

**TOWARDS EFFICIENT COMPUTATION OF
RAREFIED FLOWS USING FIELD INVERSION
AND MACHINE LEARNING TECHNIQUES**

A THESIS

Submitted by

SAEEL SHRIVALLABH PAI

for the award of the degree

of

BACHELOR OF TECHNOLOGY



**DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS**

CHENNAI-600036

MAY 2019

I dedicate this work to my parents as well as all my friends in IIT Madras who have made my stay over the past four years memorable and worthwhile.

“How lucky I am to have something which makes saying goodbye so hard!”

THESIS CERTIFICATE

This is to certify that the thesis entitled “**TOWARDS EFFICIENT COMPUTATION OF RAREFIED FLOWS USING FIELD INVERSION AND MACHINE LEARNING TECHNIQUES**” submitted by **SAEEL SHRIVALLABH PAI** to the Indian Institute of Technology, Madras for the award of the degree of **Bachelor of Technology** is a bona fide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Balaji Srinivasan
Associate Professor
Department of Mechanical Engineering
Indian Institute of Technology Madras
Chennai – 600036.

Place: Chennai

Date: 10 May 2019

ACKNOWLEDGEMENTS

It gives me a sense of fulfillment to write few words to acknowledge people who have helped with this project work. First and foremost, I would like to convey my sincere gratitude to Dr. Balaji Srinivasan for being so patient and guiding me throughout the project, and allowing me to explore various avenues by giving me the opportunity of working on a variety of problems. I would also like to thank Gaurav Yadav for helping me at every stage of the project. I wouldn't have been able to do half the things I have done in this project without him.

ABSTRACT

Keywords: Rarefied Flows, Field Inversion and Machine Learning (FIML), Artificial Neural Networks, Standing acoustic shock, High Knudsen numbers.

The Navier – Stokes equation, which is widely used to solve for a variety of flows is based on the assumption of fluid continuum. Thus, when this basic requirement is violated as in the case of rarefied flows, the Navier – Stokes equation fails to predict accurate results. The molecular level techniques which are currently used to solve for rarefied flows are computationally very expensive. In this work, we use two Machine Learning algorithms to speed up computations for these high Knudsen Number flows using available data. The first technique, called Field Inversion and Machine Learning (FIML), is used to find a spatial distribution of the functional correction to map the base model solution to experimental observations. After successfully demonstrating the use of this technique on a scalar

ordinary differential equation, where the model equation has missing terms, this method is then extended to predict the structure of an acoustic shock. The second technique uses a Maximum Likelihood Estimation (MLE) approach to develop an artificial neural network to predict the flow of rarefied gas around a spherical body. The successful employment of these techniques to predict flows in the canonical cases considered demonstrates great potential for the use of Machine Learning algorithms to the study of more complex flows.

TABLE OF CONTENTS

ABSTRACT	ii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
CHAPTER 1	7
INTRODUCTION	7
1.1. INTRODUCTION	7
1.2. OBJECTIVE OF THE WORK	9
CHAPTER 2	10
LITERATURE REVIEW	10
2.1. BACKGROUND	10
2.2. PROBLEM DEFINITION AND APPROACH	12
CHAPTER 3	14
METHODS	14
3.1. FIELD INVERSION AND MACHINE LEARNING	14
3.2. FEEDFORWARD NEURAL NETWORK USING MAXIMUM LIKELIHOOD ESTIMATION	19
CHAPTER 4	24
RESULTS AND DISCUSSION	24

CHAPTER 5	42
SUMMARY AND CONCLUSIONS	42
REFERENCES	44

LIST OF FIGURES

Fig 1: A feedforward neural network with two hidden layers

Fig 2: Solutions of the base model compared to the mean of the true process

Fig 3: Posterior distribution of temperature (L); Variation of Bmap along the rod (R)

Fig 4: Comparison between the true temperatures and the temperatures found from the Navier-Stokes equation for $M=8$

Fig 5: Temperature plots from the UGKS data, the base model, and the improved model with Bmap for $M = 6$

Fig 6: Temperature plots from the UGKS data, the base model, and the improved model with Bmap for $M = 8$

Fig 7: Variation of constants $C1$, $C2$ and $C3$ with Knudsen Number

Fig 8: Variation of constants $K1$ and $K2$ with Knudsen Number

Fig 9: Comparison of the solution predicted by the Neural Network (top) with the observed true solution (below) for $Kn=0$

Fig 10: Comparison of the solution predicted by the Neural Network (top) with the observed true solution (below) for $Kn=1$

Fig 11: Correlation plots for the neural network that has been trained

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

The Navier-Stokes equation is commonly used to solve for various fluid flows. However, one of the principle assumptions underlying the use of this equation is that of fluid continuum. The results obtained from the Navier – Stokes equations, therefore, are accurate only when the Knudsen Number (Kn) of the fluid is very close to zero (<0.01). When the flow becomes rarefied and the number of fluid molecules per unit volume decreases, the distance between the molecules (or the mean free path (MFP)) also increases and thus the continuum assumption is no longer valid. Thus, we have to depend on other, better models for an accurate solution. Various models based on the fundamental Boltzmann equation give a good understanding of the flow physics, but since these models work at the molecular levels, computations become extremely expensive. While solving for very simple geometries in case of extremely rarefied fluid

may still be feasible, solving for flows in the transition regime (neither continues, nor very rarefied) becomes prohibitively expensive due to the huge number of molecules present. Thus, alternate models which give reasonable accurate answers while reducing computational expense are required. In this work, we try to do that through the use of Artificial Neural Networks. Two different techniques have been tried out – 1) Field Inversion and Machine Learning (FIML) which was introduced by Parish and Duraisamy in their paper titled “*A paradigm for data-driven predictive modeling using field inversion and machine learning, 2016*”, and 2) Maximum Likelihood Estimation. The FIML method is initially verified on a case of heat transfer through a cylindrical rod, and then applied to modelling the structure of an acoustic shock. On the other hand, the MLE technique is used to obtain a simple forward model for flow of a rarefied gas around a spherical body.

