2018-01-01

Data-Driven Predictive Framework For Modeling Complex Multi-Physics Engineering Applications

Arturo Schiaffino Bustamante
University of Texas at El Paso, aschiaffino@miners.utep.edu

DATA-DRIVEN PREDICTIVE FRAMEWORK FOR MODELING COMPLEX MULTI-PHYSICS ENGINEERING APPLICATIONS

ARTURO SCHIAFFINO BUSTAMANTE

Master's Program in Mechanical Engineering

APPROVED:

Vinod Kumar, PhD.

Alejandro Garcia-Cuellar, PhD.

Heidi Taboada, PhD.

Arturo Bronson, PhD.

V.M.K Kotteda, PhD.

Sanjay Shantha-Kumar

Charles Ambler, Ph.D.
Dean of the Graduate School

DATA-DRIVEN PREDICTIVE FRAMEWORK FOR MODELING COMPLEX MULTI-PHYSICS ENGINEERING APPLICATIONS

by

ARTURO SCHIAFFINO BUSTAMANTE, B.S

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2018

ACKNOWLEDGMENTS

This work is dedicated to my mentors, Dr. Kumar, Dr. Garcia, Dr. Bronson, Dr. Taboada Dr. Kotteda, and Dr. Shanta-Kumar, who I would like to thank for all their professional development opportunities, patience and guidance during my graduate studies at The University of Texas at El Paso.I would like to thank my splendid research group, Ashesh, Arturo R., Luis, Diego, Zack, Jorge, Jose and Gaurav, for their constant effort, advise and support during this work. I couldn't wish for a better team.

My parents for supporting me during this struggle. My uncle Andres for his wisdom. My grandparents, Evy and Chucoy, who taught me the value of hard work and never giving up. To my siblings, Juanca, Yare and Eva. My little cousins, Edwin, and Alexa. We can do it guys. We just did it.

My most influential teachers, Lorena Rojo, Hiram Esparza and Nono Corral, who provided guidance and counseling during my early education and teenage years.

To the friends that grew up with me and have no words to thank for and show my appreciation, Bonny, Ramon Tapia, Alexis, Lary, Ramon Sida, Assahel, Julio, Vera, Daniel, Karen, Liber, Diana, Gabriel, Nancy, Roger, Juan Carlos, Nohe and all my classmates from CBTIS 117. ¡SI SE PUDO!

The friends that accompanied me through a wonderful college life, the Anguiano brothers, Quetzal, Menny, Dany, Reem, Lulu, Motta, Aaron, Jean, Nora, Jimmy, Gera, Nakul, Gabo, Vale Padilla, Efrain, Mario, Adri Dominguez, Roberto, Alex Berlanga and more importantly, everyone that I forgot to thank for their support.

ABSTRACT

Computational models are often encountered in multiple engineering application, such as structural design, material science, heat transfer and fluid dynamics. These simulations offer the engineers the capability of understanding complex physical situations before putting them to practice, either through experimentation or prototyping. The current advances in computational sciences, hardware architecture, software development and big data technology, have allowed the construction of sturdy predicting frameworks for analyzing a wide array of natural phenomena across different disciplines, either through the implementation of statistical methods, such as big data, and uncertainty quantification, or through high performance computing of a numerical model. The objective of this work is to study the implementation of a parallel, exa-scale pore network model based on Trilinos and Dakota, software packages developed by Sandia National Labs, along with machine learning techniques programmed by using TensorFlow, which are configured to predict complex multi-physics phenomena, such as flow through porous media or coolant explosions inside of industrial furnaces. Several predictions were made by using deep neural networks and uncertainty quantification of big data, proving that there is definitely a research window with big data in the Mechanical Engineering Sciences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

This chapter provides hindsight along with a brief about the topics covered by each chapter in this thesis.

Chapter 2 covers the background theory and previous research effort and contributions related to the concepts discussed in this thesis, such as governing equations, mathematical derivations, previous modeling approximation for flows through porous media, as well as important concepts for the development of this thesis, such as machine learning and big data.

Chapter 3 consists of a review of the software libraries and packages used to develop the computational tools explored in this thesis, such as TensforFlow a python-based language designed for machine learning applications, Trilinos, an exa-scale capable library for engineering and scientific calculations, Dakota, a tool for uncertainty quantification and sensitivity studies.

Chapter 4 deals with the research objectives for this work, which are the data driven, high-fidelity prediction of 3 multi-physics and multi-scale problems, which are the flow of molten metal through a packed ceramic bed, the prediction of molten metal spike length and number for coolant explosions, and the prediction of volume flow rate of oil for enhanced extraction.

Chapter 5 contains the methodology and assumptions made in order to generate the models that will predict the multi-physics problems, along with the problem description and motivation for creating such tools. The generation of the data sets containing the statistical representation of the physics, their processing and further implementations are being discussed in this chapter

Chapter 6 presents the results obtained once the models where implemented. Predictions for rate and depth of penetration for the flow of four transition metals into a boron carbide packed bed,

along with two sensibility studies are presented. Eight machine learning predictions obtained by using a deep neural network have been generated for Ti, Hf, Zr, and Y and are presented in this chapter. Also, two predictions for the interaction of coolant with molten metal is being presented. One tries to account for the number of spikes present in the space domain as a function of time, and the other one tries to predict the length that these spikes will achieve as time progresses. The last prediction presented deals with the volume flow rate for enhanced oil recovery applications.

Chapter 7 concludes the thesis and provides some remarks and explanations for the results, while Chapter 8 describes the future work and new research venues horizons that could be explored in order to improve the accuracy, fidelity and precision of the algorithms described in this thesis.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

Prediction modeling can be defined as a process that employs big data and statistics to provide with forecasts outcomes. Each model is composed of variables that have a weight, or a likeness to influence future results. These variables inside of the model are known as predictors. Once the system has been studied and a significant amount of data has been collected, a statistical model is formulated. This model may consist of linear equations, or a be formed of deep neural networks. As additional data is collected, the statistical analysis is validated and corroborated. With the sub sequential advances in high performance computing and machine learning techniques, along with the huge amount of data available nowadays, there exists a need to incorporate this data in order to attempt to predict the outcomes of an engineering problem.

This chapter is dedicated to the definitions that will be covered through this thesis, which is the prediction of the two-phase flow through a packed bed, and it's possible application in the engineering science, as well as the coolant explosions.

2.2 Ultra-High Temperature Ceramic

A matrix metal composite (MMC) is a material consisting of metal being reinforced by a mesh composed by ceramic or metal particles, in order to improve the properties of the material selected. MMCs materials are widely used in the automotive, aerospace and biomedical industry [1][2].

Ultra-high temperature ceramics (UHTC) are a type of MMC that are known to remain chemically and physically stable at high temperatures and in high reactive environments, which allows them to be used in extreme environments, such as hypersonic flights, atmospheric re-entry, rocket propulsion and high temperature components [3].

A UHTC is usually composed by a diboride compound. Special interest has been given to the diborides formed with transition metals from groups IV and V (Ti, Zr, Hf, Nb, Ta). From the previously mentioned metals, Hf and Zr were selected as the most prominent candidates for high temperature applications on future generations of reentry vehicles [4].

Several papers describe the experimental side of the molten metal infiltration into a porous media. [5] discusses the effects of the interfacial contact angle has on the infiltration process performed by them. The paper mentions that a wetting angle ($\theta < 90^0$) would allow liquid metal to better infiltrate a porous ceramic made of silicon carbide, however, it may cause degradation of the ceramic substrate due to reactive wetting. On the other side, a non-wetting contact angle ($\theta > 90^\circ$) allows for a non-reactive wetting, but it may be hard to fully impregnate the substrate ceramic. In another example, [6] reports a higher empty volume fraction than expected during the creation of a composite molten metal matrix, due in large part to the wettability effects of the molten metal (CuZnAl) against the silica gel used to create the composite material. Previous experimental results on the measurements of velocities and depth penetration of molten metals at high temperatures were found in the literature. Velocities for the flow of silicate glasses at 1200 $^\circ C$ over molybdenum are shown here [7]. Also, the spreading of molten Cu, Au and Ag over Mo has been documented here [8]. A measurement of the velocity of an aluminum alloys at 670 $^\circ C$ is provided [9] A broader description of the pore network modeling advances can be appreciated in [10]. Earlier attempts to model molten metal infiltration were taken by Medina et al. This numerical study consisted of analyzing the contribution of the viscous and inertial components to the molten metal flow into a single capillary tube in order to analyze depth penetration and velocity of the flow [11] . Another mathematical model of molten metal infiltration is being presented by Nishida et al. where he postulates the idea of using centrifugal forces as a mean of achieving the

threshold pressure necessary for the flow of the molten metal into a single pore [12]. On the pore media modeling aspect, Delgado et al presents a model to represent a porous media by using a stochastic representation of the pore distribution in order to predict the uncertainty of a porous media being invaded by another fluid [13] .

Properly understanding the molten metal infiltration will help to enhance the properties of the composites fabricated with this technique. Given the fact that the kinematics of liquid metal differs a lot from the organic flows, due to the high interfacial energies present in the molten metal[14][15]. This affects the rate at which molten metal spreads through a surface, which complicates the modeling of molten metal flow.

2.3 Multi-Scale Pore Network Flow Model

Two-phase flow through porous media is being studied in multiple disciplines and with different perspectives, in order to generate computational models, experimental studies and theoretical formulations [16]. A pore network approach, such as the one that is being suggested by this thesis, has been documented on several instances in the literature [10][17][18][19][20][21].

Several authors have worked on different strategies for understanding multiphase flow in porous media [20][22][23][24]. This interaction can be modeled by considering the relative viscosities and the wetting properties of the substances involved. In general, the flow in porous media can be explicitly expressed as a flow through a single pore at a micro scale and assembled together for a large number of pores to elucidate an aggregated flow through a porous structure. In this paper we consider the flow of a molten metal through a porous structure containing another fluid of lesser viscosity.

Regarding to two-phase pore network model by using artificial intelligence found in the literature, it was seen that the fields of geology and petroleum engineering have a special interest in investigating the flow through a porous media. It was found that different types of machine learning algorithms are being used to predict and classify the multi-scale permeability parameters that have an effect on the flow through a porous media, by using computer vision to achieve the image processing of rock and ceramic samples [25] [26][27] [28][29] bed as a result of the surface tension to viscosity dissipation.

2.4 Fluid Flow Development Through Pipes

The fluid velocity profile of any pipe ranges from zero velocity at the boundary layer due to the no slip-condition to a real number at the middle of the flow depending on the size of the pipe, as shown in Figure [30].

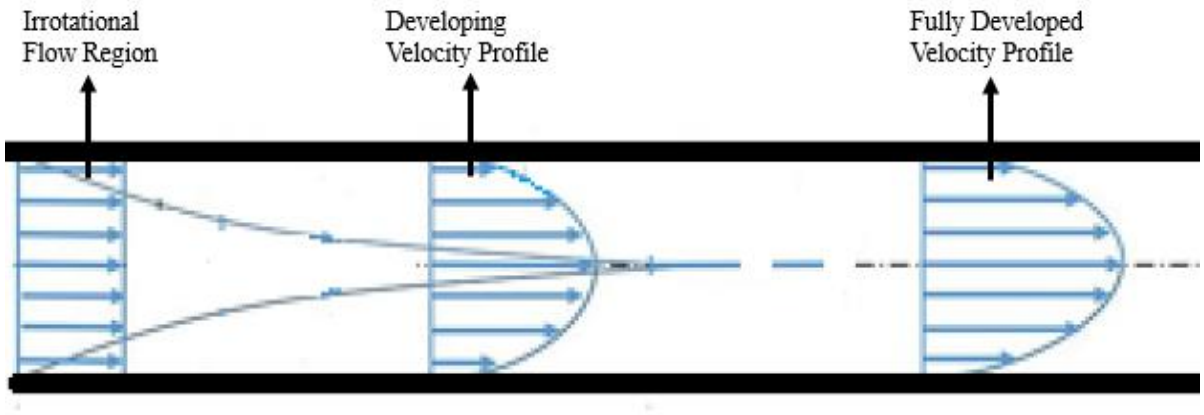

Figure 1: Laminar flow velocity profile development.

The velocity may change due to temperature change, for example; frictional heating. Although, frictional heat is insignificant that can be neglected, compared to other initial conditions that can also be applied. Flow is considered as Laminar for most of this thesis, because oil and molten metal are highly viscous, and given the fact that the significant lengths used in this thesis are in the microscale. All of the Reynold's numbers calculated in this thesis through Equation

$$Re = \frac{\rho V_{avg} D}{\mu} \quad (1)$$

2.5 Young Laplace Equation

A very important parameter when analyzing flow at the pore scale is known as the capillary pressure, which is created by the surface tension trying to keep the liquid together at the small-scale pores in sedimentary rocks and packed ceramic beds. The equation for calculating the capillary pressure is known as the Young-Laplace Equation:

$$p_c = \frac{2\sigma}{r}\cos\theta \quad (2)$$

Where the term p_c corresponds to the aforementioned capillary pressure, σ being the surface tension of the fluid, and θ representing the contact angle for the liquid-gas interface.

Furthermore, if the pressure of the fluid is higher than the capillary pressure, the fluid at the pore scale starts to move. For the fluid to move, there must be an overcoming pressure, which is the capillary pressure.

