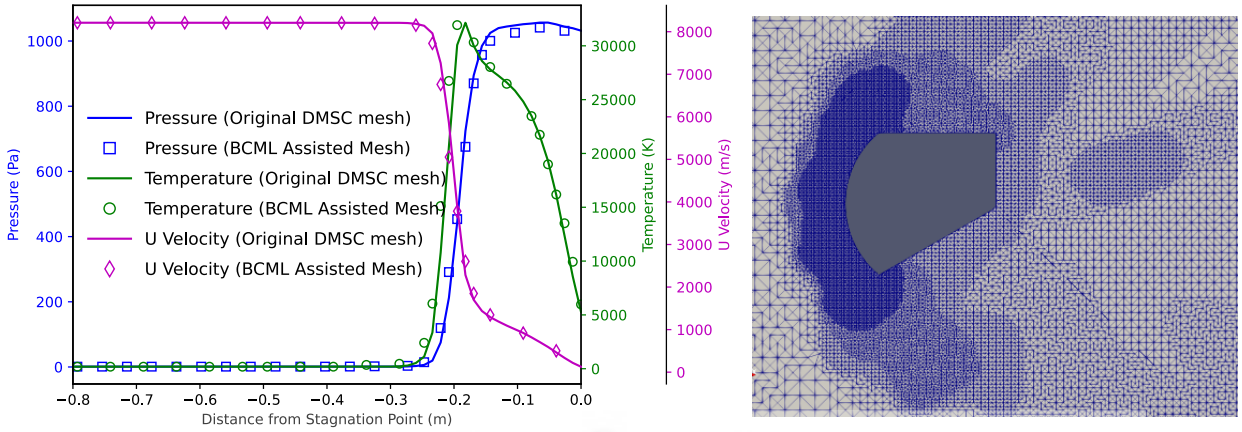

Figure 6.11: Comparison of properties (left along the stagnation line for the BCML Assisted mesh (90,344 cells) with a refined DSMC mesh (397,376 cells) for Mach 30 flow over an arbitrarily-shaped geometry at altitude 82 km and zoomed in view (right) near the stagnation point.

unstructured mesh without adaptive mesh refinement. The geometric parameters of this reference mesh are determined by iteratively doubling the number of cells in the domain and performing DSMC simulations until the flow properties converge. Since DSMC is a statistical method, convergence is assumed when the difference in flow properties between two successive mesh refinements is less than 1%. It was found that the DSMC simulation with a mesh consisting of approximately 4.0×10^5 cells was sufficient.

The magnified view of the BCML-assisted mesh near the bow shock and stagnation point is shown in Figure. 6.10 (bottom left). The BCML-assisted meshing approach effectively eliminates unproductive cells across the computational domain, thereby significantly reducing the overall computational cost. Figure. 6.10 (bottom right) shows a comparison of various properties along the stagnation line of the circular cylinder. The results demonstrate excellent agreement between the flow-field predictions obtained with the BCML-assisted mesh and those from the uniform refined DSMC mesh. Since the comparison is between results from two DSMC simulations, strong agreement is expected. In terms of computational efficiency, the original DSMC mesh required 228.21 hours on a 12-CPU-core workstation, whereas the BCML-assisted mesh on the same 12-CPU core workstation achieved the same solution accuracy in only 30.69 hours, resulting in a reduction of approximately 86.53% in computational time.

Similar flow-field and computational performance analyses were conducted for hypersonic flow over an arbitrarily shaped geometry at an altitude of 82 km and a Mach number of 30, following the same procedure as the simulation around the circular cylinder. Again, the arbitrarily shaped geometry was not included in the training dataset used for BCML model development. Consequently, the BCML model lacks explicit geometric information for this configuration, thereby enabling a stringent assessment of its generalisation to previously

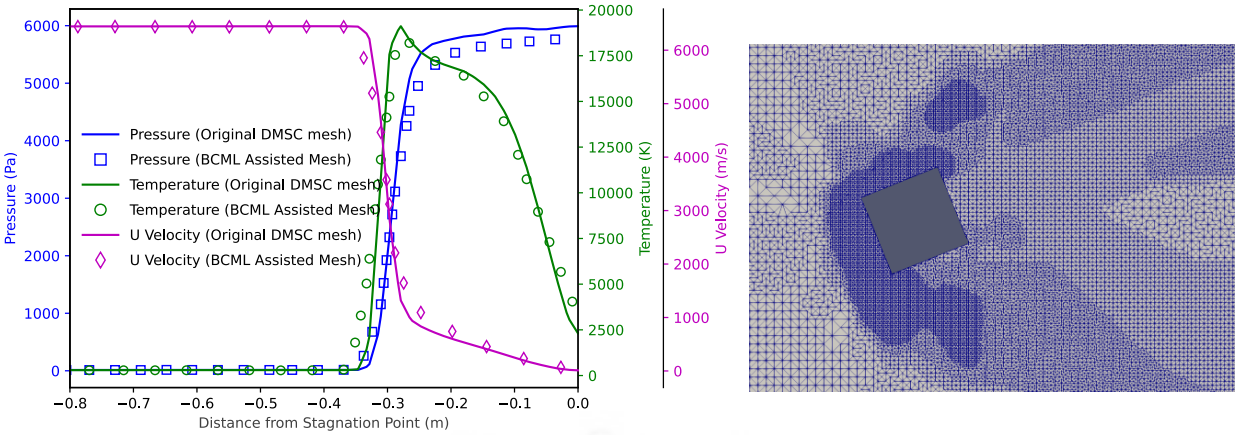

Figure 6.12: Comparison of properties (left along the stagnation line for the BCML Assisted mesh (87,365 cells) with a refined DSMC mesh (417,600 cells) for Mach 20 flow over a square cylinder at altitude 63 km and zoomed in view (right) near the stagnation point.