1.2. OBJECTIVE OF THE WORK

The objective of the work is to try of a few different statistical and machine learning algorithms to speed up computations for rarefied flows.

The first technique, Field Inversion and Machine Learning developed by Parish and Dumaisamy has been initially validated on the case of heat transfer in a cylindrical rod, and then has been successfully used to obtain the temperature distribution across a standing shock wave.

Next, a simple feedforward neural network has been developed to simulate the flow around a spherical body and it has been shown that the physics of the flow has been beautifully captured.

CHAPTER 2

LITERATURE REVIEW

2.1. BACKGROUND

Even with a tremendous increase in the computational power over the last few decades, simulations based on first principles for most practical still remains prohibitively expensive. As a result, we need to rely on coarse-models most of the times. However, derivation of these coarse-grain models or analytical solution often involves a lot of simplifying assumptions which then takes a toll on the accuracy and reliability of the solution. Thus, attempt at developing accurate solutions has been going on for a long time.

The rise and growth of Artificial Intelligence and Machine Learning algorithms over the past few years has opened up new doors towards research in this area of developing better models. Work on developing surrogate models has been going on for a long time. Data has come out to be the most valuable resource, as machine learning enhanced data driven modelling is now taking the center

stage. Neural Networks are widely used to develop surrogate models based either on historical or experimental data. A relatively new method called Physics Inspired Neural Networks (PINN), which does data-driven modelling while taking into consideration the conservation laws governing various physical phenomena has been gaining widespread popularity, especially after Prof. Maziar Raissi came out with two papers on this method earlier this year.

While carrying out fine-grain simulation for most practical cases is still not feasible, the computational power available today often allows us to carry out very accurate high-fidelity simulation of a few canonical cases. Concurrently, the experimental techniques have also developed to levels where pretty accurate information can be obtained at scales which are relevant to most practical cases. With this in mind, the technique of Field Inversion and Machine Learning (FIML) was introduced by Parish and Duraisamy in [1] in 2016. This method uses of Bayesian statistics to develop accurate forward models for physical phenomena by using

available data, as well as by taking advantage of any information that we have about the physical processes which actually take place. It involves developing a base model based on our limited understanding of the phenomenon, and then introducing a spatially and temporally varying stochastic correction term which is learned from existing data (usually of simple cases) using stochastic inverse methods.

2.2. PROBLEM DEFINITION AND APPROACH

While accurate solutions for several macro-scale laminar fluid mechanics phenomena can be obtained, the solution to turbulent flows and rarefied flows still remains an unsolved problem. Although scientists and mathematicians have come up with several models to simulate these phenomena, most accurate models are only valid for a few canonical cases only and cannot be used in most practical applications. In this work, we try to use two of the techniques mentioned above to target rarefied flows, more specifically the structure of a

standing acoustic shock and the slow flow of rarefied gas over a spherical body.

Kun Xu in [2] gives the procedure to obtain a solution for the velocity and temperature distribution across a standing shock using the Navier-Stokes-Fourier equations. On comparing this solution with the Unified Gas Kinetic Scheme (UGKS) solution (which can be regarded as a true solution), it can be seen that the Navier-Stokes solution deviates significantly from the UGKS solution. Thus, an attempt has been made at using this Navier-Stokes equation as a base model in the FIML approach, and developing a more accurate posterior distribution by using the UGKS solution as the ground truth in order to check the feasibility of the FIML method for such scenarios.

In the next case, we use a feedforward neural network to develop a surrogate model for the flow of rarefied gas around a spherical body and check its accuracy. The analytical solution for this particular case, given by Torrilhon in [3], with some Gaussian noise added to it, has been used as observed or experimental data.

CHAPTER 3

METHODS

3.1.FIELD INVERSION AND MACHINE LEARNING

3.1.1. MATHEMATICAL SETTING

Let us assume a physical system is governed by a set of non-linear equation (PDE or otherwise). Also, the truth-model if the system can be represented in the following manner:

$$R_T(\mathbf{Q}_T(\mathbf{x}, t)) = 0, \quad (1)$$

where the operator R_T contains the governing equation and the operator \mathbf{Q}_T contains the model variables.

Now, let us assume the physical phenomenon is modelled by the equation

$$R_m(\mathbf{Q}_m(\mathbf{x}, t), M) = 0, \quad (2)$$

where $R_m \neq R_T$ and \mathbf{Q}_m may also be different from \mathbf{Q}_T . The term M arises from our lack of understanding of the physical system. That is to say, if the exact value or form of M is inserted into the model equation, then the model equations can obtain very accurate values which satisfy the truth model. However, M is usually not determinable

from first principles, and thus, we resort to modelling M also as a function of the variables which we have considered in our model equation. Thus, the model equation can now be represented as

$$R(\mathbf{Q}, \mathbf{M}(\mathbf{Q})) = 0 \quad (3)$$

In the present approach, we have replaced this model equation with a stochastic system

$$R(\mathbf{Q}, \mathbf{M}(\mathbf{Q}, \beta(\omega))) = 0, \quad (4)$$

where β is a random function which is found from data-driven inversion/machine learning methods. Determining β is the main essence of this particular approach.