2.6 Hagen-Poiseuille Equation

The Hagen-Poiseuille formula is fundamental towards incompressible viscous pipe flow problems at any scale, but the numbers in it must be manipulated depending on the acting forces and energy transport mechanisms to function with the least amount of error [28]. The derivation is mathematically established from the Navier-Stokes equations. The equation involves a relationship between geometry (R), volumetric flow rate, pressure drop, and viscosity as follows

$$Q = \frac{\pi R^4}{8\mu L}\Delta P \quad (3)$$

This analytical formula provides a guide for analyzing laminar Newtonian fluids, such as; water, oil, gasoline, alcohol, and glycerin. This method has proven to analyze the insane amount of problems in the past since Poiseuille's law is so fundamental. The formula is mainly used to

calculate volumetric flow rate or pressure drop in the past. Hagen-Poiseuille formula demonstrates to work well for steady laminar flow pipelines.

2. 7 Hybrid Hagen Poiseuille Equation

A derivation of the Hagen-Poiseuille equation to calculate the volumetric volume flow rate of a 2-phase immiscible flow, such as the one described in Figure 2 can be found in here.

$$Q = \frac{\pi R^4}{8\mu_{eff} L} \Delta P \tag{4}$$

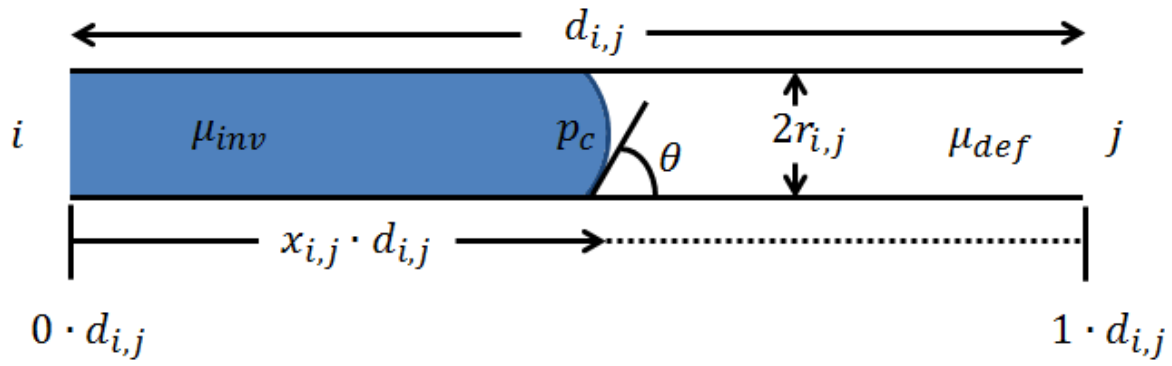

Figure 2: Multiphase laminar flow inside of a tube.

Another way of looking into this problem is to consider an "invading fluid" penetrating into a pipe completely filled with a "defending fluid". As inferred in Figure 2, the blue liquid is invading a pipe with a contact angle of θ. The variable x represents the length that the attacking fluid has penetrated into a pipe with length l and diameter d filled with the defending fluid, denoted by a white color.

In this equation, effective viscosity (μ_{eff}) is defined as:

$$\mu_{eff} = x\,\mu_{inv} + (1 - x)\,\mu_{def} \tag{5}$$

, for this equation, μ_{inv} quantifies the viscosity of the invading fluid, μ_{def} is the defending fluid's viscosity, and x represents the length invaded by the attacking fluid.

2.8 Semlak-Rhines Equation

The equation used to describe the flow through a pore is the Rhine's equation. This equation describes the transient rise of a flow meniscus through the capillary duct by the action of surface tension [31].

$$\frac{d}{dt}\left(\pi r^2 h \frac{dh}{dt}\right) = 2\pi R\sigma\cos(\theta) - 8\pi\mu h\frac{dh}{dt} \tag{6}$$

Past work by Kumar et al. [32] derived the Rhine's equation from including both viscous and surface tension forces.

$$h = \sqrt{\frac{R\ \sigma\cos(\theta)\ t}{2\mu}} \tag{7}$$

Equation 5 establishes a relationship between the depth of penetration, the time of the infiltration, contact angle, surface tension, capillary radius, and viscosity.

2.9 Uncertainty Quantification

Uncertainty quantification (UQ) is known as the method for scientifically quantify and characterize the uncertainty in computational and real-world applications in order to reduce it. Basically, it consists in an attempt to constrain how probable certain outcomes are if certain aspects of the observed phenomena are not fully known or understood [33].

There are a lot of problems in engineering and overall natural sciences that are filled with uncertainty. For this reason, computer modeling techniques are becoming useful in generating predictions in order to approach the uncertainty ruling over the engineering system.

2.10 Big Data

The term "Big Data" is used to describe enormous data sets composed by a wide array of intricate data structures stored by different means. The act of using the aforementioned data sets to generate predictions, correlations or classifications is known as big data analytics. Big data research is gaining importance in a multitude of scientific fields and several different businesses around the

world, since nowadays a significant amount of data is generated by virtually every person and every device connected to the Internet [34].

The defining feature of Big Data is its variety, since it comes from a great variety of sources and is classified in 3 types: structured, semi structured and unstructured. Structured data usually deals with well-organized and tagged data that is easily sorted, however, unstructured data is aleatory and presents a lot of challenges for its proper analysis. Semi structured data does not belong to a specific field, but contains tags to differentiate each element [35].

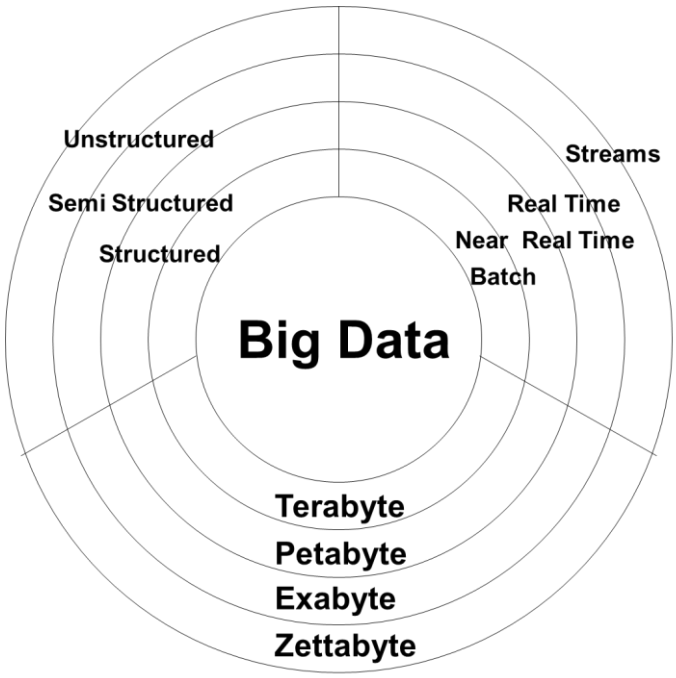

Figure 3: The defining features of big data

Currently, the size of the data being handled can larger than the terabyte and petabyte scale. The sheer volume of data being handled surpasses traditional store and analysis techniques.

Velocity is required for both, acquiring the big data and processing it into something useful. For processes with a finite amount of time, big data could be used as it streams into the computer in order to maximize its value.

Several uses of big data have been found to be used in the biomedical industry[36][37][38][39]. Regarding engineering applications that are based in the big data concept, examples were found in transportation systems engineering [40][41][42], supply chain management [43][44][44], manufacturing engineering [45] [46] [47][48]

2.11 Machine Learning

Machine learning is about extracting knowledge from data. It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning. The application of machine learning methods has in recent years become ubiquitous in everyday life.

Machine learning is about extracting knowledge from data. It is a research field at the intersection of statistics, artificial intelligence, and computer science and is also known as predictive analytics or statistical learning. The application of machine learning methods has in recent years become ubiquitous in everyday life.

In recent years, machine learning has driven advances in many different fields [49–54] This is due to the successful invention of more sophisticated machine learning models [55][56–58] , the availability of large datasets for tackling problems in these fields, and the development of software platforms that enable the easy use of large amounts of computational resources for training such models on these large datasets.

The most successful kinds of machine learning algorithms are those that automate decision-making processes by generalizing from known examples. In this setting, which is known as supervised learning, the user provides the algorithm with pairs of inputs and desired outputs, and the algorithm finds a way to produce the desired output given an input. The model can make predictions for new

data set which was never seen before. This type of models requires human effort to build the training set, but afterward automates and often speeds up an otherwise laborious or infeasible task.

There are two major types of supervised machine learning problems, called classification and regression. In classification, the objective is to predict a class label, which is a choice from a predefined list of possibilities. For regression tasks, the goal is to predict a continuous number. We can distinguish between classification and regression tasks based on the continuity in the output. If there is continuity between possible outcomes, then the problem is a regression problem.

On the other hand, in unsupervised algorithm, the input data is known and output data is given to the algorithm. These are successfully used in some applications. However, they are harder to understand and evaluate. For both of these learning algorithms, it is important to have a representation of input data. Each data point that we want to reason about is a row, and each property that describes that data point is a column. Each entity or row here is known as a sample or data point in machine learning, while the columns—the properties that describe these entities—are called features.

I. Multilinear Regression

For m samples and n features, the task of linear regression can be expressed as a task of finding vector w such that w^T x + b is as close as possible to y

$$[w_1 \quad w_2 \quad . \quad . \quad . \quad w_n] \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,m} \\ x_{2,1} & x_{2,2} & \dots & x_{2,m} \\ & & & \\ x_{n,1} & x_{n,2} & \dots & x_{n,m} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \\ b_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_m \end{bmatrix} \tag{8}$$

Where, x is input array, y is the output vector, w is the weight vector and b is the bias vector. The loss/cost function $J(w)$ is defined as the mean square error between wT x(i) + b and y(i) for $i \in [1, m]$

$$J(w) = \frac{1}{2m} \sum_{i=1,j=1}^{m,n} (w_i \, x_{i,j} + b_i - y_i)^2 . \quad (9)$$

We find w and b by minimizing the cost function using gradient descent optimizer. TensorFlow [59] uses the data flow graph to compute derivatives of the loss function with respect to every feature. It automatically adjusts them using the learning rate. Once the desired number of steps has been completed, we record the final values of vector w and scalar bias b.

II. Random Forest

RF algorithm have gained popularity in applications of machine learning during the last decade due to their good performance, scalability, and ease of use. In this model, average of multiple (deep) decision trees which individually suffer from high variance to predict the output. And, this algorithm is less sensitive to outliers in the dataset and don't require much parameter tuning. The trees are independent from each other and the algorithm makes different random choices for each tree. At each node of the tree, the algorithm randomly selects a subset of the features, and it looks for the best possible test involving one of these features. The number of features that are selected is controlled by the max_features parameter. This selection of a subset of features is repeated separately in each node, so that each node in a tree can make a decision using a different subset of the features. To make a prediction, the algorithm first makes a prediction for every tree in the forest. For regression, we can average these results to get our final prediction.

The random forest algorithm can be summarized in four simple steps:

1. Choose n samples randomly from the training set with replacement.

2. Make a decision tree from the bootstrap samples of size n.

At each node: a. Randomly select d features without replacement.

b. Split the node using the feature that provides the best split according to the objective function.

3. Repeat the steps 1-2 k times. Where k is the number of trees

4. Aggregate the prediction by each tree to get the final prediction.

Using the randomized feature selection process, we fit the tree based on the bootstrap sample {(X1 ,y1)…,(Xn,yn) generated from the training data.

III. Deep Neural Network

Deep learning is getting a lot of attention from many fields and it is the hottest topic in the machine learning field. The aim of the learning is to train artificial neural networks with many layers most efficiently. Neural network, in general, is a highly interconnected network of billions of neurons with trillion of interconnections between them. Artificial Neural Networks (ANN) are the biologically inspired simulations performed on the computer to perform certain specific tasks such as clustering, classification and regression. ANN resemble the human brain in two ways: acquire knowledge through learning and the knowledge is stored within inter-neuron connection strengths known as synaptic weights. ANN and more complex deep learning techniques are some of the most capable Artificial intelligence tools for solving very complex problems, and will continue to be developed and leveraged in the future.

ANN can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edges with weights are connections between neuron outputs and neuron inputs. This can be seen in Fig. 1. Hidden layers are in between input and output layers. These layers transform the input into something that output unit can use in some way. Each input is multiplied by its corresponding weights. Weights are the information used by the neural network to solve a problem. Typically, weight represents the strength of the interconnection between neurons inside the neural network. The weighted inputs are all summed up inside computing unit. This can be seen in Fig.2. An artificial neuron. The sum corresponds to any numerical value ranging from 0 to infinity). In order

to limit the response to arrive at desired value, the threshold value is set up. For this, the sum is passed through activation function. The activation function is set of the transfer function used to get desired output. There are linear as well as the non-linear activation function. Some of the commonly used activation function are—binary, sigmoidal (linear) and tan hyperbolic sigmoidal functions (nonlinear).

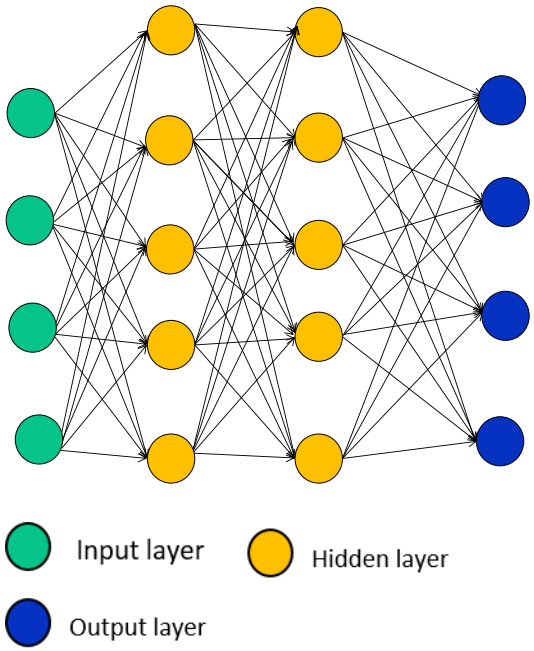

Figure 4 Diagram showing the mechanism behind a DNN algorithm.