unseen geometries. Subsequently, the proposed BCML-assisted meshing algorithm was applied to an initial pre-generated mesh containing approximately 1.0×10^5 cells, resulting in a reduced mesh comprising approximately 8.3×10^4 cells. A magnified view of the BCML-assisted mesh in the shock-dominated region is presented in Figure. 6.11 (right). Again, the uniform refined DSMC mesh, after grid independence tests, consisting of approximately 4.0×10^5 cells, was generated for the same arbitrarily shaped geometry. Flow simulations conducted on both the original DSMC mesh and the BCML-assisted mesh under identical numerical and physical conditions yielded flow-field data that were extracted along the stagnation line after steady state was achieved.

In the flow-field analysis, the peak temperature and pressure predicted using the original DSMC mesh are 31,541.4 K and 1051.2 Pa, respectively. In comparison, the BCML-assisted mesh yields peak values of 32,036.8 K for temperature and 1039.52 Pa for pressure, indicating a marginal over-prediction. The corresponding differences in peak temperature and pressure are less than 1%, demonstrating good agreement between the BCML-assisted and original DSMC mesh solutions, as depicted in Figure 6.11 (left). Compared to the simulation for a circular cylinder at an altitude of 75 km, it was expected that the number of cells in the mesh around the arbitrarily shaped geometry would be fewer. However, due to the shape and flow asymmetry, the optimised mesh contains a large number of cells. However, since the number of cells in the optimised BCML-assisted mesh is much lower than that in the uniform refined mesh, the overall computational cost is reduced. In terms of computational efficiency, the original DSMC mesh required 309.26 hours on a 12-CPU-core workstation, whereas the BCML-assisted mesh on the same 12-CPU core workstation achieved the same solution accuracy in only 49.536 hours, resulting in a reduction of approximately 83.982% in computational time.

Table 6.3: Simulation conditions with T_{wall} = 1000 K and computational cost/time comparison for BCML-assisted mesh with the original DSMC mesh for various geometries.

Parameter	Square Cylinder	Circular Cylinder	Arbitrarily Shaped Body
Altitude (km)	63.00	75.00	82.00
Pressure (Pa)	13.332	3.544	1.010
Temperature (K)	295.35	211.1	186.7
AoA	22°	0°	0°
Inlet Velocity (m/s)	6456.216	8737.160	8216.71
Uniform Refined DSMC Mesh Cells	417,600	397,795	397,376
Uniform Refined DSMC Mesh (hrs)	311.16	228.21	282.00
BCML Assisted Mesh Cells	87,365	60,634	90,344
BCML Assisted Mesh (hrs)	76.742	30.69	30.54
Speed Up (× times)	4.054	7.435	9.233
Reduction in Computational Cost (%)	75.33%	86.53%	89.172%

Table 6.4: Summary of computational cost associated with the proposed BCML frameworks and BCML-assisted solvers.

Framework	Offline Cost	Training Cost	Online Cost	BCML-assisted Cost
Low-altitude NSF-BCML	500 NSF-CFD simulations (∼12 h/case)	∼185 min	<10 s	22–25% speed-up
Reacting NSF-BCML + ChemDNN	29 reacting hy2Foam simulations (48 h/case)	∼30 min	<8 s	N/A
High-altitude DSMC-BCML (Meshing)	154 DSMC simulations (∼20 h/case)	∼240 min	<15 s	76.742 h (vs 311.16 h)
Reacting DSMC-BCML + ChemDNN	154 reacting DSMC simulations (∼20 h/case)	∼280 min	<20 s	N/A
SurfPropDNN	Surface-property DSMC database	∼5 min	∼6 s	N/A
BCML-assisted FVM (Cylinder)	BCML initial field	N/A	N/A	17.78 h (vs 22.72 h)
BCML-assisted FVM (Square)	BCML initial field	N/A	N/A	14.87 h (vs 19.09 h)
BCML-assisted FVM (FIRE-II)	BCML initial field	N/A	N/A	31.57 h (vs 41.61 h)

Notation: NSF = Navier–Stokes–Fourier; DSMC = Direct Simulation Monte Carlo; N/A = Not Applicable; h = CPU hours.

Finally, similar flow-field and computational performance analyses were conducted for hypersonic flow over a square cylinder at an altitude of 63 km and a Mach number of 20. A uniform refined DSMC mesh consisting of approximately 4.17×10^5 cells was generated. The BCML-assisted meshing algorithm generated an optimised mesh comprising approximately 8.7×10^4 cells. A magnified view of the BCML-assisted mesh in the shock-dominated region is presented in Figure 6.12 (right). Flow simulations conducted on both the uniform refined mesh and the BCML-assisted mesh under identical numerical and physical conditions demonstrate excellent agreement between the BCML-assisted and original DSMC mesh solutions as depicted in Figure 6.12 (left). The DSMC simulation employing the uniform refined mesh ran for 311.16 hours on a 12-CPU-core workstation, whereas the BCML-assisted mesh on the

same 12-CPU core workstation achieved the same solution accuracy in only 76.742 hours, resulting in a reduction of approximately 75.33% in computational time. The summary of the details of all three cases can be found in Table 6.3.

6.3 Computational Cost and Workflow Considerations

The computational speed-up associated with the BCML framework primarily occurs during the online prediction stage, after completion of the offline dataset generation and neural network training processes. The overall workflow, therefore, includes computationally expensive NSF-CFD/DSMC-based data generation and model training stages prior to deployment. Consequently, the BCML framework should primarily be viewed as an assisting tool for rapid flow-field estimation, convergence acceleration, and optimised mesh generation within the trained operating regime, rather than as a complete replacement for high-fidelity NSF/DSMC solvers.