3.1.2. APPLICATION OF THE FIML METHOD

The challenge of creating the stochastic system lies in finding out a distribution function β . Model inaccuracies prevent us from directly finding the value of β . Hence, an inverse problem is posed to infer the distribution of β from the available data. Bayesian Inversion is used to obtain β in the form of functional corrections. The functional corrections are obtained by calculating β at every point in the computational domain. We start off by

assuming a value of β (at all points in the domain, so β might be vector) with a certain level on confidence in our assumption. This gives the prior probability distribution of β , given by $p(\beta)$. Next we collect observational data, which may come either from experiments, or from high fidelity simulations. This data \mathbf{d} is considered as the ground truth. Thus, given a value of β , there will be a certain probability that the dataset \mathbf{d} , may be reproduced by the model equation that has been considered. This probability is the likelihood, and is given by $p(\mathbf{d}|\beta)$. Given $p(\beta)$ and $p(\mathbf{d}|\beta)$, there exists some probability of β given \mathbf{d} . This is the posterior probability of β given \mathbf{d} , and is given by the Bayes theorem as

$$p(\beta|\mathbf{d}) = \frac{p(\mathbf{d}|\beta)p(\beta)}{\int p(\mathbf{d}|\beta)p(\beta)d\beta} \quad (5)$$

In principle, the above stochastic functions can follow any distribution. However, for the purpose of simplicity, we assume a Gaussian distribution for both, the prior as well as for the likelihood. With these assumptions, the problem of solving equation (5) reduces to finding the

maximum a posteriori (MAP) solution, which can be obtained by optimizing the following equation:

$$\beta_{map} = \arg \min 0.5[(\mathbf{d} - h(\beta))^T \mathbf{C}_m^{-1}(\mathbf{d} - h(\beta)) + (\beta - \beta_p)^T \mathbf{C}_\beta^{-1}(\beta - \beta_p)] \quad (6)$$

where \mathbf{C}_m and \mathbf{C}_β are observational and prior covariance matrices respectively. The observational covariance is obtained from the data \mathbf{d} , and the prior covariance is chosen using domain knowledge. This term being minimized is called the cost function J . Thus.

$$\beta_{map} = \operatorname{argmin} J \quad (7)$$

The next step is to find the posterior of the covariance. In the linear case, this is given as the inverse of the Hessian found at the β_{map} point. In the non-linear case, this turns out to be an approximation of the Hessian at the β_{map} point. Thus

$$\mathbf{C}_{\beta_{map}} = \mathbf{H}^{-1}|_{\beta_{map}} \quad (8)$$

A standard method of obtaining realizations of β involves the Cholesky Decomposition of the posterior covariance matrix. Therefore,

$$\mathbf{RR}^T = \mathbf{C}_{\beta_{map}}, \quad (9)$$

and then random samples of the posterior can then be drawn by

$$\beta = \beta_{map} + \mathbf{R}\mathbf{s}, \quad (10)$$

where \mathbf{s} is a vector, the components of which are independent standard normal variates.

This inversion gives us values for β which are varying in space as well as time (depending on the problem). If we put this value back in our model equation, we will get a fairly accurate value of the physical quantity, consistent with the observations.

While carrying out this procedure, choosing the prior covariance plays an important part. The choice of the prior covariance often influences the final β that we obtain. The chosen prior covariance can be taken to be good enough if the final value of β gives values of the

physical quantity under consideration within a range of \pm standard deviation of the observed physical quantity values. Thus, finding the right covariance is an iterative process.

If the inversions are performed over a large number of cases, problem specific inferences can be converted into general modelling knowledge via Supervised Machine Learning algorithms. These inferences can then be used to obtain a functional relationship of $\beta(\eta)$, where $\eta(Q)$ are the input parameters for the model, or in some cases, the physical quantity itself may be directly obtained from these inferences.

3.2.FEEDFORWARD NEURAL NETWORK USING MAXIMUM LIKELIHOOD EXTIMATION

3.2.1. BASIC STRUCTURE AND WORKING

Feedforward Neural Networks, also known as Multilayer Perceptron, are one of the most common architectures used in Machine Learning. These are mostly used for supervised machine learning algorithms, where we already know the output what our functions is supposed

to achieve. The underlying principle on which the feedforward network is based is the **Universal Approximation Theorem**, which states that a feedforward network with a single hidden layer and a finite number of neurons can approximate any continuous function to a prescribed level of accuracy. Higher the prescribed accuracy, more the number of neurons required. However, in most practical applications, instead of using a very large number of neurons in a single hidden layer, more than one hidden layer is used with significantly lesser number of neurons in each layer, thus reducing the total number of weights required.

Thus, the goal of a feedforward network is to approximate a function $y = f(x)$, where y is the value of the quantity we are trying to model, and x is a vector consisting of all the input parameters to the problem. When the network is properly trained, we would be able to get the correct value of y for the given conditions x . The basic structure of a Feedforward Neural Network with two hidden layers is shown below.

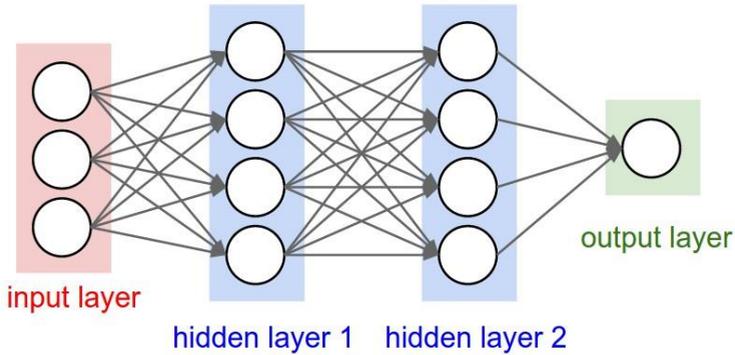


Figure 1: A feedforward neural network with two hidden layers

Each neuron in one layer receives input from all the neurons/elements from the previous layer in the form of a linear combination, with a non-linearity on top of it. I.e. $y_{neuron} = g(w \cdot x + b)$, where w is the weight vector, x is the input vector from the previous layer, and b is the bias term. The function $g(z)$ is a non-linear function which acts on z . The commonly ones include sigmoid, ReLU, Tanh, etc.