The neural network learns by adjusting its weights and bias (threshold) iteratively to yield desired output. These are also called free parameters. For learning to take place, the neural network is trained first. The training is performed using defined set of rules also known as the learning algorithm.

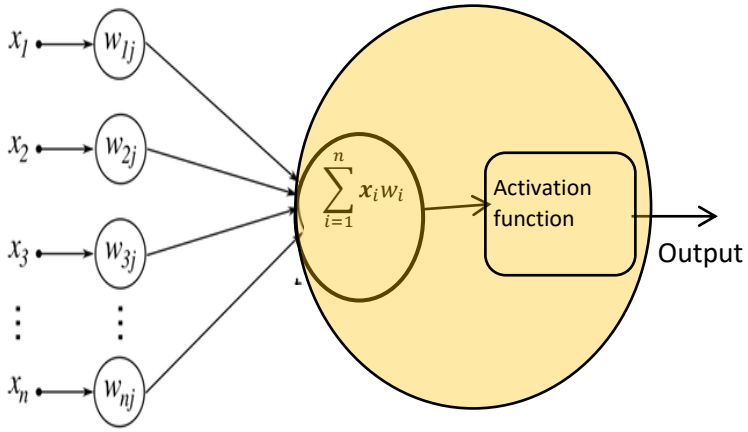

Figure 5: A graphical representation of an artificial neuron

2.12 Summary

In this chapter, a literature review has been performed, along with the revision of crucial concepts, such as a definition for ultra-high temperature ceramics, a discussion on multiscale pore network models and approaches, a brief discussion of the mathematical equations governing the flow, such as flow development, Hagen-Poiseuille equation describing laminar flow passing through a pipe along with its two-phase formulation, the hybrid Hagen Poiseuille equation. The Semlak-Rhine's equations are presented, which describe the capillary flow driven by surface tension through a small radius tube. Also, important techniques are defined, such as uncertainty quantification, a tool used to predict the stochastic uncertainty present in a system; big data, which consists of immense amounts of data and its possible uses, and machine learning, which consists in the utilization of big data in the preparation of artificial intelligence models.

CHAPTER 3: AN OVERVIEW OF THE ASSOCIATED SOFTWARE LIBRARIES

3.1 Introduction

In order to be able to generate predictions based on big data and stochastic procedures, one needs to have specific software packages with capabilities to process amount of data being offered, as well as to quantify the uncertainty associated with complex multi-physics systems. Commercial CFD software packages lack the ability of quantifying uncertainty or are just simply not optimized for high performance computing, a fact that points to the need of using libraries and software like Trilinos, TensorFlow and Dakota, which will be used to generate predictions for this work

3.2 TensorFlow

I. Introduction to TensorFlow

Machine learning is a computer or a processing machine that can learn from information given to it. Machine learning is design for many applications, for example, image processing, analysis of data, etc. Machine learning has the characteristic in laying in several different fields, such as artificial intelligence, statistics, computer science and some others [60]. For these simple reasons, TensorFlow machine learning is one of the most interesting tools used during these days, even for solving engineering problems. Machine learning has several libraries that can be used to extract models from it. Similarly, TensorFlow is based on models that can be implemented in Python. Therefore, the focus of the research paper will be to examine and discuss three models from [61]TensorFlow: deep neural network, random forest, and multi-variable regression.

TensorFlow is a second-generation system created by Google Brain that can run neural network algorithms at small portable processing systems, as well, large scalable processing units, such as; compositions of CPUs, GPUs, and hybrid systems [62,63] Its applications are many, such as; image processing, sound detection, engineering problems, etc. [64]. The TensorFlow mechanism works as any high-performance computing platform, where data is given to the machine as a form

of a matrix, then the platform must analyze it by mapping into small sections where it can parallelize in multiple processing units; known as High-Performance Computing Clusters. TensorFlow can be scalable through a range of processors, even though, having recognized that parallelism has its limits and cannot always be accomplished. TensorFlow second-generation has greater capabilities than its older version; DistBelief [65]ea. Such as flexible programming, better performance and supports training of data for a lot of neural network regression models, linear regression models, and non-linear regression models [66].

II. Composition of TensorFlow

TensorFlow is composed by a graph, which is guided by nodes. The main graph function is to sustain the data while the nodes keep it running [18]. The graph can be written in C++ or Python depending on the user preference. Each TensorFlow graph has nodes that can be any real number of inputs and outputs in any operation. The primary outcome of these activities is the creation of tensors, which are arrays that flow along the graph at discretize time. Control dependencies can coexist in the graph, and they refer to the time from when the system starts executing till it ends. Even though there can be some discrepancies with the implementation for specific factors, for example; crowing the memory.

There is real number of operations by several inputs and outputs in the graph of TensorFlow. There is some occurrence depending on the computation, for example; memory overflowing or operation execution. An operation can be defined as a simple computation, such as; simple scripts where you add or multiply a matrix. To accomplish these computations, there must be a kernel where most computations are executed in different processing units, such as; GPU and CPU. These computations can have an outcome of the different type of tensors. There must be a link between the kernels and computations. At any discretize time there must be a connection between the user

and the entire TensorFlow, called session. The session interconnects all the components, such as; the graphs and nodes, of TensorFlow implementation and Execution. Execution is when the user makes a call of the Run through TensorFlow. After a graph is executed, thousands of Run calls are also completed. Once the graph is implemented there some interesting facts, such as; the mutable tensors that survive the computation and create results. Depending on the operation, some tensors will survive predicting future data.

III. Supervised Algorithms in TensorFlow

Supervised algorithms seem to work the best based on its literature. These algorithms are based on training data including inputs and outputs, where the user gives the algorithm an enormous amount of data that must be analyzed, and the algorithm predicts future data. This approach task may seem laborious, but at the end is feasible and easy to adapt. All machine learning algorithms will depend on datasets. For machine learning applications the data must be normalized. The other requirement is that data must have partitions, which are: train, and test sets. Then the algorithm must have hyperparameters that hold constant during the procedure of the algorithm. There is also need of telling TensorFlow what can be or cannot be modified by initializing variables and placeholders. Then, there is also a need for defining the model that the graph is going to use to predict the values. The model will come the knowledge of knowing if the predicted values are correct compared to the expected values. The last thing is to validate if the model chosen is best for solving the given problem. All these statements can be found in Figure 6.

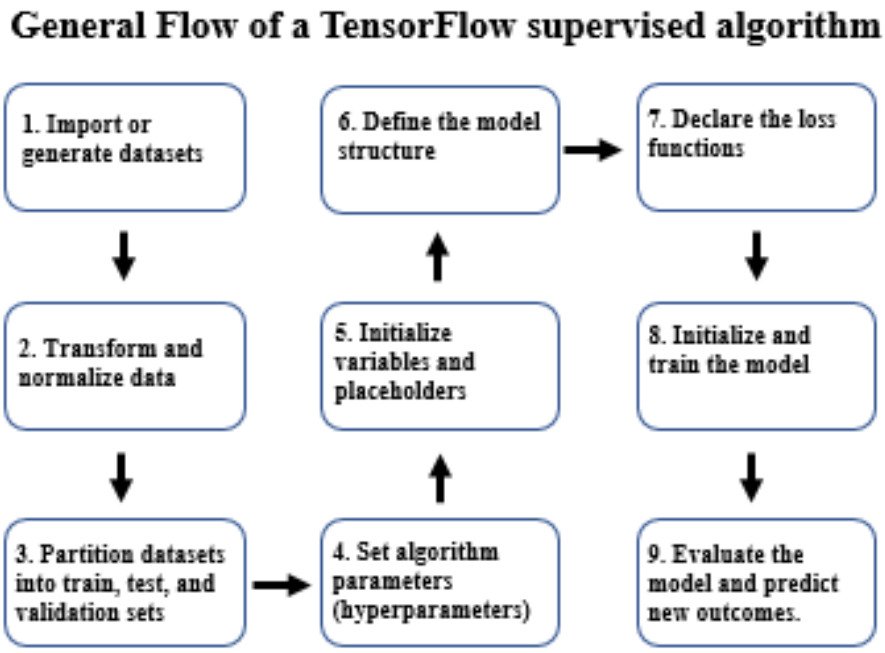

Figure 6: Exemplification of the general flow of a TensorFlow supervised algorithm

3.3 Trilinos: An Exa-Scale Capable Linear Algebra Library

Trilinos is a project developed by Sandia National Lab. It consists of a scalable linear algebra library that incorporates parallel sparse matrix classes, linear solver packages and preconditioner packages for high performance scientific computing. This library was created in order to provide a new object-oriented framework. Trilinos is composed of packages for linear and non-linear systems, Krylov methods, and FEM. Trilinos formulation allows for time domain and space domain discretization.

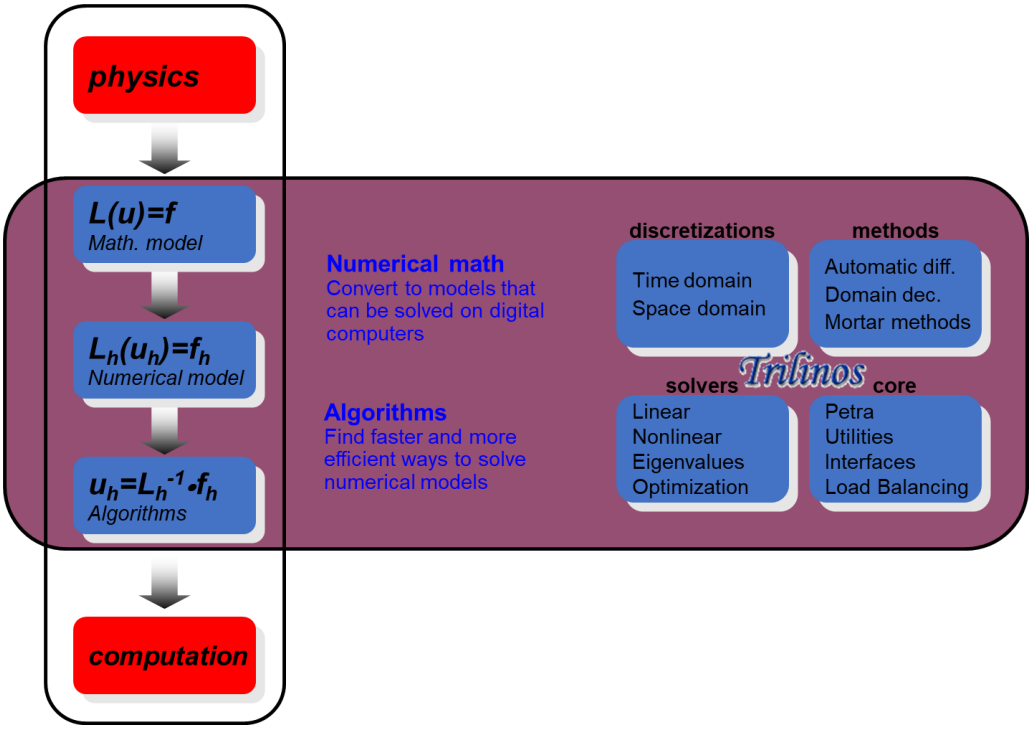

Figure 7: The design philosophy of Trilinos

Figure 7 depicts the philosophy of Trilinos. Everything starts in the physics domain, which is governed by certain mathematical formulation. That mathematical model is translated into a numerical model that can be interpreted by a digital computer. Once a numerical model has been established, it is necessary to implement algorithms that will generate a precise computation which describes the physical phenomena.

3.4 Dakota

The Dakota software delivers a strong, novel usable software for optimization and uncertainty quantification. Broadly, the Dakota software's advanced parametric analyses enable design exploration, model calibration, risk analysis, and quantification of margins and uncertainty with computational models. The Dakota toolkit provides a flexible, extensible interface between such simulation codes and its iterative systems analysis methods.

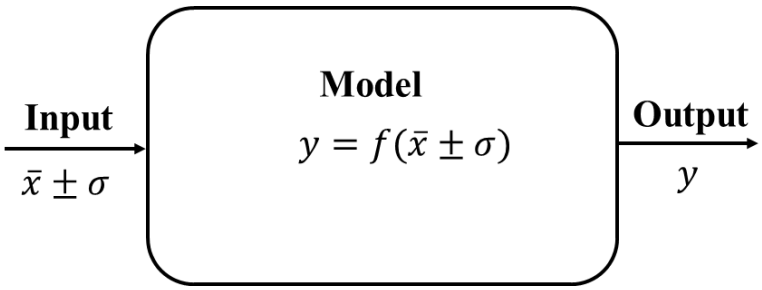

Figure 8: Philosophy behind Dakota

Figure 8 describes the concept behind Dakota. Certain sample input data is generated (in this case is being denoted by the variable x) through a normal distribution with a standard deviation of σ, which is then feed into a model $y = f(x \pm \sigma)$. After that, the output y is calculated for every value of x present in the domain. In other words, Dakota is a tool that calculates the variation of output y with respect to the change in value of input x. This is also known as a sensibility study.

3.5 Summary

In this chapter, two libraries and software package have been described. The first library of this chapter is TensforFlow, developed by Google Inc. TensorFlow is used to generate models based in machine learning and artificial intelligence that are based on big data. The second library analyzed is Trilinos. Trilinos is a library based in C++, and it is used to perform high performance computing simulations of complex scientific and engineering problems. The last software described consists of Dakota. Dakota has been developed by Sandia National Laboratories, and it provides with a framework to study the stochastic uncertainty quantification of computer models.