Table 6.4 summarises the computational cost associated with the proposed BCML frameworks, including offline database generation, neural-network training, online prediction, and BCML-assisted solver performance. Although the offline data-generation stage is computationally intensive, the trained BCML models enable rapid predictions and significantly reduce the computational cost of repeated simulations and solver-assisted workflows within their trained regime of applicability.

Chapter 7

Conclusion and Future Scope

A preliminary attempt has been made to develop a novel machine learning-based model capable of predicting the flow field without explicitly solving the NSF equations or running the DSMC algorithm. The intention of designing the new model was to study external supersonic/hypersonic flow independently of the shape of the bluff body, accounting for the influence of various surrounding boundaries. The concept of a generator in the present work models the effect of various boundary conditions on a point within the domain. The lengths of the generators from the points along the planar angle from the point to the boundary condition, along with the set of Dirichlet/Neumann conditions for various properties, serve as the input layer to the deep neural network. A total of four BCML frameworks have been proposed. Two of these can predict flow-field properties in frozen and chemically reactive hypersonic flow at low altitudes, respectively, using training data generated by a continuum-based NS solver. Additionally, the other two can predict flow-field properties in frozen and chemically reactive hypersonic flow conditions at high altitude, respectively, using data generated by the DSMC method. Flow-field data over a wide range of altitudes (40–84 km) and Mach numbers (2–35) are included in the training data for the BCML models, covering the entire re-entry trajectory.

Table 7.1: Summary of datasets used for BCML training.

Model	**Solver/Geometry**	**Simulations**	**Cells**
Low-altitude BCML (Frozen)	NSF-CFD / Circular Cylinder	80	214244
	NSF-CFD / Square Cylinder	420	104400
Low-altitude BCML (Reacting)	*hy2Foam* / Stardust	29	30393
High-altitude BCML (Frozen)	DSMC / Square Cylinder	80	104400
	DSMC / Circular Cylinder	36	214244
	DSMC / Arbitrary-shaped Cylinder	38	$\sim$ 70500
High-altitude BCML (Reacting)	DSMC / Square Cylinder	80	104400
	DSMC / Circular Cylinder	36	214244
	DSMC / Arbitrary-shaped Cylinder	38	$\sim$ 70500

The total dataset, summarised in Table 7.1, is used to train the BCML frameworks, which include 500 NSF-CFD simulations for the low-altitude BCML model (frozen), 29

reacting *hy2Foam* simulations, and 154 DSMC simulations for each of the frozen- and reacting-high-altitude BCML frameworks. The cumulative dataset comprises approximately 0.5 billion spatial data samples from NSF-CFD/hy2Foam simulations and approximately 18.3724 million from DSMC simulations.

One of the important outcomes of the present work is the development of the high-altitude DSMC-based BCML framework combined with the ChemDNN model for estimating thermophysical and species properties in hypersonic rarefied flows. The BCML models were validated for a wide range of test cases with arbitrary inlet and boundary conditions, which were not a part of the training data. It was found that the flow field predicted by the BCML model was in reasonable agreement with those obtained from the NS-CFD/DSMC solver. The BCML model captured the major bow-shock characteristics and overall flow trends with reasonable agreement with the NSF-CFD/DSMC solutions for the validation cases considered. Furthermore, the new BCML model can capture other wave features with reasonable accuracy. The overall error in any of the flow field properties was less than 5% compared to that predicted by the NS-CFD/DSMC method. *The main benefit of the BCML models was negligible computational cost ($< 10s$) and very high speed-up ($>> 1000$) in estimating the flow field compared to the traditional solver.*

Further, the BCML model was used with the FVM solver to accelerate the CFD simulations. Hypersonic flow simulations with BCML-generated flow fields were at least 25% faster for the given FVM solver. In addition, a BCML-assisted mesh generation framework has been developed for DSMC simulations of hypersonic rarefied flows. The approach uses BCML-predicted flow fields to construct a mesh based on the local mean free path, enabling refinement in regions of strong non-equilibrium effects while avoiding unnecessary over-resolution elsewhere. The BCML-assisted mesh generation algorithm combined logarithmic smoothing with Delaunay triangulation to produce high-quality unstructured meshes that meet resolution constraints. Benchmark simulations showed that DSMC solutions obtained on the BCML-assisted optimised meshes closely match those obtained on highly refined static meshes, while requiring significantly fewer computational cells and reduced computational time. *The utility of the BCML model should be viewed primarily as an assisting framework for CFD and DSMC simulations by providing improved initial conditions, aiding mesh design, and generating approximate flow-field estimates, rather than as a complete replacement for conventional CFD or DSMC solvers.*

The overall accuracy improves as the number of generators increases. Although the possibility of ambiguity in the boundary-to-field mapping cannot be completely ruled out for arbitrary geometries, no such ambiguity was observed for the range of flow configurations considered in the present study. Increasing the number of generators also increases the dimensionality of the input space, thereby introducing a trade-off between accuracy and

computational cost. Also, the adopted distance function naturally emphasises the influence of nearby boundaries while reducing the contribution of distant boundaries, enabling the network to learn localised flow features, including regions influenced by shocks and strong gradients, from the training data. A rigorous mathematical analysis of the approximation properties and theoretical foundations of the BCML framework remains an important direction for future research.

The overall contributions of the present work can be summarised as:

1. A new Boundary Condition-based Machine Learning (BCML) framework is introduced, which uses generators to model how surrounding boundary conditions and geometry influence arbitrary points in a hypersonic flow field.