The network is initialized with a random set of weights. Every neural connection has its own set of weights. These weights are then learned during the network training process. This learning of weights happens through the popular Back Propagation algorithm. This

algorithm is basically nothing but optimization of the Cost Function using the chain rule of differentiation, and finding how the Cost Function J depends on each of the weights. Optimizing this Cost Function leads to the selection of weights which will then give the final output closest to what is expected. Once the network is satisfactorily trained, the network is then validated and tested on a different data set to see how good the learning really is. For detailed information on how the network actually functions and how it is trained, it is recommended to follow any standard book on Deep Learning.

Coming to the Cost Function, several functions such as the Mean Squared Error function, Cross Entropy function, Exponential function, etc. are commonly used. However, in the Maximum Likelihood Estimation approach that we have followed, the Mean Squared Error (MSE) function, which is shown below, has been used.

$$J(\mathbf{x}|W) = \frac{0.5 \sum_{j=1}^N (y_j - \hat{y}_j(\mathbf{x}, W))^2}{N}, \quad (11)$$

where N is the number of training examples used, \mathbf{x} is the vector consisting of input parameters, W is the weight matrix, y_j is the observational data and $\hat{y}_j(\mathbf{x}, W)$ is the value obtained from the network.

CHAPTER 4

RESULTS AND DISCUSSION

4.1. FIELD INEVRSION AND MACHINE LEARNING

4.1.1. HEAT TRANSFER IN A ROD

To validate the FIML method, we initially try it out on a scalar non-linear ordinary differential equation which resembles one-dimensional heat conduction with radiative and convective heat sources. This is the same case that has been considered by Parish and Duraisamy in [1], and it has been successfully replicated here. The ground truth is taken to be

$$\frac{\partial^2 T}{\partial z^2} = \varepsilon(T)(T_\infty^4(z) - T^4) + h(T - T_\infty), \quad z \in [0,1] \quad (12)$$

where ε is the emissivity of the material and h is the convective coefficient. For the true process, the emissivity is give as

$$\varepsilon(T) = \left[1 + 5 \sin\left(\frac{3\pi T}{200}\right) + e^{0.02T} + \mathcal{N}(0, 0.1^2) \right] \times 10^{-4} \quad (13)$$

The convective coefficient h is taken to be 0.5 SI units. To verify the efficiency of this framework, we shall assume that we do not know the true process, but only have the observational data (ground truth). Let us assume the process is modelled by the following equation

$$\frac{\partial^2 T}{\partial z^2} = \varepsilon_0 (T_\infty^4(z) - T^4), \quad (14)$$

where $\varepsilon_0 = 10^{-5}$. Let this be known as the base model.

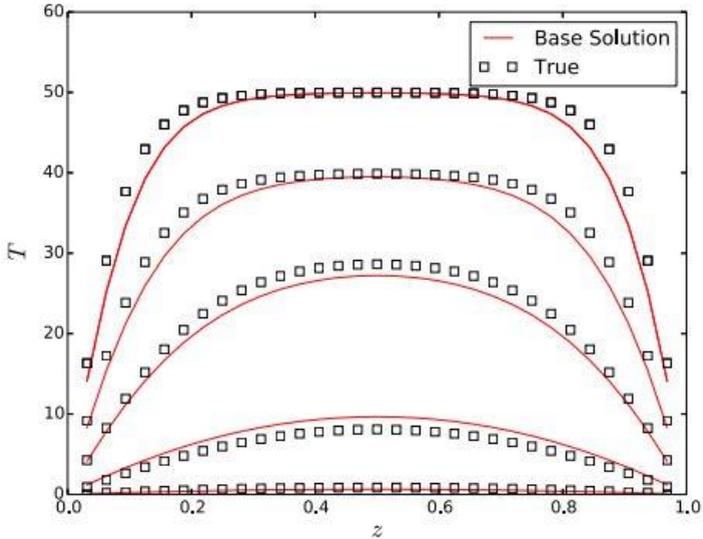


Figure 2: Solutions of the base model compared to the mean of the true process

From the figure above, it can be seen that the base model solutions do not accurately predict the actual temperatures in several cases. To improve this, we enhance the base model by introducing a correction function β in it. Therefore, the new equation is

$$\frac{\partial^2 T}{\partial z^2} = \beta(z) \varepsilon_0 (T_\infty^4(z) - T^4) \quad (15)$$

The goal of this method is to obtain $\beta(z)$ from inversion and then feed it to a neural network to generalize to get $\beta = \beta(T, T_\infty)$ or $\beta = \beta(\frac{x}{L}, T_\infty)$.

The entire process is summarised below.

1. Start with a base model. This may not represent the actual physics of the process, and is usually based on deficient understanding of the phenomenon.
2. Introduce a stochastic term β in the base model, and with an assumed value of β and prior covariance \mathbf{C}_β , find the value of β_{map} and the Hessian of the Cost Function at that point.
3. Check if the assumed value of the prior covariance matrix is reasonable by finding the value of the

- physical quantity (in this case, temperature) and seeing if the observed value of the physical quantity lies within $\pm 2\sigma$ of the one obtained from this model.
4. Sample the values of β using the equations mentioned in the section 3.1. and $\mathbf{C}_{\beta_{map}}$.
 5. Solve this for a number of cases with varying input parameters (T_∞ in this case), and generate data to train a neural network.
 6. Using appropriate machine learning algorithms, train a network to find β as a function of the input parameters.