CHAPTER 4: RESEARCH OBJECTIVE, PROBLEM STATEMENT AND APPLICATIONS

4.1 Introduction

This thesis discusses the development of machine learning algorithms along with the use of exa-scale computing and big data in order to generate robust tools with the capability of predicting several multi-physics and multi-scale systems. Given the high degree of uncertainty that characterizes these systems, it is logical that such tools are required to handle huge amounts of data, and provide accurate predictions based on uncertain model parameters. The physical problems being analyzed in this chapter consists in the prediction of depth of penetration and rate of penetration of molten metal flowing inside of a packed bed with ceramic, with the purpose of creating ultra-high temperature ceramics for high speed vehicle applications, the possibility of predicting flow rate of oil being pushed through a porous underground reservoir, and the study of melt spikes in coolant explosions.

4.2 Problem Description Infiltration of Molten Metal

Infiltration of molten metal represents a formidable challenge to be accurately modeled. First, the structural randomness associated with porous mediums complicates prediction of the flow passing through it. Secondly, the properties of the molten metal could vary inside our control volume, since the temperature inside the control volume is not constant. In addition, there are several chemical reactions and solidification rates occurring in during the impregnation.

The objective of this thesis is to analyze the use of network inspired modeling strategies [11][18,19,67–72] for molten metal percolation through a packed bed as previously studied for liquids (e.g., crude oil) through porous rock in numerical and experimental fronts [71]. Because the flows in these systems can be viewed at different length scales, the amount of detail needed to predict the response of the system depends on four different length scales [73]. The first scale,

which is discernible only through scanning electron microscopy or thin sections of a pore at the microscopic scale. Fluid flow at the pore scale is explained by the Navier-Stokes equations. Except for only trivial cases, the equations cannot be solved as a result of the complex boundary conditions along the interfaces between the fluids and the solid matrix. The next scale is the core of rock, or macroscopic scale upon, which empirical correlations are developed from laboratory data using known fluid and porous rock properties to predict flow. Immediately following the core scale is the megascopic scale representing the entire crude oil reservoir and is modeled as a collection of thousands or millions of cores. The final scale is the gigascopic scale and is encountered in landscapes that may contain several reservoirs. In this paper, we have focused on a model for liquid metal flowing at the microscopic scale within a network of capillaries and pores.

In general, an analysis of a capillary-pore network followed with a subsequent assimilation and upward expansion to a macroscopic scale [74,75] becomes a computational challenge. Numerical simulations on traditional commercial packages have been performed [76], but to scale up such a capillary-pore model with a significant number pores would require optimized, parallel-computing techniques [77]. To simulate successfully a full-scale network, infusion physics for a million to billion pores need to be resolved which demands exa-scale capable programing models[78,79]. The proposed model EXPNS has been developed on a Trilinos[80] and supported through Kokkos[81] for performance portability across many-core architectures. The hybrid-parallel nature of EXPNS allows extensive scalability capabilities with optimized-data structures and communications through the Trilinos library. Moreover, the infusion physics has been studied at different temperatures which changes the viscosities of the competing fluids inside the porous structure, while accounting for the uncertainty in the viscosities. This uncertainty quantification has been performed on the penetration depth[82][83] through a massively parallel uncertainty

quantification tool Dakota[84]. The range of viscosities [85] contribute in the uncertainty in penetration depth and an extensive study has been performed in this paper along with the randomized geometry, both radii and length of the pores.

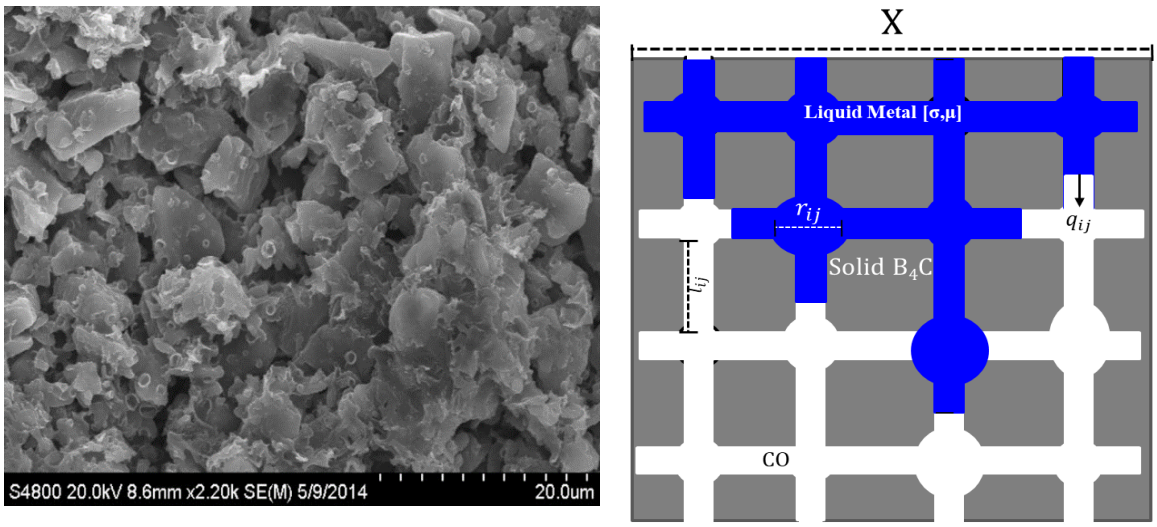

Figure 9: Pore network representation used for this thesis, both for the machine learning algorithms and EXPNS

Also, this work presents a model for the molten metal infiltration by using machine-learning algorithm, implemented through the TensorFlow library. This is a data-driven approach, judged to be properly given the high level of uncertainty present in this manufacturing process to predict the depth of penetration as a function of time. Being able to predict the behavior of the molten metal flowing through the porous ceramic will benefit the properties of the materials, by assisting in the understanding of the physics at the micro-scale level, therefore, preventing incomplete infiltration and other defects that could be originated during fabrication.

Figure 9 depicts the objectives of this research in a brief manner. The research objective can be summarized as obtain the capability of generating accurate predictions of the depth of penetration by using a data driven framework, obtained from a pore network with an assumed stochastic pore

radii distribution. By using this micro-scale approach, one can generate very accurate models of high resolution to understand the phenomena of two phased flow through a pore network.

The approach taken in the modeling of this problem is of a multiphase flow. We want to study the flow physics, especially the pressure distribution on the multiphase fluid domain, as the invading molten metal creeps through the porous structure displacing the fluid that was already existing in it. In order to do so, one may approach this multi-phase problem in various different ways, one common technique is to solve a coupled multiphase flow equation through a Galerkin finite element formulation and studying an over-all average flow physics of the invading fluid. This can be seen in various previous publications [86][87][88] and although is a resilient approach , is computationally challenging and can only solve porous structures with a small to decent number of pores. In this paper we harness the capabilities of an in-house pore network simulator (EXPNS)[89] that we had developed on an object-oriented next generation computing platform using Sandia National Labs' Trilinos and Kokkos , the structure of which would be described in brevity in the next section. The simulator models each micro scale pore of the network as a pipe and establishes axis-symmetric boundary conditions in order to simplify the problem. The simulator in general provides capabilities to introduce an elastic structure of the pore by modeling the coupled fluid-structure interaction as well but for this study we have considered each pore as a rigid body with a void through which the molten metal would flow and displace the existing fluid. The general flow equation of the fluid through a single pore is governed by the Navier-Stokes formulation.

$$\frac{\partial}{\partial t}(\rho u_t) + \frac{\partial}{\partial x_j}(u_i u_j) = \frac{\partial}{\partial x_j}\left(\frac{\mu \partial u_j}{\partial x_j}\right) - \frac{\partial p}{\partial x_i} \tag{10}$$

which is subsequently reduced and approximated through axis-symmetric assumptions along with other assumptions such as incompressible, and laminar to a form which looks like:

$$\frac{\mu \partial^2 u}{\partial r^2} = \frac{-\partial P}{\partial x} \tag{11}$$

The following can be solved exactly assuming a linear pressure difference, resulting in the following equation.

$$Q = \frac{\Delta p \pi r^4}{8 \mu x} \tag{12}$$

where x is the length of the pipe that the fluid occupies. This is in general a linear relationship between Δp and Q while the constant term, which a function of viscosity and length can be treated as a conductance, which is described in Equation 4.

$$K = \frac{\pi r^4}{8 \mu x} \tag{13}$$

In this case since a fluid already exists in the porous structure the invading molten metal, as it displaces the fluid inside, changes the effective conductance of the pore, which is in turn described by Equation 5

$$K_{eff} = \frac{\pi r^4}{8(\ \mu_1 x_{inv} + \mu_2 (l - x_{inv})} \tag{14}$$

where μ_1 and μ_2 are the invading and defending fluid's viscosities and x_1 and x_2 are the volume fractions respectively. From the effective conductance's expression, it is easy to notice that the value of conductance would change as the invading molten metal creeps through the pore and displaces the existing fluid, thus effectively changing the pressure drop across the pore. The problem has been modeled as a marching problem in time where each time step has been assumed to display a quasi-static behavior of the fluid interactions wherein the volume fraction of the invading fluid increases with each time step in a pore until it reaches the length of the pore. At this stage, we have a full control of the fluid dynamics at different time steps in a single micro pore; the challenge however lies in up-scaling from a micro pore to a macro scale network structure without losing out on the resolution that we have obtained. This capability comes with the EXPNS

that uses a robust data structure to assemble millions of such pores together by performing a mass balance of the total fluid flux at each node of the network. A glimpse of the up-scaling methodology and the data-structure used can be seen in Fig1[89]

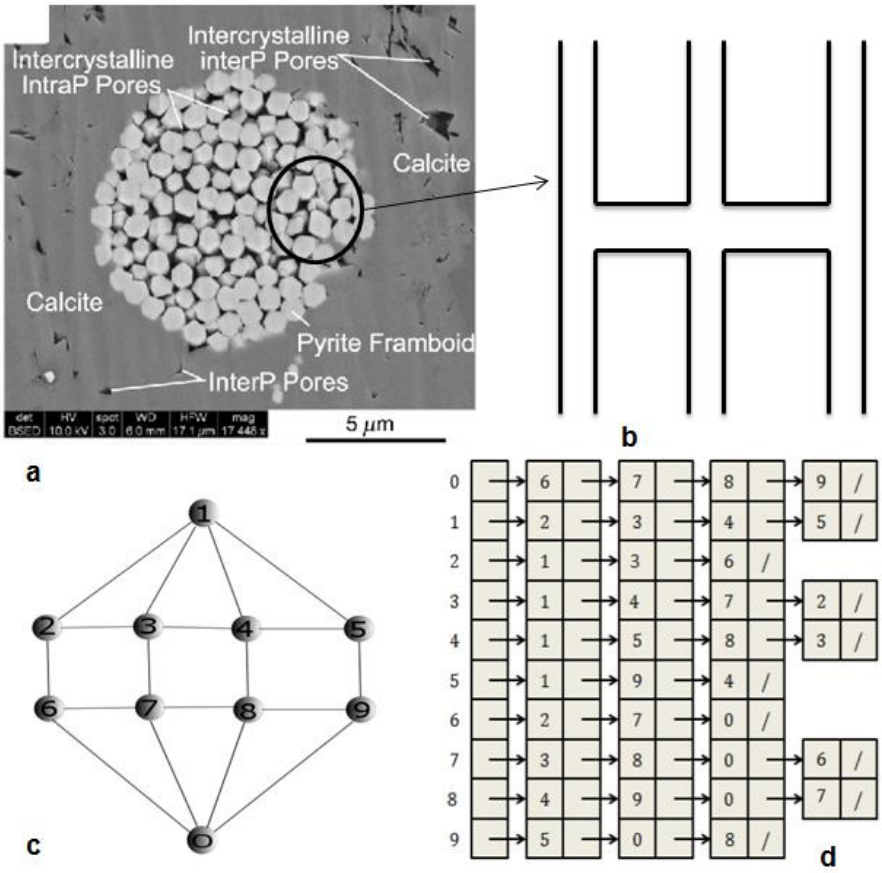

Figure 10:(a) An actual porous network (b) regularized geometry of the network model (c) graph based model of the pore network (d) adjacency list of the DAG [89]

EXPNS uses a DAG (Directed Acyclic Graph) to represent the porous structure and stores the DAG in template compressed row storage based sparse matrix classes provided by the Trilinos Library. Figure 10 (a) and (b) shows us how the simulator approximates the network structure. It also allows us to use a pseudo-random generator to build an irregular network structure by keeping the connectivity between each node as a 2-D regular mesh. Fig (c) and (d) shows us how the network is represented as a data structure and the adjacency list(d) implemented to store the

connectivity data. It allows us to use the data structures to build an assembled Saddle Point Matrix equation[90] which solves for all the pressure nodes on a defined pore network structure. The subsequent calculation for flux and volume fraction is intuitive but needs to be performed in an efficient and memory optimized fashion for post processing at each time step, each of which is provided by EXPNS. The exa-scale feature of EXPNS allows us to go up to sizes of 50M to 1B nodes and solve for a single degree of freedom while post processing for flux and volume fraction data at all time steps. The pressure distribution data obtained from running such simulations has been analyzed to gain some understanding of the flow physics in this thesis.