2. Separate BCML frameworks have been reported spanning atmospheric re-entry regimes from 40 to 84 km, utilising continuum-based CFD for lower altitudes and DSMC simulations to account for high-altitude rarefaction and nonequilibrium effects.

3. Auxiliary neural networks, ChemDNN and SurfPropDNN, have been developed to predict reacting-species mole fractions and critical surface properties such as heat flux and pressure coefficients under rarefied hypersonic conditions.

4. Practical engineering utility is demonstrated by leveraging BCML predictions to accelerate conventional CFD convergence and to automate mesh adaptation in DSMC simulations based on BCML-predicted local flow properties.

It should be emphasised that the present study reports preliminary findings in the initial stage of development of the BCML framework. The following needs to be undertaken as part of the future work:

- In the immediate future, the primary objective of the high-altitude DSMC-based BCML framework, particularly in rarefied re-entry flows, is to facilitate efficient domain decomposition strategies that improve load balancing in multiprocessor simulations. A potential application of the proposed BCML framework is in improving load balancing for DSMC and hybrid CFD-DSMC solvers with optimised domain decomposition. Since the BCML model can rapidly predict density distribution prior to the actual numerical simulation, it provides an estimate of the computational cost across different regions of the domain. Regions characterised by small mean free paths will have a higher particle density and can therefore be allocated a larger number of processors. Conversely, low-density regions can be assigned fewer computational resources. Such an adaptive domain decomposition strategy has the potential to improve load balancing, reduce processor idle time, and enhance the parallel efficiency of large-scale DSMC and hybrid CFD-DSMC simulations. This is currently in progress.

- The BCML framework can be used to improve the efficiency of an NSF–DSMC hybrid solver. Generally, the hybrid solver starts with a solution from the FVM solver, which is then split between the NSF and DSMC zones. This continues iteratively until the boundary converges. BCML can significantly improve efficiency by accurately predicting the final continuum breakdown regions, thereby reducing computational cost.

- The BCML algorithm can be easily extended to three-dimensional simulations by modelling generators that span the 4π solid angle rather than the current planar angles. An appropriate 3D-BCML model can be trained with sufficient data from three-dimensional simulations.

- Further investigation is required to establish a deeper understanding of the underlying mathematical principles governing the BCML model architecture and its learning dynamics. Due to the high dimensionality of the input layer, the topology is extremely complex and may include regions of the multidimensional domain that are not present in the training data. Such regions can potentially introduce localised inaccuracies in model predictions, as the neural network may not have been sufficiently exposed to all areas of the input topology during training.

- Hypersonic flow is a complex physical phenomenon in which non-equilibrium effects and finite-rate chemistry play an important role in the evolution of the flow, the structure of the shock, and the associated wave patterns. The BCML model, in its current form, relies solely on boundary conditions and is therefore limited by the absence of explicit chemical reaction modelling and higher-order moment effects, which are inherently flow phenomena and not solely determined by boundary conditions. Future BCML models should incorporate physics more rigorously.

- In its current form, SurfPropDNN predicts the surface properties directly from the geometry of the bluff body and the far-field flow properties. The model does not use the flow-field temperature and velocity as inputs, especially in the vicinity of the bluff body. This approach is computationally simple and serves as a preliminary demonstration primarily for completeness in the overall flow analysis. A more physically consistent framework would incorporate flow-field information as inputs to the surface model, allowing surface quantities to be derived directly from the underlying flow physics. This limitation will be addressed in future work.

- Although BCML has presently been developed and validated for hypersonic re-entry flows, its underlying framework is not restricted to this class of problems. BCML can be extended to a wide range of compressible and rarefied flow applications, including nozzle flows with expansion-induced non-equilibrium, high-altitude plume expansions,

shock–boundary layer interactions, micro-nozzle flows, and other internal or external aerothermodynamic configurations. With appropriate training datasets covering the relevant flow regimes, the model can generalise beyond re-entry flows. In fact, the BCML algorithm is independent of the methodology used to generate the training data. The concept of the BCML model can be used in conjunction with molecular dynamics, discontinuous Galerkin, smooth particle hydrodynamics, or even for applications and computational methods beyond fluid flows.

Data Availability Statement

The training and validation codes developed for the Boundary Condition-Based Machine Learning (BCML) flow-field prediction framework are publicly available on the GitHub repository (`https://github.com/saiprakash2026`). The datasets generated and analysed during the current study are available from the author(s) upon request. Additional data, scripts, or supporting materials required to reproduce or further investigate the reported results will also be made available upon reasonable request.

Bibliography

[1] M. Fang, Z. H. Li, Z. H. Li, J. Liang, and Y. H. Zhang, "DSMC modeling of rarefied ionization reactions and applications to hypervelocity spacecraft reentry flows," *Advances in Aerodynamics*, vol. 2, no. 1, p. 7, 2020.

[2] E. Farah and T. R. Teschner, *Aerodynamic performance investigation through different chemistry modelling approaches for space re-entry vehicles using the DSMC method*. Tech. Report, Cranfield University, 2022.

[3] R. Votta, A. Schettino, and A. Bonfiglioli, "Hypersonic high altitude aerothermodynamics of a space re-entry vehicle," *Aerospace Science and Technology*, vol. 25, no. 1, p. 253, 2013.

[4] C. Park, *Nonequilibrium Hypersonic Aerothermodynamics*. John Wiley and Sons, 1989.

[5] T. K. Mankodi, U. V. Bhandarkar, and B. P. Puranik, "Hypersonic flow over Stardust re-entry capsule using ab-initio based chemical reaction model," *Acta Astronautica*, vol. 162, p. 243, 2019.