Since the inversion is the trickier part of this method, we will be concentrating more on that rather than on the ML algorithms which can be used.

The inverse problem was solved for cases for T_∞ going from 5K to 50K in steps of 5K. Both the ends of the rods were kept at 0K. The temperature distribution as well the variation of β_{map} has been plotted for the case where $T_\infty = 50K$. It can be seen that the base temperatures are

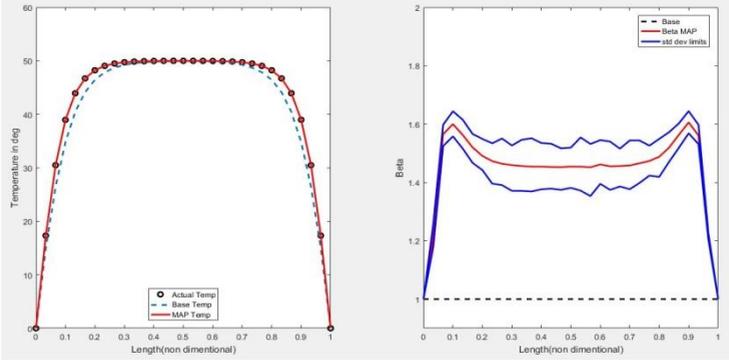


Figure 3: Posterior distribution of temperature (L); Variation of β_{map} along the rod (R)

not very good agreement with the ground truth. However, the temperatures obtained by using the β_{map} in the base model are in very good agreement with the observed values of temperature. Figure 3 (R) shows the variation of β_{map} along the length of the rod. As expected, the distribution is mostly symmetrical except for the stochastic variations. Since the base as well as the observed temperatures match at the rod ends (due to imposed boundary conditions), the β_{map} value at these places is zero, and this is what is observed from the β_{map} distribution. Similar results are obtained for cases with other T_{∞} too. Thus, it can be seen this this method works fairly well, at least in this simple case.

4.1.2. SHOCK STRUCTURE

After the successful implementation of the inversion framework on a heat transfer case, we next move to the studying the structure and temperature variation across a standing acoustic shock. The ground truth used here is the data obtained from the Unified Gas Kinetic Scheme (UGKS) simulations, for which Kun Xu's code [6] was used. The Navier – Stokes – Fourier equations is used as the base model. Thus, base model is

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix}_x = \begin{pmatrix} 0 \\ \frac{4}{3}\mu u_x \\ \frac{5}{4}\frac{\mu}{Pr}T_x + \frac{4}{3}\mu u u_x \end{pmatrix}_x \quad (16)$$

with the parameters $\gamma = 5/3$, $\mu \sim T^{0.8}$, $\mu_{-\infty} = 0.0005$ and the Prandtl Number $Pr = 1.0$ and $Pr = 2/3$. For a steady state solution, we put the time derivative = 0, and then integrate the above equations with respect to x . Thus, we get

$$\rho u = A \quad (17)$$

$$\rho u^2 + \frac{\rho}{2} T - \frac{4}{3} \mu u_x = B \quad (18)$$

$$\frac{1}{2} \rho \left(u^3 + \frac{5}{2} u T \right) - \frac{5}{4} \frac{\mu}{Pr} T_x - \frac{4}{3} \mu u u_x = C \quad (19)$$

where A, B and C are constants, and the following ODEs for the shock structure can be derived,

$$u_x = \frac{-3}{4\mu} \left[B - Au - \frac{AT}{2\mu} \right] \quad (20)$$

$$T_x = \frac{4Pr}{5\mu} \left[-\frac{Au^2}{2} + \frac{3}{4} AT - C + Bu \right] \quad (21)$$

where $\mu = \mu_{-\infty} \left(\frac{T}{T_{-\infty}} \right)^{0.8}$. The upstream and downstream

Rankine – Hugoniot shock conditions are

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_{-\infty} = \begin{pmatrix} 1.0 \\ 1.0 \\ 1/\gamma M^2 \end{pmatrix} \quad (22)$$

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_{\infty} = \begin{pmatrix} \frac{(1+\gamma)M^2}{2+(\gamma-1)M^2} \\ \frac{\gamma-1}{\gamma+1} + \frac{2}{(1+\gamma)M^2} \\ \left(\frac{2\gamma}{\gamma+1} M^2 - \frac{\gamma-1}{\gamma+1} \right) \frac{1}{\gamma M^2} \end{pmatrix} \quad (23)$$

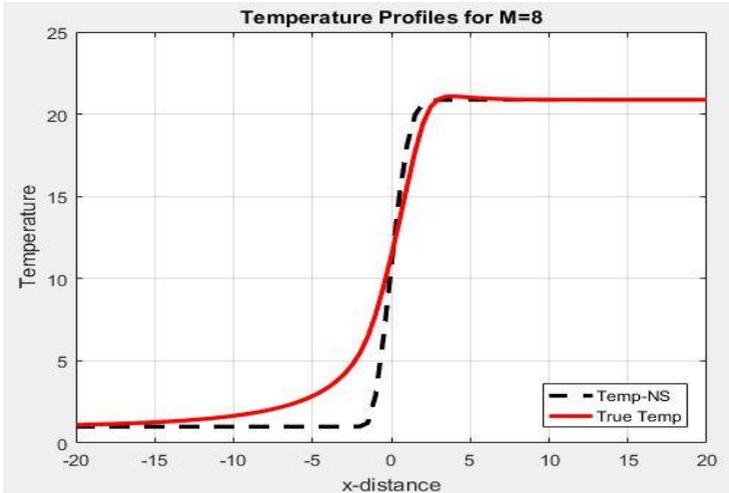


Figure 4: Comparison between the true temperatures and the temperatures found from the Navier-Stokes equation for $M=8$

The ODEs were solved in MATLAB using the ODE45 solver. Since the solver uses an adaptive grid, the solutions are pretty accurate. From the figure, we can see that there is a pretty large disagreement between the true temperature values and the solution obtained from the base model, especially in the region where the shock is formed. The boundary conditions, however, are the same for both the cases and these match well.