4.3 A Brief Description of the Algorithmic Features in EXPNS

This section discusses in brevity, the methodology of the computer implementation for the molten metal flow through the porous structure. The porous flow problem was defined as a molten metal invading the porous structure at a constant pressure of 10 units (scaled pressure) while the atmosphere is maintained at 0 units (scaled pressure). Each pore would have a no slip boundary condition, modeled as a circular pipe (EXPNS is capable of modeling any arbitrary pore structure though machine learning based predictive modeling techniques) and would exhibit laminar and quasi-static behavior at each time step. At $t = 0$ when a sudden pressurized molten metal is brought into the porous media, an immediate redistribution of the pressure values at each node would occur for the existing fluid which was at rest. This new redistribution leads to a multi-phase fluid flux through the porous media and increases the volume fraction of the invading molten metal. At $t = \Delta t$ the increase in volume fraction changes the K_{eff} value of the affected pores which leads to a further redistribution of pressure and change in multi-phase fluid flux. In this fashion the simulator marches the volume fraction in time for the invading molten metal as it slowly fills up the porous structure. A schematic of the core numerical algorithmic kernel inside EXPNS is shown in Fig 2 [89]. EXPNS has a hybrid-parallel structure(utilizes both shared memory

and distributed memory in conjunction) to exploit massive scale parallelism and has been tested to be scalable across 1024 processors while at the same time guaranteeing performance portability natively through Kokkos[81]. This allows to extract very high-resolution physics at pore scale while at the same time getting an overall picture of the pressure distributions at a macro scale.

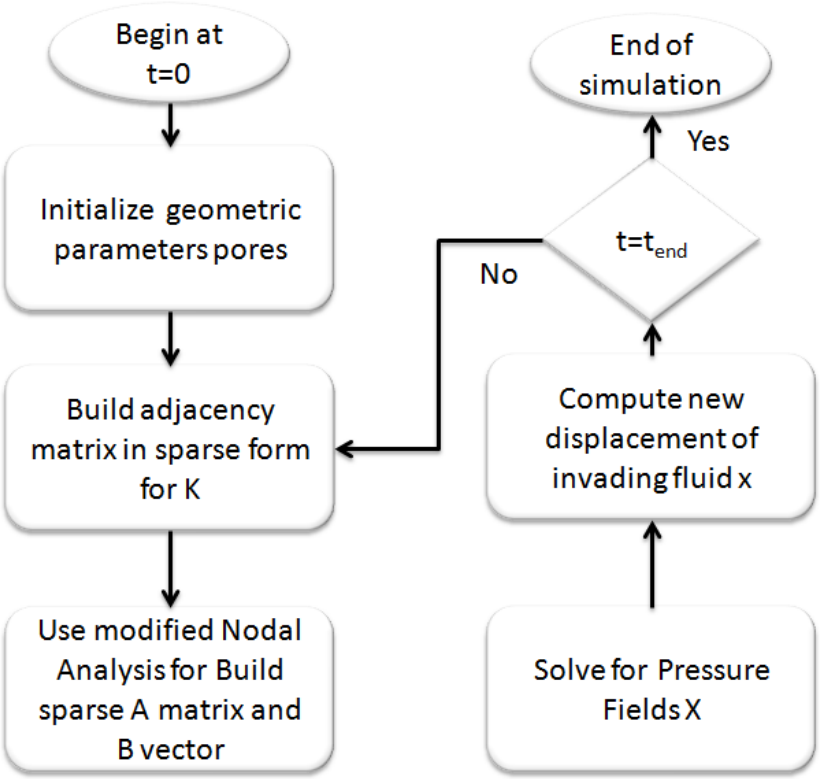

Figure 11: Schematic of the algorithmic kernel is EXPNS[89]

4.4 Problem Description of Melt Jets in Melt-Coolant Interactions

A fuel-coolant interaction (FCI), for this case melt-coolant interaction, is the fast evaporation of coolant, commonly water, due to the heat from the fuel. If the rate of steam generation due to FCI is high, then the rise in pressure cannot be relieved in time causing a shock wave in the mixture of fuel, steam, and water [91].

The contact of a coolant with a hot molten metal which is at a temperature is above the boiling temperature of the coolant can generate a vapor explosion [92]. Vapor explosions have been

presented in the metal industry, paper industry, contact of volcanic lava with water and in nuclear power plants.

Several experiments and simulations have been carried out to understand the preconditions of a vapor explosion. Kim and Corradini [91] studied the case of a molten fuel droplet inside a coolant pool and regarding their observations, they proposed a model where a film boiling occurs around a molten fuel droplet. After film collapses, formed coolant jets penetrate into the fuel droplet and get trapped. Rapid evaporation of the trapped water causes the fragmentation of the fuel. Ciccarelli and Frost [92] performed several experiments, in a particular one they placed a molten metal droplet inside water and observed the formation of melt spikes in the droplet surface. Regarding this, they suggested that local generation of high-pressure vapor at the droplet surface causes the formation of a wave and craters on the droplet surface, giving rise to the formation of melt jets (spikes). To prove this mechanism, they created a stratified system of melt and water with exploding wires just above melt surface to generate high-pressure vapor and obtain the formation of melt jets.

Simulations for molten metal droplets were made by Zhou et al. [93] and Zhong et al. [94]. They simulated the formation of melt jets (spikes) around a melt droplet that was surrounded initially by a vapor film.

Our simulation is inspired by the stratified water/liquid metal system made by Ciccarelli and Frost [92]. Volume of fluid (VOF) method is used to track the interfaces.

This work is dedicated to investigating how the melt jets are formed and to finding the effect of significant parameters like density and pressure in this process.

4.5 Introduction and Problem Description for molten metal infiltration

The purpose of this side of the study is to model the liquid flow into the capillaries of the packed bed by using machine learning algorithms from an open source available as TensorFlow library created by Google Brain. The library has a variety of algorithms including training and inference algorithms forming deep neural network models to predict the wetting dynamics, flow resistance, and the depth/rate of penetration into the capillaries of the packed bed. In the present work, the results from the machine-learning python code based on the TensorFlow library is compared against the experimental data obtained for molten Hf-Ti-Y-Zr alloys infiltrating into a packed bed of boron carbide at temperatures up to 2300°C. A summary of the techniques used to tweak the machine learning algorithms to predict the infusion behavior will be presented.

The first assumption made for this model is about the composition of the porous medium. In this case, it is composed of a packed bed of boron carbide (B_4C), with a pore radius distribution going from 1µm -10µm, based on experimental measurements conducted by Shanta-Kumar et al. [96]. Thus, it is supposed that a pore media can be simplified as a network of interconnected capillary tubes, with constant length and a Gaussian radii distribution.

4.6 Problem Description and Parameter Discussion

Table 1 is presented with real conditions in this paper for calculating some of the capillary pressure values in different rocks [110,111]

Table 1: Capillary Pressure for Different Rock

Name of the sedimentary reservoir rock	Tension between H_2O and oil [N/m]	Contact Angle (θ) [Degrees]	Diameter of the pipe[m]	Capillary Pressure (MPa)
Upper Jurassic Bossier Interval	0.05	20	$5.47x10^{-7}$	0.344
East Texas	0.05	20	$1x10^{-6}$	0.188
Basin Reservoir Rock	0.05	20	$9.40x10^{-8}$	1.99

4.7 Summary

This chapter has described with exhaustive detail the research statement. It mentions the two phase pore network model being developed with Trilinos for the study of molten metal, the uncertainty quantification that will need to evaluated with Dakota to account for the change of properties of the metal and irregularities in the microstructure, as well as defining the approach that will be used to estimate this depth of penetration by using TensorFlow and other possible applications of the machine learning pore network model, such as the possibility of quantify the oil extraction. Also, it details the process to study the phenomena of coolant explosions through machine learning, and numerical methods.

CHAPTER 5: METHODOLOGIES BEHIND TENSORFLOW ALGORITHMS AND EXPNS

5.1 Introduction

The machine learning algorithms generated with TensorFlow and the model EXPNS were created with the explicit objective of predicting multi-physics phenomena on a multi-scale coordinate system. In essence, the TensorFlow algorithm is trying to generate a prediction based on a statistical representation of the physics being studied, whether is flow through porous media or trying to predict the coolant explosions, while EXPNS is trying to generate a prediction by employing a formulation based on the hydraulic resistance of each individual pure that composes the microstructure. The methodology behind the development and application of the two approaches is covered in this chapter.

5.2 Sensitivity of Surface Tension-Viscosity in Simulating Molten Hf, Ti, Y, and Zr Infusion into a B4C Packed-Bed at the Microscopic Scale.

I. Background and Problem Statement

In considering the effect of the surface tension-viscosity dissipation driving the high-temperature liquid flow in a capillary tube, a high-fidelity computational model capable of quantifying surface tension and viscosity uncertainties was analyzed for liquid Hf liquid Hf, Ti, Y and Zr infusion into a B4C packed at 2300°C and at their melting temperatures. The model considers the flux balance within the capillary tubes of randomized size distributions between 1-10 μm assumed as an interconnected pipe network to mimic the porosity of a B4C packed bed. The rate and depth of infusion was validated with the Semlak-Rhines model for liquid Ti penetration into a packed bed. The viscosity of molten metal decreases with increasing temperature affecting the rate of penetration. Statistical uncertainty quantifications for the simulations were determined as a function of viscosity and boundary conditions under stochastic conditions. The computations were performed using exa-scale simulations developed with an in-house software, referred as Exa-Scale Poro-Elastic Network Simulator (EXPNS), which incorporates directed acyclic graph (DAG) and

modified nodal analysis algorithms. EXPNS was built on Trilinos and Kokkos libraries from Sandia National Laboratories (SNL).

A model of the dissipation extended the Semlak-Rhines (SK) model with consideration of a momentum balance incorporating the forces of surface tension, viscosity, gravity and end-drag to determine the rate of penetration into a packed bed assumed as a bundle of tubes mimicking its porosity. The modified model discerned that for liquid Ti, the surface tension dominated the capillary flow at times less than 1 μs with the viscous force having a synergistic effect at times > $10^{-4.5}$ s. Also, the penetrating rate reached a maximum of 7 m/s at approximately 1 μs, and the rate decreased with increasing time and agreed with the SK equation for times > 10 μs. Although surface tension is considered to drive capillary flow, the contribution of the viscous force increases and becomes significant for times > 50 μs.

In pore scale network modeling, the porous medium is considered as an idealized network of simple geometries representing a packed B_4C bed. While this idealization in general leads to loss of geometrical and topological information, simplification in the medium allows simulators to model flow behavior in larger domains with less computational effort compared to other approaches. Despite the simplification, however, coupling two-scales would require an enormous effort in times of computations, since flow needs to be resolved in both pore scale and network scale. Hence, a large number of pores typically within 10^4 to 10^8 needs to be modeled in order to successfully simulate the network scale.

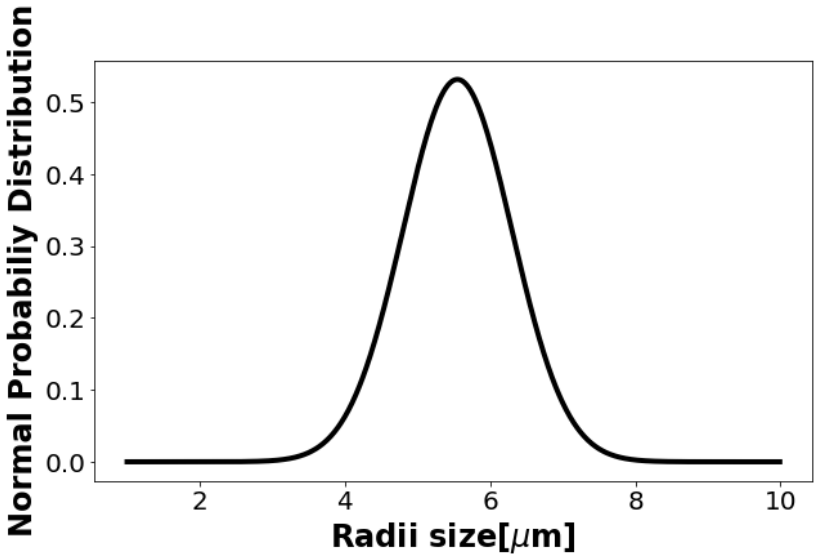

Figure 12 :Radius distribution used for EXPNS Code

In order to achieve this, a porous media with a normal radii distribution was generated with EXPNS. After that, 200 viscosity samples were prepared using a pre-assumed standard deviation.

II. EXPNS

For this purpose, we have developed EXPNS, an exa-scale pore-network simulator that is built on a scalable Trilinos and Kokkos backend to perform high fidelity simulations on a porous flow structure. In general, the application that we are interested in is the displacement of a liquid molten metal inside the porous structure which is filled with CO in its gaseous form with a very low viscosity ($\mu \ll 1$) making the system behave like a single-phase system. Figure 9 shows us the approximate geometry of the pore network as an interconnected network of tubes through which the liquid molten metal percolates. Here, we consider that the radii and the length of the pores are not constant and follows a probability distribution which forces cross-flow of fluid across the pores. In a real porous flow phenomenon for molten metal we would see some deposition along the sides of the pore which would hinder the flow process in while creating a meniscus at the moving front in. In general, a microstructure of a porous sample is imaged in and based on the

images a randomized pore network is generated in. The objective of this paper is to perform high resolution simulations as shown in previous examples, evidently the depth varies as we move along the breadth of the sample. If the breadth is L mm, then at any $\in (0, L)$, the penetration depth $y(x)$ varies with x. The average penetration depth is defined in Equation 15

$$\bar{y} = \int_0^L \frac{y(x)dx}{\int_0^L dx} \tag{15}$$

and the average penetration rate is defined in Equation 16.

$$\bar{y} = \int_0^L \frac{y(x)dx}{\int_0^L dx} \tag{16}$$

The partial derivative with respect to time implies that the penetration rate is a function of both time and space. In this thesis, a comprehensive analysis of how the penetration depth changes as time increases on an irregular sample with geometries generated from un uniform random probability distribution function for 4 different molten metals at two temperatures is presented.