[6] D. Burnett, "The distribution of molecular velocities and the mean motion in a non-uniform gas," *Proceedings of the London Mathematical Society*, vol. s2-40, no. 1, pp. 382–435, 1935.

[7] X. Zhong, R. K. Agarwal, and R. Balakrishnan, "Stability of Burnett equations for hypersonic flow positions," *AIAA Journal*, vol. 31, no. 6, pp. 1036–1041, 1993.

[8] H. Brenner, "Kinematics of volume transport," *Physica A: Statistical Mechanics and its Applications*, vol. 349, no. 1-2, pp. 11–59, 2005.

[9] H. Brenner, "Is the Navier-Stokes equation mistake-free?," *Physics of Fluids*, vol. 17, no. 6, p. 061102, 2005.

[10] H. Struchtrup, "The Onsager-Burnett equations: A set of stable second-order hydrodynamic equations for rarefied gases," *Annals of Physics*, vol. 327, no. 9, pp. 2032–2063, 2012.

[11] N. Singh, R. S. Jadhav, and A. Agrawal, "Derivation of stable Burnett equations for rarefied gas flows," *Physical Review E*, vol. 96, no. 1, p. 013106, 2017.

[12] R. S. Jadhav, N. Singh, and A. Agrawal, "Force-driven compressible plane Poiseuille flow by Onsager-Burnett equations," *Physics of Fluids*, vol. 29, no. 10, p. 102007, 2017.

[13] N. Singh and A. Agrawal, "Onsager's-principle-consistent 13-moment transport equations," *Physical Review E*, vol. 93, no. 6, p. 063111, 2016.

[14] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon Press, Oxford, UK, 1994.

[15] I. D. Boyd and T. E. Schwartzentruber, *Nonequilibrium gas dynamics and molecular simulation*, vol. 42. Cambridge University Press, 2017.

[16] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual review of fluid mechanics*, vol. 52, no. 1, p. 477, 2020.

[17] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo, "Fast flow field prediction over airfoils using deep learning approach," *Physics of Fluids*, vol. 31, no. 5, p. 057103, 2019.

[18] J. Z. Peng, N. Aubry, S. Zhu, Z. Chen, and W. T. Wu, "Geometry and boundary condition adaptive data-driven model of fluid flow based on deep convolutional neural networks," *Physics of Fluids*, vol. 33, no. 12, p. 123602, 2021.

[19] J. Ling and J. Templeton, "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier-Stokes uncertainty," *Physics of Fluids*, vol. 27, no. 8, p. 085103, 2015.

[20] Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan, "Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation," *Physics of Fluids*, vol. 30, no. 12, p. 125101, 2018.

[21] Z. Zhang, J. Geiger, J. Pohjalainen, A. E. D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, p. 1, 2018.

[22] H. Jiang, J. Liu, S. Luo, W. Huang, J. Wang, and M. Liu, "Thermochemical non-equilibrium effects on hypersonic shock wave/turbulent boundary-layer interaction," *Acta Astronautica*, vol. 192, p. 1, 2022.

[23] D. Sun, Z. Wang, F. Qu, and J. Bai, "A deep learning based prediction approach for the supercritical airfoil at transonic speeds," *Physics of Fluids*, vol. 33, no. 8, p. 086109, 2021.

[24] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, p. 686, 2019.

[25] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, p. 422, 2021.

[26] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," *Acta Mechanica Sinica*, vol. 37, no. 12, p. 1727, 2021.

[27] M. De Florio, E. Schiassi, B. D. Ganapol, and R. Furfaro, "Physics-informed neural networks for rarefied-gas dynamics: Thermal creep flow in the Bhatnagar–Gross–Krook approximation," *Physics of Fluids*, vol. 33, no. 4, p. 047110, 2021.

[28] L. Zhang, W. Ma, Q. Lou, and J. Zhang, "Simulation of rarefied gas flows using physics-informed neural network combined with discrete velocity method," *Physics of Fluids*, vol. 35, no. 7, p. 077124, 2023.

[29] I. Zanardi, S. Venturi, and M. Panesi, "Towards efficient simulations of non-equilibrium chemistry in hypersonic flows: a physics-informed neural network framework," in *AIAA SCITECH 2022 Forum*, (San Diego, California), p. 1639, 2022.

[30] D. Villanueva, B. Paez, A. Rodriguez, A. Chattopadhyay, V. K. Kotteda, R. Baez, J. Perez, J. Terrazas, and V. Kumar, "Field predictions of hypersonic cones using physics-informed neural networks," in *Fluids Engineering Division Summer Meeting*, vol. 85840, p. V002T05A015, ASME, 2022.

[31] J. M. Tucny, M. Lauricella, M. Durve, G. Guglielmo, A. Montessori, and S. Succi, "Physics-informed neural networks for microflows: Rarefied gas dynamics in cylinder arrays," *Journal of Computational Science*, vol. 87, p. 102575, 2025.

[32] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, "Stiff-PINN: Physics-informed neural network for stiff chemical kinetics," *The Journal of Physical Chemistry A*, vol. 125, no. 36, p. 8098, 2021.

[33] L. Liu, S. Liu, H. Xie, F. Xiong, T. Yu, M. Xiao, L. Liu, and H. Yong, "Discontinuity computing using physics-informed neural networks," *Journal of Scientific Computing*, vol. 98, no. 1, p. 22, 2024.

[34] J. Abbasi, A. D. Jagtap, B. Moseley, A. Hiorth, and P. Ø. Andersen, "Challenges and advancements in modeling shock fronts with physics-informed neural networks: A review and benchmarking study," *NeuroComputing*, vol. 657, p. 131440, 2025.