The model equation is modified by multiplying the viscosity term by a stochastic parameter β , which is a

functions of the Mach Number M . However, for sake of simplicity, this term is not introduced in the main equations but only in the ODEs obtained after the first integration with respect to x . Therefore, the modified ODEs are

$$u_x = \frac{-3}{4\mu\beta(M)} \left[B - Au - \frac{AT}{2\mu} \right] \quad (24)$$

$$T_x = \frac{4Pr}{5\mu\beta(M)} \left[-\frac{Au^2}{2} + \frac{3}{4}AT - C + Bu \right] \quad (25)$$

The adaptive grid which the ODE45 solver uses is one of the reasons why it gives the solution to such a high degree of accuracy. If a fixed grid is used instead, the solutions obtained are unrealistic, and even diverge for some of the cases. Since solving the inverse problem requires solutions at pre-specified points, it was accomplished by actually solving with an adaptive grid and then interpolating at the specified location using the MATLAB function `interp1`. Although the cost function turned out to be pretty computationally heavy and extremely slow to converge, the modified model with the

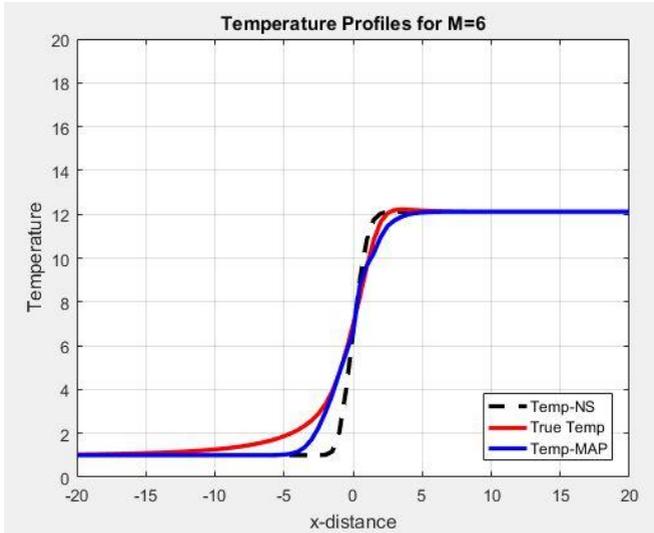


Figure 5: Temperature plots from the UGKS data, the base model, and the improved model with B_{map} for $M = 6$

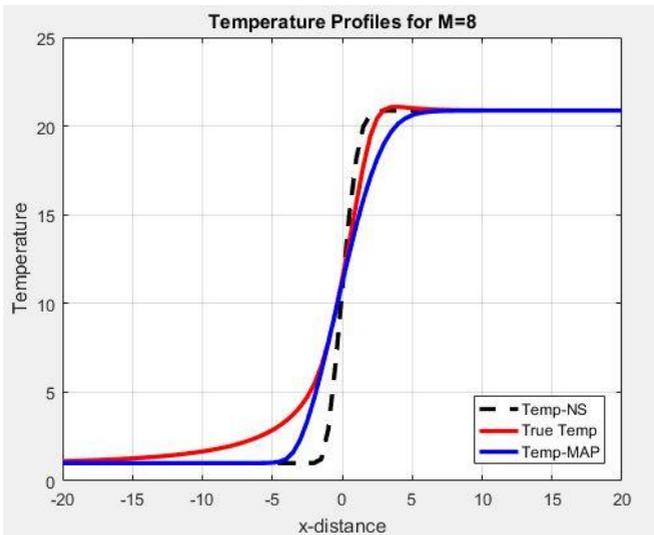


Figure 6: Temperature plots from the UGKS data, the base model, and the improved model with B_{map} for $M = 8$

β_{map} solution is an extremely fast way to solve and obtain the properties of the standing shock.

From the figures, it can be clearly seen that the temperature values obtained after the inversion process are significantly better than the base values obtained from the actual Navier – Stokes equations.

The introduction of just one term in a set of 3 equations brought about a significant improvement in the temperature profile. It is thus likely that an introduction of another such parameter, probably with the Prandtl Number might bring on more improvement and rectify the divergence from the true value which occurs at the curved portions.

4.2. MAXIMUM LIKELIHOOD ESTIMATION

4.2.1. SLOW RAREFIED FLOW PAST A SPHERICAL BODY

Although slow flow of rarefied gas over most over geometries does not have an analytical solution, the slow flow of rarefied gas over a spherical body has a more or

less analytical solution which is obtained from the 13-Moments (R13) method. As shown by Torrilhon in [3], using the right slip boundary conditions, the velocity of rarefied flow around a sphere, in spherical coordinates is given by

$$v(r, \theta) = \begin{pmatrix} \left(1 + a\left(\frac{r}{R}\right)\right) \sin \theta \\ -\left(1 + b\left(\frac{r}{R}\right)\right) \cos \theta \end{pmatrix}, \quad (26)$$

where

$$a(x) = \frac{C1}{2x} + \frac{C2}{3x^3} - K1 \left(\frac{6Kn^3}{5x^3} + \frac{2Kn^2}{5x^2} \right) e^{-\sqrt{\frac{5(x-1)}{9Kn}}} \quad (27)$$

$$b(x) = \frac{C1}{4x} - \frac{C2}{6x^3} + K1 \left(\frac{3Kn^3}{5x^3} + \frac{Kn^2}{\sqrt{5}x^2} + \frac{Kn}{6x} \right) e^{-\sqrt{\frac{5(x-1)}{9Kn}}} \quad (28)$$

$$\text{And } x = r/R \quad (29)$$

The flow is axisymmetric, and hence the \emptyset component of velocity has been dropped out.