III. Uncertainty Quantification

It is challenging to predict the behavior of the molten metal flowing through the ceramic porous media, since there is a high uncertainty in the properties of the metal at high temperatures and in the geometric. As previously mentioned, Dakota was used to perform uncertainty quantification for this model. Basically, the idea is to measure the variation of the output (penetration length and depth of penetration) with respect to the variation of the input parameters selected for this model, which are viscosity and radii variation.

5.3 Predicting the Depth of Penetration of Molten Metal into A Pore Network Using Tensorflow

I. Data Sets and Training

For this problem, eight datasets were created with the purpose of training the algorithms. The depth of penetration for the four-molten element will be predicted for their corresponding melting temperature and at 2300 °C.

Each training dataset is composed of 2,000 points. Equation 2 was used to provide a theoretical depth of penetration based of 4 constant inputs (μ, R, θ, σ), and one variable input (t).

For the training set, the time domain was considered to be a linear vector [0,45] seconds, with a step size of 0.0225 seconds.

The time domain used in the test set was the same, but the step size was set to 0.45 seconds.

Regarding the constant variables, several assumptions were made to generate the training set. Based on the powder packed bed pore microstructure, the value for R was set out to be of 10 micrometers. For problem simplification, there was assumed to be no contact angle, so θ was set to be equal to zero. The values of μ and σ are set constant at that specific infiltration temperature as shown in Table1 and Table 2.

Table 2: Physical Properties and Penetration Depth of the Liquid Elements at 2300°C

Element	T_m (°C)	Surface Tension (σ_{mp}) Nm^{-1}	Viscosity (μ_{mp} 10^3) $kgm^{-1}s^{-1}$ (Pa•s)	σ_{mp}/μ_{mp} ms^{-1}
Hafnium	2231	1.612	7.1	227
Titanium	1668	1.555	3.3	471
Yttrium	1522	0.804	4.54	177
Zirconium	1855	1.464	4.74	316

Table 3: Physical Properties and Penetration Depth of the Metals at their Melting Temperature (T_m)

Element	$\boldsymbol{T_m}$ (°C)	Surface Tension (σ_{mp}) Nm^{-1}	Viscosity (μ_{mp} 10^3) $kgm^{-1}s^{-1}$ (Pa•s)	σ_{mp}/μ_{mp} ms^{-1}
Hafnium	2231	1.612	7.1	227
Titanium	1668	1.555	3.3	471
Yttrium	1522	0.804	4.54	177
Zirconium	1855	1.464	4.74	316

Based on the aforementioned variables, it was decided that the algorithm would consists of 5 inputs and one output columns. It was decided to test the algorithm with 2 hidden layers and 5 neurons per layer. The training was set up to be composed of 5,000 steps. A relu6 activation function was selected to calculate the weight for each input variable.

5.4 Machine Learning Approach to Predict the Flow Rate for An Immiscible Two-Phase Flow at Pore Scale for Enhanced Oil Recovery Application.

I. Introduction and Background

Due to the demand of energy at the world scale, there is a need of extracting the most oil at the pore scale of wet reservoir sedimentary formations. The physical approach is presented by pressurizing water into the porous media for enhanced oil recovery applications; this is done by applying pressure into the wetting oil pore of the rock for displacing and extracting the oil. This two-phase flow will come with some complications due to the behavior of the fluid flow at the pore scale. In these types of applications, there is a need of considering the capillary quantities, such as; surface tension, viscosities, pressure drop, the radius of the medium and contact angle. In the present work, we use a machine learning algorithm in TensorFlow to predict the volumetric flow rate for a given pressure drop, surface tension, viscosity and geometry of the pores. TensorFlow library is developed at Google Brain and is one of the most recent tools created for machine learning applications. The Black Box approach to machine learning is to have a history of data that can be learned through use to predict future data. In this paper, the predicted values for

a two-phase flow of various pore sizes and liquids are validated against the numerical and experimental results in the literature. To consider a fluid as an immiscible fluid, there must be a difference of densities between the two fluids, like the solvation forces created by a fluid-fluid interaction repulsion [97][98][99]. Showing a heterogenous depiction of the flow, meaning that the two fluids are not soluble to each other [100][101]. For classifying classes of fluids, there's a domain that must be analyzed for ranges of problems. In the case of this paper multiphase flow is going to be discretized into sections for analysis of one fluid while examining the other as well for this oil displacement immiscible two-phase flow [102][103][104]. If this two-phase flow is assumed to have different properties for each fluid, for solving purposes, they must be a couple regarding the pressure. This coupling can be done through the combination of two equations; the Hagen-Poiseuille Equation and the Lucas-Washburn Equation [105]. Knowing that Newton's law describes that the summation of forces must equal mass times acceleration for dynamic particles [106]. In our case, there is a transferring effect from pressures to forces, since summation of forces is the ideal case for Newton's law. Even though, the summation of pressures is an ideal case for a manometer problem, which can also be used to solve this problem [30]. Therefore, the pressure that is going to be applied to H_2O is going to add to oil as an attacking pressure for the movement of oil. While for H_2O the oil will be the defending pressure for the displacement of H_2O[103]. This problem does not involve air as a third phase due to its low density.

The porous media problems have been modeled with different approaches used by scientists. During this modeling, the main physics scale has been clarified during the analysis of this type of systems in the literature review. The main scale proposed for the approach is to model at the pore scale with a geometry generated randomly for the unknown perspective of simulating different pores with different radiuses [103]. For solving the flow rate, the equation that is going to be used

is a hybrid combination of the Hagen-Poiseuille Equation, Lucas-Washburn Equation, and Young-Laplace Equation[105]. Which are the leading equations for internal laminar flows. There some viscous fingering instability effects incorporate inside of the porous media as the flow accumulates [107][108][109]. After obtaining the data from the hybrid equation, it must be trained.

II. Parameter Discussion

The next data would display the ranges for water displacement that were used to generate randomize data in excel before the coupling:

$$\Delta P_{i,j} = (10 - 100)\ \text{MPa} \tag{17}$$

$$R_{i,j} = (0.5 - 1.6) * 10^{-5}\ \text{m} \tag{18}$$

$$\mu_{eff} = (1 - 1.6) * 10^{-3}\text{kg/ms} \tag{19}$$

$$D_{i,j} = (1 - 10) * 10^{-2}\text{m} \tag{20}$$

The only thing that would change for the Oil & Water coupling is the effective viscosity.

5.5 Methodology behind computational modeling in analyzing coolant explosions

A 2D planar simulation in ANSYS Fluent software is executed. The 2D simulations can still capture the fragmentation phenomena with a less computational effort. Standard k-ε model is used and the results with the k-ε model are similar to the experimental results in a qualitative way. The transient flow solver is used to simulate the multi-phase flow. The equations governing the flow, (1-3), are conservation of mass, momentum, and energy.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_m \tag{21}$$

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \vec{v}) = -\nabla P + \nabla \cdot (\bar{\bar{\tau}}) + \rho \vec{g} + \vec{F} \tag{22}$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\vec{v}(\rho E + P)) = \nabla \cdot \left(k_{eff} \nabla T\right) + S_h \tag{23}$$

Molten metal (tin), steam, water, and air are used in the current simulation. Default properties for steam, water, and air are taken from the software meanwhile tin properties are defined by the user. The domain consists of a rectangular shape with three adiabatic walls, one pressure outlet and a stratified system of the mentioned fluids. The domain, as well as the boundary conditions, can be seen in Figure 13. No slip boundary condition is specified for walls.

The volume of fluid method (VOF) is used because it is simple and accurate, this method allows us to track fluid-fluid interfaces. The VOF method is considered since it has already been successfully used in earlier works [93–95].

Four phases were defined, one for each fluid. A geo-reconstruct solution scheme for volume fraction is used to track the interfaces. This scheme is recommended when the jet breakup is analyzed with VOF method. SIMPLEC scheme used for pressure-velocity coupling helps in achieving the convergent solution. The evaporation-condensation mechanism is used for the phase change from liquid water to water vapor. Continuum Surface Force model is used in the surface

tension model as well as wall adhesion. Surface tension values for the materials, except air which was used as zero, were obtained from [95]. Table 4 lists surface tension values. Pressure pulse in the vapor film is placed with a patch in the initial conditions, the pulse starts from 0 s until 50 µs. The computations are carried out with a time step of size 10 µs. Time independence study was performed for time steps of 20 µs, 15 µs, and 10 µs. The spike height for the flow at 0.4 ms is shown in Figure 17. The results with three different time steps are matching and the maximum difference in the spike height is less than 3%.

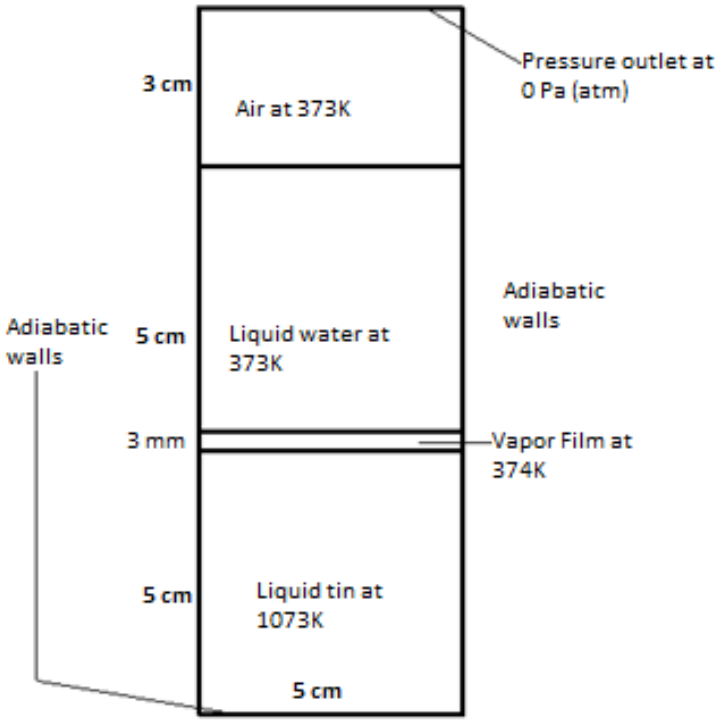

Figure 13: Geometry and boundary conditions of the simulated domain.

Table 4: Surface tension values use for the numerical simulations.

Phases	Surface tension
Melt-vapor	0.52
Melt-liquid water	0.468
Vapor-liquid water	0.072

I. Mesh and Time Convergence Study

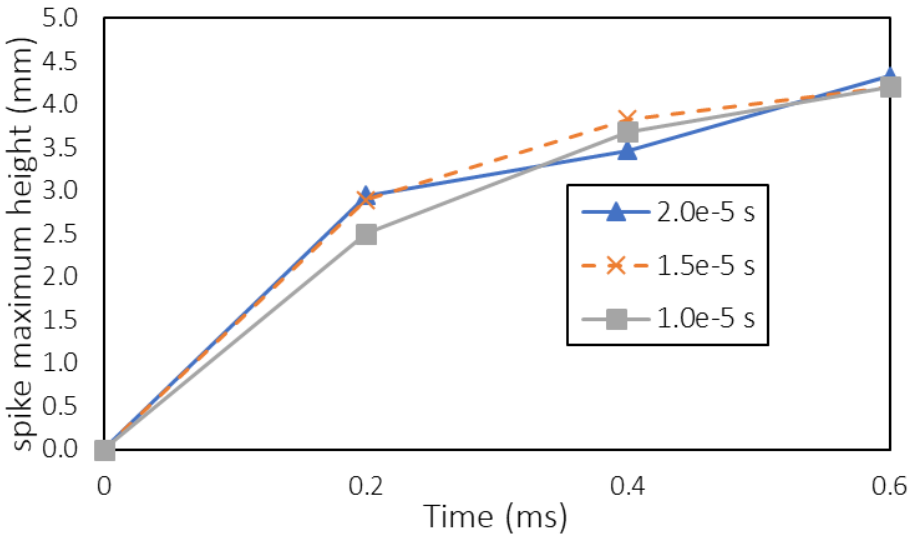

Figure 14 Time-step independence study: variation of spike maximum height with the time step size.

In systems with several phases, it is important to capture all the interfaces with a good accuracy. We created six different types of meshes to study the mesh convergence. And, the number of cells in the vapor film region was increased until similar measures were obtained in the height of the melt spikes versus time. Table 2 lists the minimum (h) element length, number of nodes (NN) and number of cells (NE) of various meshes considered for the mesh convergence study.

Table 5: Mesh convergence study for numerical simulations

	h (μm)	NN	NE
Mesh M1	500	7885	7711
Mesh M2	250	10643	10490
Mesh M3	125	21283	21148
Mesh M4	85	21265	21050
Mesh M5	93.75	30681	30579
Mesh M6	62.5	56222	56114

Figure 15 illustrates the mesh convergence study with different minimum element sizes, the study was made for a density of 7000 kg/m^3 and 100 MPa. The results with M4, M5, and M6 are similar. Therefore, we used the mesh M4 for the computations.