[35] J. Brownlee, *Deep learning for computer vision: image classification, object detection, and face recognition in python*. Machine Learning Mastery, 2019.

[36] A. A. Khan, A. A. Laghari, and S. A. Awan, "Machine learning in computer vision: A review.," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, p. 32, 2021.

[37] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, p. 362, 2020.

[38] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa, "A review Bayesian deep learning in healthcare: Applications and challenges," *IEEE Access*, vol. 10, p. 36538, 2022.

[39] M. Frank, D. Drikakis, and V. Charissis, "Machine learning methods for computational science and engineering," *Computation*, vol. 8, no. 1, p. 15, 2020.

[40] R. Vinuesa and S. L. Brunton, "Enhancing computational fluid dynamics with machine learning," *Nature Computational Science*, vol. 2, no. 6, p. 358, 2022.

[41] C. Cheng and G. T. Zhang, "Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems," *Water*, vol. 13, no. 4, p. 423, 2021.

[42] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, p. 1, 2017.

[43] H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian, "Deep neural networks for nonlinear model order reduction of unsteady flows," *Physics of Fluids*, vol. 32, no. 10, p. 105104, 2020.

[44] C. Kong, J. Chang, Z. Wang, Y. Li, and W. Bao, "Data-driven super-resolution reconstruction of supersonic flow field by convolutional neural networks," *AIP Advances*, vol. 11, p. 6, 2021.

[45] F. Lorenzen, A. Zargaran, and U. Janoske, "Potential of Physics-informed neural networks for solving fluid flow problems with parametric boundary conditions," *Physics of Fluids*, vol. 36, no. 3, p. 037143, 2024.

[46] A. D. Jagtap, Z. Mao, N. Adams, and G. E. Karniadakis, "Physics-informed neural networks for inverse problems in supersonic flows," *Journal of Computational Physics*, vol. 466, p. 111402, 2022.

[47] M. Guo, X. Deng, Y. Ma, Y. Tian, J. Le, and H. Zhang, "Hypersonic inlet flow field reconstruction dominated by shock wave and boundary layer based on small sample physics-informed neural networks," *Aerospace Science and Technology*, vol. 150, p. 109205, 2024.

[48] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[49] S. Fresca, L. Dede, and A. Manzoni, "A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs," *Journal of Scientific Computing*, vol. 87, p. 1, 2021.

[50] I. Scherl, B. Strom, J. K. Shang, O. Williams, B. L. Polagye, and S. L. Brunton, "Robust principal component analysis for modal decomposition of corrupt fluid flows," *Physical Review Fluids*, vol. 5, no. 5, p. 054401, 2020.

[51] J. Wang, C. He, R. Li, H. Chen, C. Zhai, and M. Zhang, "Flow field prediction of supercritical airfoils via variational autoencoder based deep learning framework," *Physics of Fluids*, vol. 33, no. 8, p. 086108, 2021.

[52] M. Sieber, C. O. Paschereit, and K. Oberleithner, "Spectral proper orthogonal decomposition," *Journal of Fluid Mechanics*, vol. 792, p. 798, 2016.

[53] S. E. Ahmed, O. San, A. Rasheed, and T. Iliescu, "Nonlinear proper orthogonal decomposition for convection-dominated flows," *Physics of Fluids*, vol. 33, no. 12, p. 121702, 2021.

[54] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annual review of fluid mechanics*, vol. 51, no. 1, p. 357, 2019.

[55] J. D. Anderson, *Hypersonic and high temperature gas dynamics*. AIAA, 1989.

[56] J. J. Shang and H. Yan, "High-enthalpy hypersonic flows," *Advances in Aerodynamics*, vol. 2, no. 1, p. 19, 2020.

[57] M. B. Gerdroodbary, *Aerodynamic heating in supersonic and hypersonic flows: advanced techniques for drag and aero-heating reduction*. First Edition, Elsevier, 2022.

[58] I. Nompelis, T. W. Drayna, and G. V. Candler, "A parallel unstructured implicit solver for hypersonic reacting flow simulation," in *Parallel Computational Fluid Dynamics 2005*, p. 389, Elsevier, 2006.

[59] R. J. Gollan and P. A. Jacobs, "About the formulation, verification and validation of the hypersonic flow solver eilmer," *International Journal for Numerical Methods in Fluids*, vol. 73, no. 1, p. 19, 2013.

[60] G. V. Candler, P. K. Subbareddy, and J. M. Brock, "Advances in computational fluid dynamics methods for hypersonic flows," *Journal of Spacecraft and Rockets*, vol. 52, no. 1, p. 17, 2015.

[61] V. Casseau, D. E. Espinoza, T. J. Scanlon, and R. E. Brown, "A two-temperature open-source CFD model for hypersonic reacting flows, part two: Multi-dimensional analysis," *Aerospace*, vol. 3, no. 4, p. 45, 2016.

[62] G. V. Candler, "Rate effects in hypersonic flows," *Annual Review of Fluid Mechanics*, vol. 51, no. 1, p. 379, 2019.

[63] W. L. Wang and I. Boyd, "Hybrid DSMC-CFD simulations of hypersonic flow over sharp and blunted bodies," in *36th AIAA Thermophysics Conference*, (Orlando, Florida), p. 3644, 2003.

[64] J. Moss, V. Dogra, J. Price, and D. Hash, "Comparison of DSMC and experimental results for hypersonic external flows," in *30th Thermophysics Conference*, (Křtiny Castle, Czech Republic), p. 2028, 1995.