$C1$, $C2$ and $K1$ are constants of integration. The variation of these constant as a function of Kn is shown the figure below.

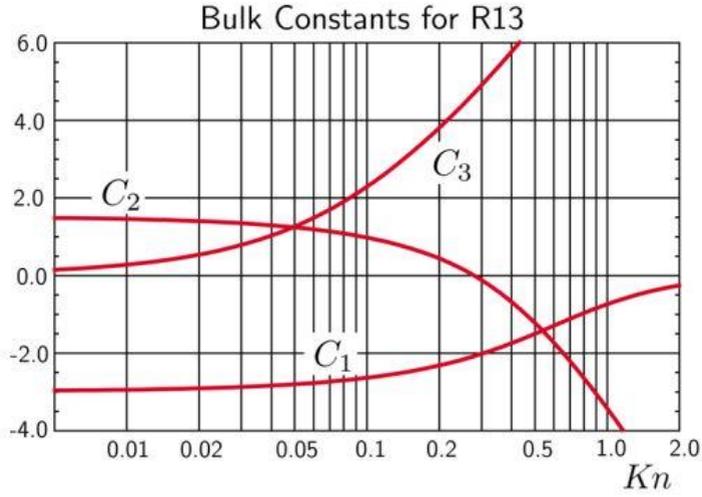


Figure 5: Variation of constants C_1 , C_2 and C_3 with Knudsen Number

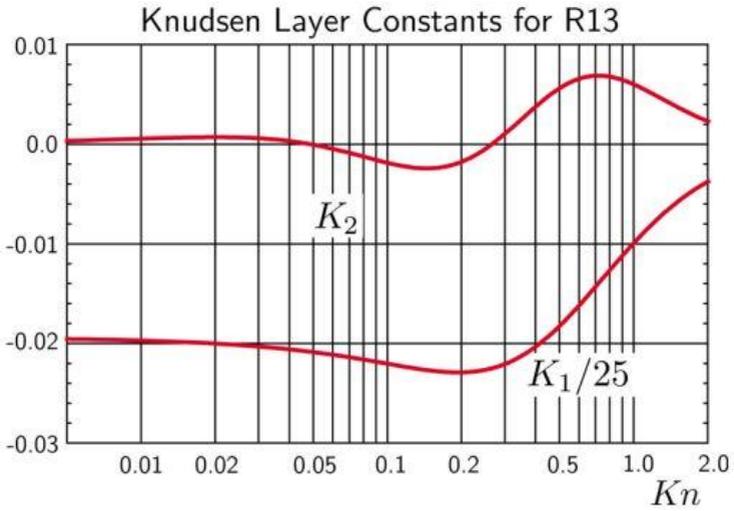


Figure 6: Variation of constants K_1 and K_2 with Knudsen Number

The availability of an analytical solution made this the perfect case to try out because ground truth data can be easily generated. The ground truth or training data has been generated for a range of Knudsen Numbers, ranging right from 0 to 2, by obtaining the analytical solution at the predefined computational grid points, and then adding a 5% Gaussian Noise to the data. This is done in order to simulate the actual uncertainty present in actual measurements. Also, in order to keep the case experimentally realistic, the data has been generated only for 11 Knudsen Number.

It has been assumed that for a slow flow, the velocity at all points in the domain scales linearly with the free stream inlet velocity. Therefore, the solution has been obtained for velocity which has been non-dimensionalised with the free stream inlet velocity. The radial computational grid has also been non-dimensionalised with the radius of the sphere. This non-dimensionalizing allows for more generalization.

The training of the Neural Network was done using the MATLAB Neural Network Toolbox. A single hidden

layer with 40 neurons was found to be sufficient for producing pretty good results. In the first case, inputs to the network were the location on the grid point and the Knudsen Number, and the output was the magnitude of velocity. In the second case, with the same inputs, the output was the r and θ components of velocity. While the second case was expected to give better results, the first case where the output was the direct magnitude of the velocity was found to be giving better results. A computational grid of 90×100 points was used, and with 11 cases, the total number of examples in the basic training data set was 99,000. This set was split in a 70%, 15% and 15% into the training, validation and testing set.

Figure 9 compares the solution obtained from the neural network (top) with the observed true solution (below) for a Knudsen Number of 0. In other words, this case represents the flow of a continuous viscous liquid around a sphere. This the same solution that we would obtain if we use the Navier-Stokes equation to find the velocity field.

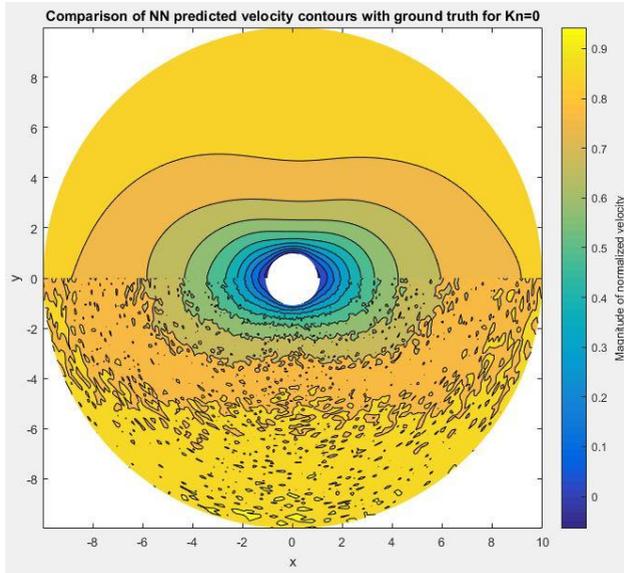


Figure 7: Comparison of the solution predicted by the Neural Network (top) with the observed true solution (below) for $Kn=0$