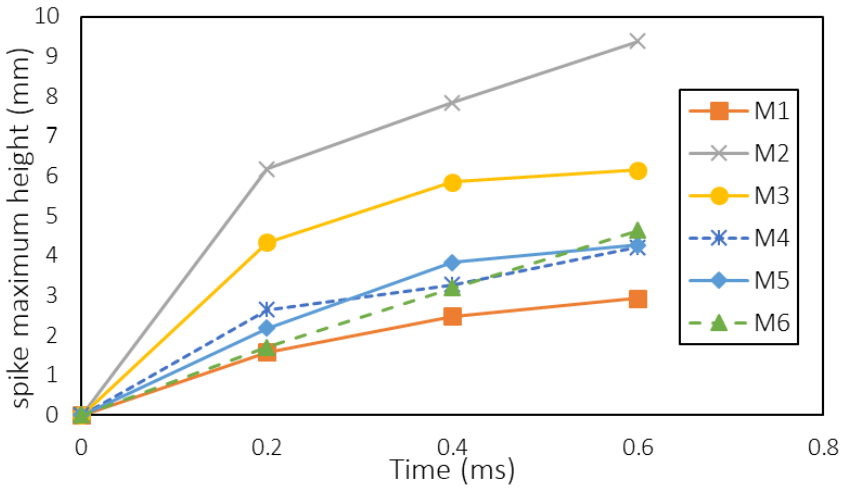

Figure 15: Mesh convergence study: variation of spike maximum height with time for different meshes

In addition, the mesh M4 contains approximately 2/3 of the number of elements in the mesh M5 and helps us to minimize the computational time.

Convergence in residuals was presented for both meshings, the criteria were 1e-3 for continuity, velocities and k-ε, and 1e-6 for energy. The mesh M4 had a 3 times faster convergence in residuals. The mesh M4 is shown in Figure 16. The average quality of the mesh is 0.8935.

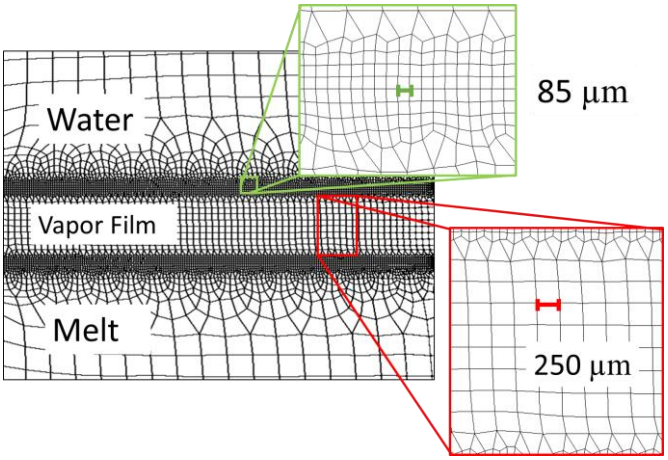

Figure 16: The computational mesh M4.

II. Methodology behind machine learning algorithm used for melting spike prediction

In order to train the machine learning algorithm, two data sets were created from the computational results obtained in ANSYS Fluent; each data set is composed of 25 data points. The first dataset contains the number of the metal spike as a function of time. The number of spikes present at a given time was determined through a manual count of the spikes present in the video frame.

The second data set is composed of the maximum height of the spike present in the video frame at a certain time. The spikes are measured by using the software Digimizer, a software where a reference length value is given to a certain number of pixels. Figure 17 shows a measurement made with Digitizer.

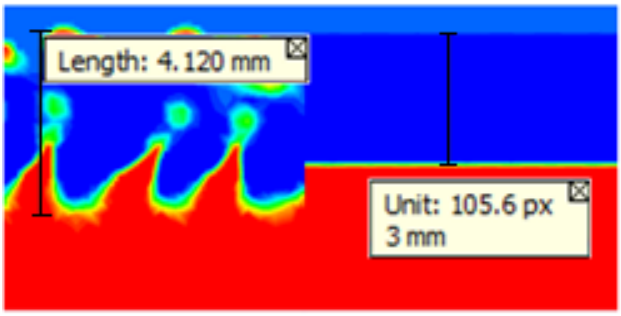

Figure 17: Measurement with Digitizer. In the right, reference measure is added indicating the vapor film thickness.

III. Datasets

In order to train the machine learning algorithm, two data sets were created from the computational results obtained in ANSYS Fluent; each data set is composed of 25 data points. The first dataset contains the number of the metal spike as a function of time. The number of spikes present at a given time was determined through a manual count of the spikes present in the video frame.

The second data set is composed of the maximum height of the spike present in the video frame at a certain time. The spikes are measured by using the software Digimizer, a software where a reference length value is given to a certain number of pixels. Figure 17 shows a measurement made with Digitizer.

5.6 Summary

This chapter starts by describing the motivation to develop the two type of numerical models being used in this thesis, followed by the description behind the methodology and procedure on the study of the sensitivity of surface tension and viscosity in the prediction of depth of penetration of molten metal by using the model EXPNS and performing uncertainty quantification techniques. Also, the machine learning algorithms along with their assumptions and development are described in detail, while including and methodology behind the data recollection, processing and implementation.

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Introduction

Following the development and application of the different machine learning, uncertainty quantification and high-performance computing models, the multiple physics has been resolved for the four different problems being aforementioned, and the results in the form of predictions have been obtained. This chapter is dedicated to present and summarize the results obtained with the computations performed.

6.2 Sensitivity of Surface Tension-Viscosity in Simulating Molten Hf, Ti, Y, and Zr Infusion into a B4C Packed-Bed at the Microscopic Scale

I. Numerical Simulation for molten metals

In this section, it is discussed in detail all the simulations and numerical experiments that have been conducted with EXPNS. We have in general considered 10^4 nodes, 100 in each direction. Initially, the simulations are conducted on 4 different metals at 2300^0C according to Table 2 and 3. Fig 4.3 shows us the penetration depth of a porous sample with a constant $5.5\ mm$ pore length and radii generated from a uniform random distribution between $1mm$ and $10mm$ for Hafnium with viscosity $\mu = 6.69 \times 10^{-3}$ Pa-s. We have considered 10^4 pores in the sample for this simulation. The statistical distribution of the radii samples is shown in Figure

In Figure 19 we can see that $\bar{y} \sim 1 \times 10^2(m)$ and the change in penetration depth is quite low as we move along in time, whereas which suggests that the rate of molten metal percolation becomes quite slow, which is suggested further by rate of penetration plot which shows that the flow starts to creep as we move forward in time. Moreover, the average rate curve is smooth throughout time, hence the perturbation of penetration depth along the breadth of the sample does not change in time. The perturbation arises due to the random nature of the geometry wherein cross flow amongst the pores of the network occur.

For Hf at melting temperature, in Figure 18, the average $\bar{y}$~0.10 (m), which remained fairly

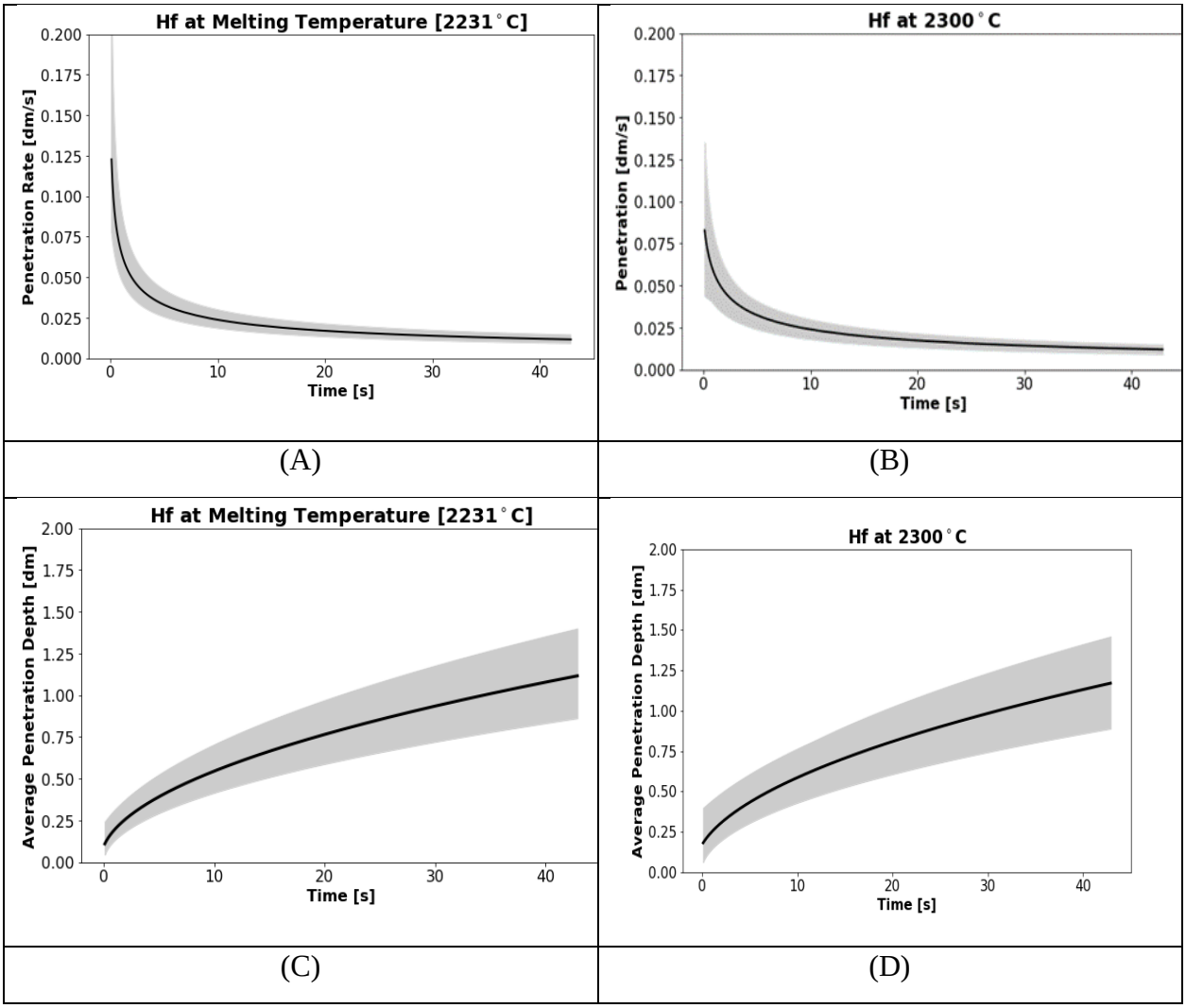

Figure 18: Predicted behavior for Hf infiltration

similar to the penetration depth predicted at 2300^0C. This could be explained by

,

Figure 19 shows us the percolation pattern of molten Titanium at both temperatures, as it percolates through the same sample. Titanium in Figure 19 (D) has a lesser viscosity than in Figure 19 (C) and hence penetrates further into the sample and goes beyond $\bar{y}$~0.19 (m), while Ti at melting temperature managed to reach an average penetration depth of 0.22 m.