[65] J. Moss, C. Glass, and F. Greene, "DSMC simulations of apollo capsule aerodynamics for hypersonic rarefied conditions," in *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, (San Francisco, California), p. 3577, 2006.

[66] X. Jin, P. Cheng, W. L. Chen, and H. Li, "Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," *Physics of Fluids*, vol. 30, no. 4, p. 047105, 2018.

[67] M. D. Ribeiro, A. Rehman, S. Ahmed, and A. Dengel, "DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks," *arXiv preprint arXiv:2004.08826*, 2020.

[68] L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa, "Convolutional-network models to predict wall-bounded turbulence from wall quantities," *Journal of Fluid Mechanics*, vol. 928, p. A27, 2021.

[69] Z. Mao, L. Lu, O. Marxen, T. A. Zaki, and G. E. Karniadakis, "DeepM&Mnet for hypersonics: predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators," *Journal of Computational Physics*, vol. 447, p. 110698, 2021.

[70] P. Novello, G. Poëtte, D. Lugato, S. Peluchon, and P. M. Congedo, "Accelerating hypersonic reentry simulations using deep learning-based hybridization (with guarantees)," *Journal of Computational Physics*, vol. 498, p. 112700, 2024.

[71] E. Monti, N. Singh, J. Sirignano, J. F. MacArt, M. Panesi, and G. Gori, "Physics-constrained deep learning-based model for non-equilibrium flows," in *: AIAA Scitech 2024 Forum*, (Orlando, Florida), p. 0654, 2024.

[72] G. Dai, C. Shao, Z. Zhu, W. Zhao, G. Fan, D. Tian, and W. Chen, "Hypersonic flow field and heat flux prediction model based on physics-informed neural networks with positional encoding," *Physics of Fluids*, vol. 37, no. 12, p. 126143, 2025.

[73] M. Schouler, Y. Prévereaud, and L. Mieussens, "Machine learning based reduced models for the aerothermodynamic and aerodynamic wall quantities in hypersonic rarefied conditions," *Acta Astronautica*, vol. 204, p. 83, 2023.

[74] B. Zhang, G. Cai, D. Gao, H. Weng, W. Wang, and B. He, "Development of convolutional neural network-based surrogate model for three-dimensional vacuum plume prediction via direct simulation monte carlo method," *Physics of Fluids*, vol. 36, no. 7, p. 077154, 2024.

[75] G. Bird, "Recent advances and current challenges for DSMC," *Computers & Mathematics with Applications*, vol. 35, no. 1, p. 1, 1998.

[76] S. Yao, W. Zhao, C. Wu, and W. Chen, "Nonlinear constitutive calculation method of rarefied flow based on deep convolution neural networks," *Physics of Fluids*, vol. 35, no. 9, p. 096103, 2023.

[77] S. Nair and co authors, "Deep learning closure of the navier–stokes equations for transition-continuum flows," *AIAA Journal*, 2023.

[78] G. Garg, T. K. Mankodi, E. Esmaeilifar, and R. S. Myong, "Neural network-based finite volume method and direct simulation monte carlo solutions of non-equilibrium shock flow guided by nonlinear coupled constitutive relations," *Physics of Fluids*, vol. 36, no. 10, p. 106113, 2024.

[79] G. Garg, T. K. Mankodi, and R. S. Myong, "Fast neural network-based direct simulation monte carlo solutions of shock flow of diatomic gases with vibrational modes," *Physics of Fluids*, vol. 37, no. 7, p. 076105, 2025.

[80] T. J. Baker, "Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation," *Engineering with Computers*, vol. 5, no. 3, p. 161, 1989.

[81] L. P. Chew, "Constrained Delaunay triangulations," in *Proceedings of the third annual symposium on Computational geometry*, (Waterloo,Ontario), p. 215, 1987.

[82] K. Reichert, J. Skoczylas, and T. Tarnhuvud, "Automatic mesh generation based on expert-system-methods," *IEEE Transactions on Magnetics*, vol. 27, no. 5, p. 4197, 2003.

[83] A. Papagiannopoulos, P. Clausen, and F. Avellan, "How to teach neural networks to mesh: Application on 2-D simplicial contours," *Neural Networks*, vol. 136, p. 152, 2021.

[84] T. Song, J. Wang, D. Xu, W. Wei, R. Han, F. Meng, Y. Li, and P. Xie, "Unsupervised machine learning for improved delaunay triangulation," *Journal of Marine Science and Engineering*, vol. 9, no. 12, p. 1398, 2021.

[85] P. Lu, N. Wang, X. Chang, L. Zhang, Y. Wu, and H. Zhang, "An ANN-based advancing double-front method for automatic isotropic triangle generation," *Scientific Reports*, vol. 12, no. 1, p. 13109, 2022.

[86] J. Anderson, *Fundamentals of Aerodynamics for Engineers*. McGraw-Hill Education, 2017.

[87] R. Myong, "Analytical solutions of shock structure thickness and asymmetry in navier–stokes/fourier framework," *AIAA journal*, vol. 52, no. 5, p. 1075, 2014.

[88] V. Casseau, *An Open-Source CFD Solver for Planetary Entry*. PhD thesis, University of Strathelyde, 2017.

[89] F. G. Blottner, M. Johnson, and M. Ellis, *Chemically Reacting Viscous Flow Program for Multi-component Gas Mixtures*. Tech. Report, Sandia Labs., Albuquerque, N. Mex., 1970.

[90] G. V. Candler and I. Nompelis, "Computational fluid dynamics for atmospheric entry," *Journal of Spacecraft and Rockets*, vol. 46, no. 3, p. 481, 2009.