Figure 10 compares the solution obtained from the neural network (top) with the observed true solution (below) for a Knudsen Number of 1. A Knudsen Number of 1 signifies a pretty rarefied flow. The difference in the contours in figures 9 and 10 are easily visible. At lower Knudsen Number, the no slip condition at the solid boundary seems to dominate. However, as the fluids becomes more and more rarefied, the layers of fluid in contact with the solid surface start moving (relative

motion). This effect is beautifully captured by the Neural Network that has been trained. It is also seen that the overall flow field has been accurately captured by the neural network. This is also apparent from small value of Mean Squared Error obtained for all the three sets (i.e. Training, Validation and Testing). The correlation plots also contain most of the points near the 45 degree line, showing.

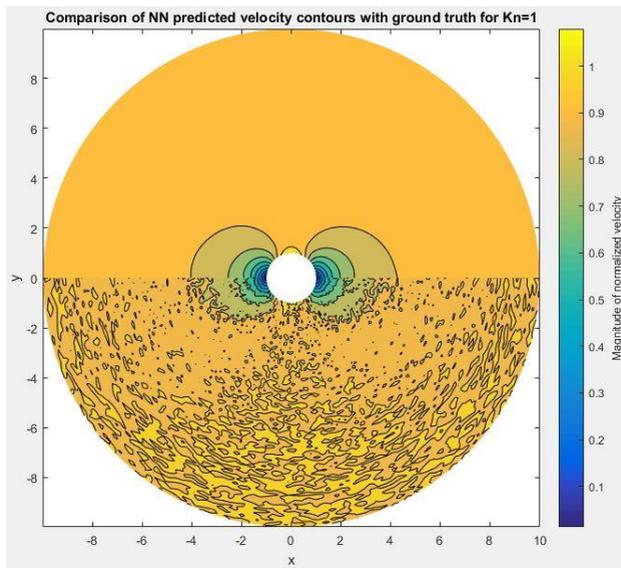


Figure 8: Comparison of the solution predicted by the Neural Network (top) with the observed true solution (below) for $Kn=1$

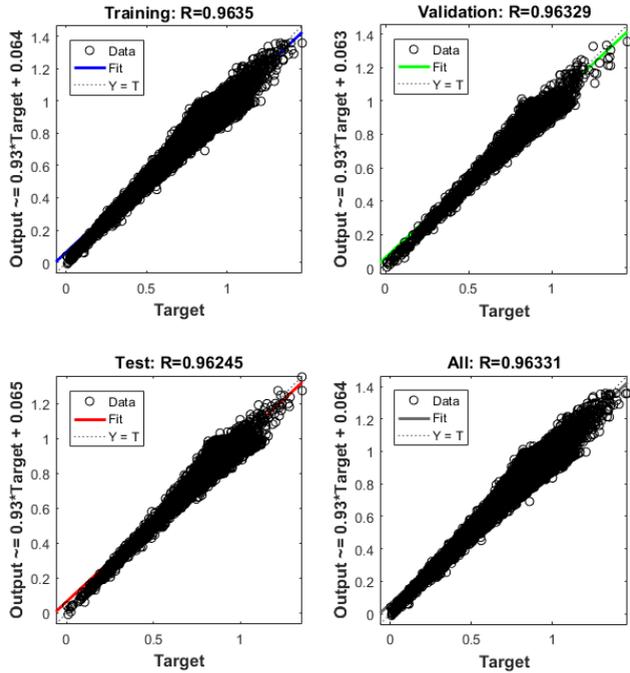


Figure 9: Correlation plots for the neural network that has been trained

CHAPTER 5

CONCLUSION AND FURTHER WORK

The feasibility of two methods, namely Field Inversion and Machine Learning, and Feedforward Neural Network (based a MLE) has been demonstrated by applying them on 2 different cases, the actual solutions to which were actually known. Although the work in this project concentrates primarily concentrates in inversion, further work on developing a suitable machine learning algorithm to learn and understand the physics of the problem will be extremely helpful, and is the ultimate goal of the FIML approach. With the addition of just a single stochastic parameter to a set of three equations, we were able to get a significant improvement in the output. Thus, there is reason to believe that the introduction of another such stochastic parameter combined with some other term (probably Pr) might give an even better result. This can be tried out in a further work.

In the case of flow around a spherical geometry, the network gives pretty good predictions about the velocity around the sphere. Further work on this front can involve mapping the flow around a sphere to flow around a different shape, either using Convolutional Neural Networks or by using mathematical transforms like the Joukowski Transforms, and then using these two maps (rarefied flow around sphere + flow around sphere to flow around a different geometry) to obtain the velocity field for rarefied flow around objects of more complex shapes.

REFERENCES

1. Eric J. Parish, Karthik Duraisamy, “A paradigm for data-driven predictive modeling using field inversion and machine learning”, *Journal of Computational Fluids* 305 (1016) 758-774
2. Kun Xu, “A Gas –Kinetic BGK Scheme for the Navier-Stokes Equations and Its Connection with Artificial Dissipation and Godunov Method”, *Journal of Computational Physics* 171, 289-335 (2001)
3. Manuel Torrilhon, “Slow gas microflow past a sphere: Analytical solution based on moment equations”, *Physics of Fluids* 22, 072001 (2010)
4. <http://cs231n.github.io/neural-networks-1/>
5. <https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620>
6. <https://www.math.ust.hk/~makxu/>