[91] W. G. Vincenti, C. H. Kruger Jr, and T. Teichmann, *Introduction to Physical Gas dynamics*. American Institute of Physics, 1966.

[92] R. N. Gupta, J. M. Yos, R. A. Thompson, and K. P. Lee, "A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculations to 30000 K," NASA Reference Publication NASA-RP-1232, NASA Langley Research Center, 1990.

[93] G. E. Palmer and M. J. Wright, "Comparison of methods to compute high-temperature gas viscosity," *Journal of Thermophysics and Heat Transfer*, vol. 17, no. 2, p. 232, 2003.

[94] C. R. Wilke, "A viscosity equation for gas mixtures," *Journal of Chemical physics*, vol. 18, no. 4, p. 517, 1950.

[95] B. Armaly and K. Sutton, "Viscosity of multicomponent partially ionized gas mixtures," in *15th Thermophysics Conference*, (Snowmass, Colorado), p. 1495, 1980.

[96] B. F. Armaly and K. Sutton, "Thermal conductivity of partially ionized gas mixtures," in *16th Thermophysics Conference*, (Palo Alto, California), p. 1174, 1981.

[97] U. Ghia, K. Ghia, and C. Shin, "High-Re solutions for incompressible flow using the Navier- Stokes equations and a multigrid method," *Journal of Computational Physics*, vol. 48, no. 3, p. 387, 1982.

[98] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.

[99] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[100] M. A. Nielsen, *Neural networks and deep learning*, vol. 25. Determination Press, San Francisco, 2015.

[101] M. Wright, M. Loomis, and P. Papadopoulos, "Aerothermal analysis of the project Fire II afterbody flow," *Journal of Thermophysics and Heat Transfer*, vol. 17, no. 2, p. 240, 2003.

[102] V. Casseau, T. J. Scanlon, and R. E. Brown, "Development of a two-temperature open-source CFD model for hypersonic reacting flows," in *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, (Glasgow), p. 3637, 2015.

[103] C. Park, "Review of chemical-kinetic problems of future NASA missions. I-Earth entries," *Journal of Thermophysics and Heat transfer*, vol. 7, no. 3, p. 385, 1993.

[104] R. N. S. Prakash, S. Raghav, T. K. Mankodi, and N. Sahoo, "Fast prediction of chemically reactive rarefied hypersonic flows using boundary condition-based machine learning algorithm," *Physics of Fluids*, vol. 38, no. 2, p. 026103, 2026.

[105] R. N. S. Prakash, T. K. Mankodi, and N. Sahoo, "Boundary condition-based machine learning algorithm for fast prediction of supersonic flows," *Physics of Fluids*, vol. 37, no. 8, p. 086107, 2025.

List of Publications

Journal(s)

[1] Rachakonda Naga Sai Prakash, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Boundary Condition-based Machine Learning Algorithm for Fast Prediction of Supersonic Flows*, Physics of Fluids, Vol. 37, No. 8, Article 086107, 2025.

[2] Rachakonda Naga Sai Prakash, Sumati Raghav, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Fast Prediction of Chemically Reactive Rarefied Hypersonic Flows Using Boundary Condition–Based Machine Learning Algorithm*, Physics of Fluids, Vol. 38, No. 2, Article 026103, 2026.

[3] Rachakonda Naga Sai Prakash, Nishant Sharma, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Physics-assisted Machine Learning Algorithm for Optimized Grid Generation for Capturing Shocks in Rarefied Hypersonic Conditions*, (Aerospace Science and Technology, Vol. 177, Part E, Article 112934, 2026).

[4] Rachakonda Naga Sai Prakash, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *A Boundary-Condition-Based Machine Learning Approach for Fast Prediction of Chemically Reactive Flows Over a Stardust Reentry Vehicle*, (Manuscript to be submitted to Aerospace Science and Technology (AST)).

[5] Rachakonda Naga Sai Prakash, Nikhil Dewangan, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Boundary Condition-based Machine-Learning-Driven Domain Decomposition for Efficient DSMC Simulations*, (Manuscript to be submitted to Aerospace Science and Technology (AST)).

Conference(s)

[1] Rachakonda Naga Sai Prakash, Sumati Raghav, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Boundary Condition-based Machine Learning Algorithm for Fast Prediction of Chemically Reactive Hypersonic Flows in Rarefied Atmosphere*, 35th International Symposium on Shock Waves, University of Queensland, Brisbane, Australia, 2025.(*Oral Presentation*)

[2] Rachakonda Naga Sai Prakash, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Boundary Condition-based Machine Learning Algorithm for Incompressible Viscous Flows*, Conference on Fluid Mechanics and Fluid Power (FMFP 2023), Springer Proceedings, pp. 285–295, 2023.(*Oral Presentation*)

[3] Rachakonda Naga Sai Prakash, Nishant Sharma, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Physics-assisted Machine Learning Algorithm for Optimized Grid Generation for Capturing Shocks in Rarefied Hypersonic Conditions*, 35th International Symposium on Shock Waves, University of Queensland, Brisbane, Australia, 2025. (*Poster Presentation*)

[4] Rachakonda Naga Sai Prakash, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Boundary Condition-based Machine Learning Algorithm for Fast Prediction of Supersonic Flows*, 1st International Conference on Applied AI and Scientific Machine Learning, IISc Bangalore, Bengaluru, India, 2024. (*Poster Presentation*)

[5] Rachakonda Naga Sai Prakash, Tapan Krishnakumar Mankodi, Niranjan Sahoo, *Machine Learning-based Riemann Flux Solver for Shock Tube Simulations*, 8th National Symposium on Shock Waves, IIT Kanpur, Kanpur, India, 2024. (*Poster Presentation*)