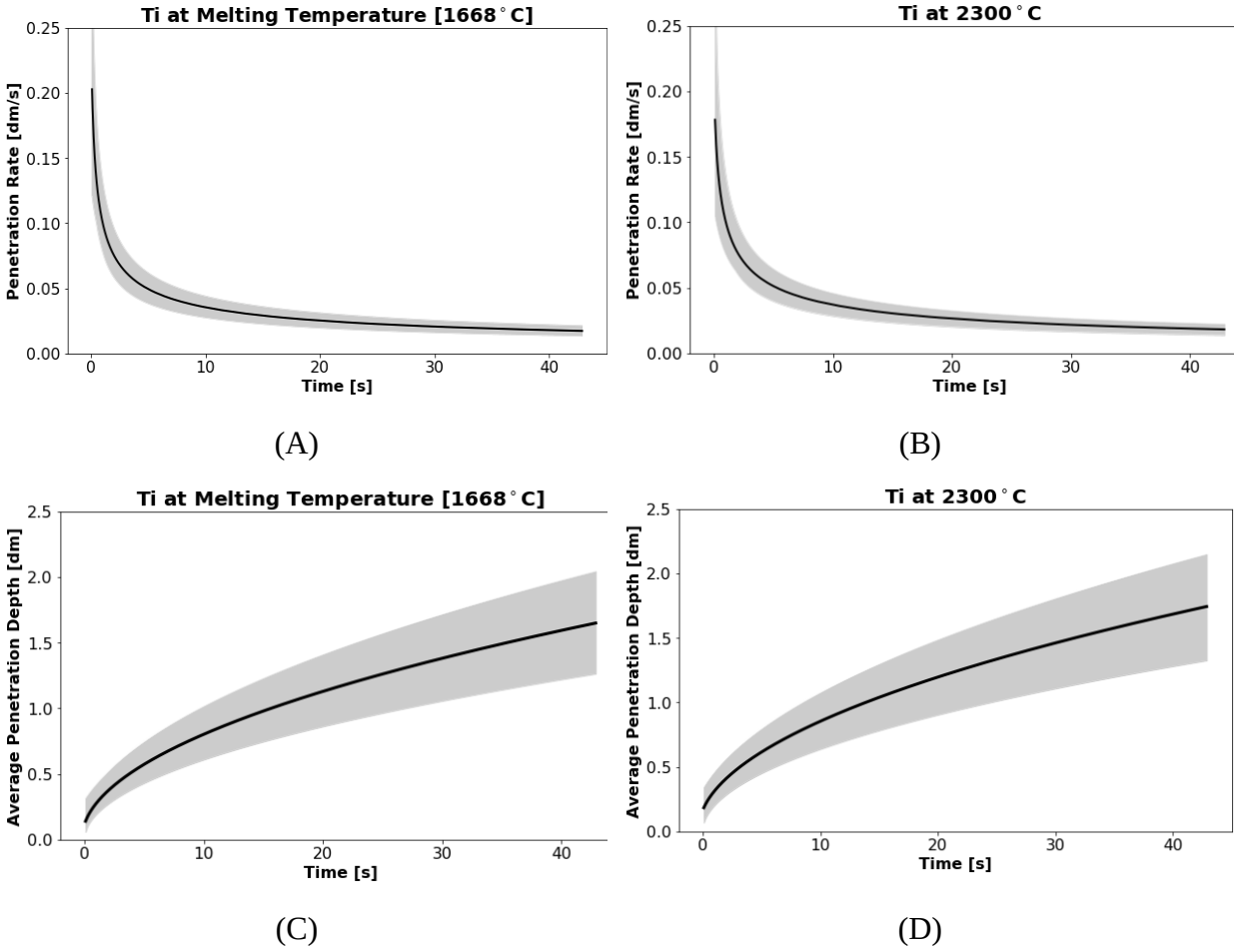

Figure 19: Predicted behavior for titanium at melting temperature

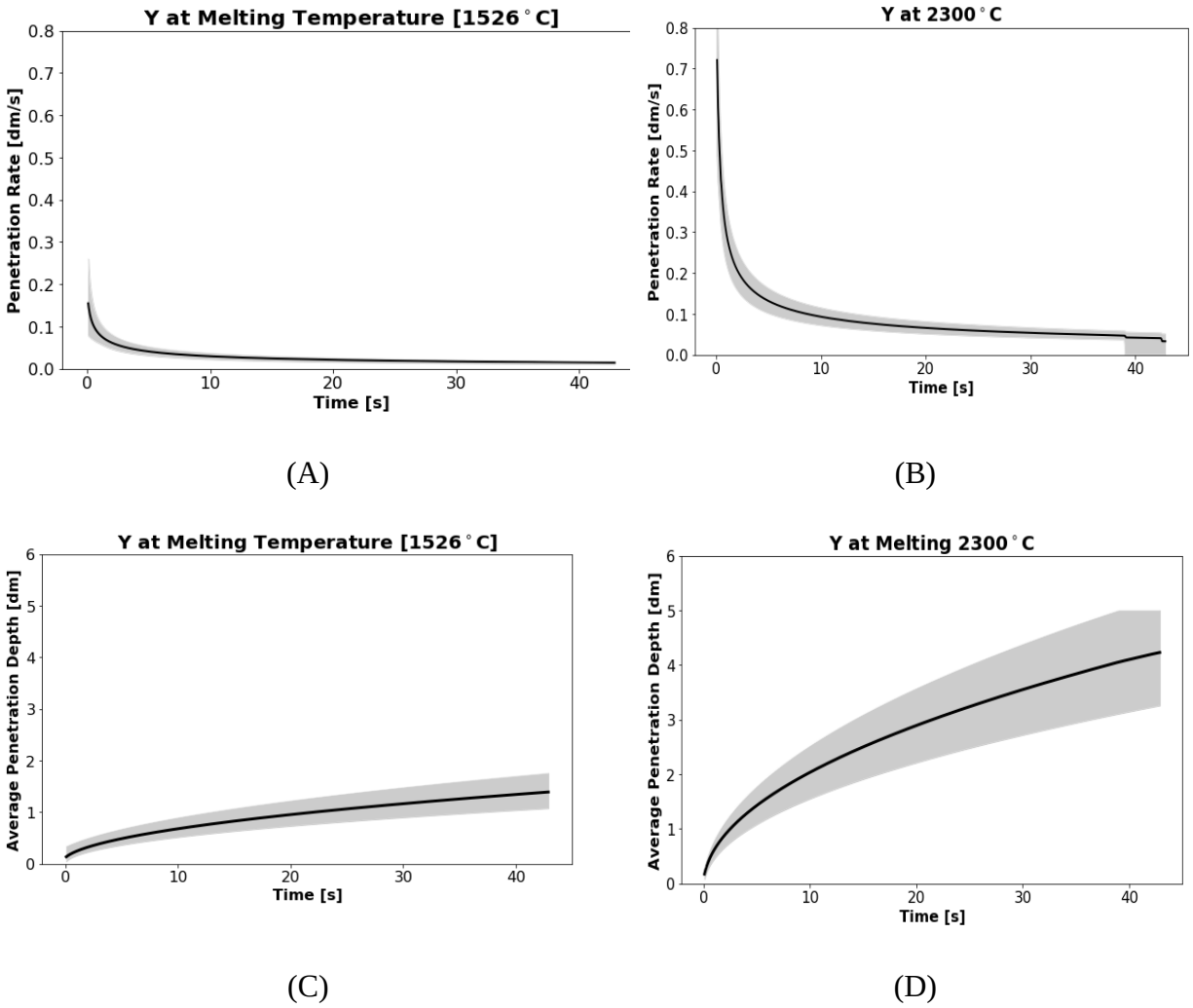

Figure 20: Prediction for Y percolation

Of all the four metals, yttrium is the metal that exhibits the widest patter of infiltration. Figure 20 describes the rate and depth of penetration of yttrium at both temperatures. It has reached a depth of penetration in average of 0.32m at 2300°C, and an average depth of penetration of 0.1m by the end of the infiltration

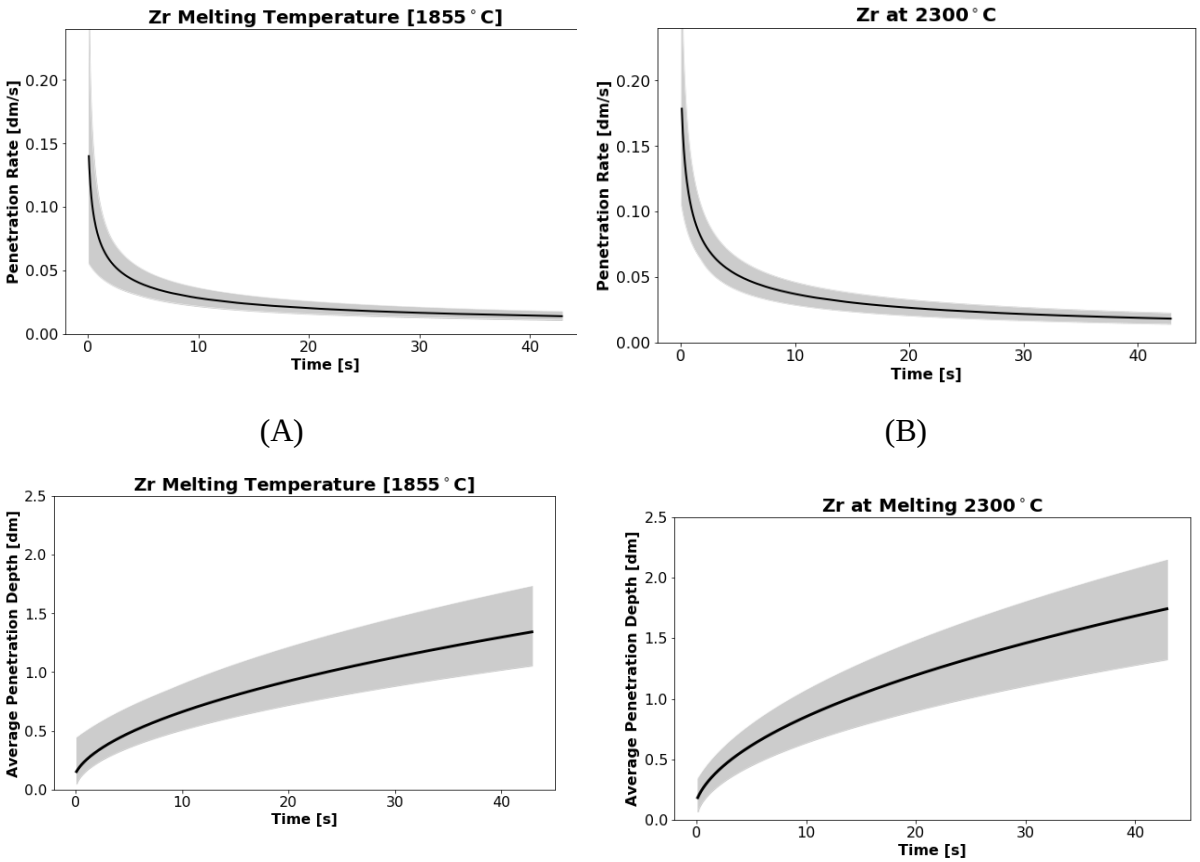

Figure 21: Prediction for Zr penetration depth

In case of Zirconium in Figure 21 whose viscosity lies between that of Titanium and Hafnium at $\mu = 3.0 \times 10^{-3}$ Pa-s, the behavior of the percolation is as expected and its average penetration depth at a temperature of 2300 Celsius lies somewhere between that of Titanium and Hafnium at $\bar{y}$~0.175 (m), while the prediction of percolation at melting temperature is close to $\bar{y}$~0.13 (m).

II. Uncertainty Quantification with Dakota

In this section we consider the pore network model as a separate simulation code that is integrated with Dakota[84] in order to perform statistical uncertainty quantification for EXPNS on the metal percolations. Here we consider that μ for each metal is an uncertain input parameter which varies as a log-normal distribution with its mean at the values given respectively in Figure 12. and its standard deviation ($s(\mu)$) varying as 1%, 5%, 10%, and 20% of its mean. The output parameter is $\bar{y}$ at $50\ s$. We perform this study for each of the metals t both $2300^{o}C$ and T_m. A Monte Carlo simulation has been performed with a Latin Hypercube Sampling of 20 sample points for each of these σ.

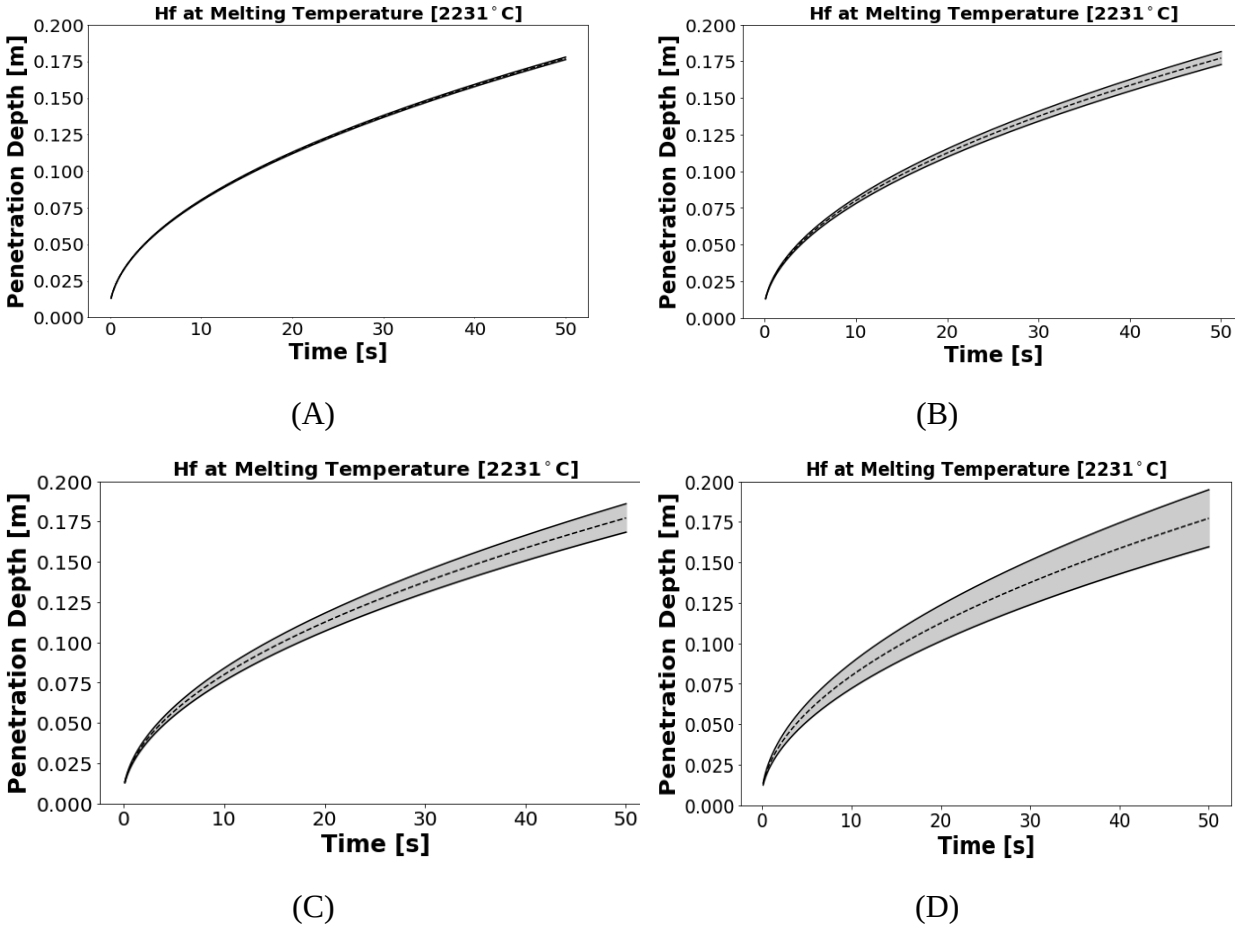

Figure 22: Prediction for Hafnium infiltration at melting temperature

In the case of Hafnium at melting temperature, we can see that a standard deviation of $\pm 7.1x10^{-5}\frac{Pa}{s}$ in the viscosity of the liquid metal translates curve line, which could be regarded to be just an application of the Rhines Equation, as appreciated in Figure 22 (A). For a standard deviation of $\pm 3.55x10^{-4}\frac{Pa}{s}$. there exists a difference of about 0.025 m difference between the minimum and maximum depth of penetration achieved at the end of the time (Figure 22 B). A standard deviation of $\pm 7.1x10^{-4}\frac{Pa}{s}$ will result in a maximum penetration depth of 0.180 m and a minimum 0.150 m, leaving the uncertainty range in about 0.30 meters. The maximum range of uncertainty occurs in Figure 23 (D), when the viscosity possesses a standard deviation which has a value of $\pm 1.42x10^{-3}\frac{Pa}{s}$, the range of uncertainty within that region is of about 0. 040 meters.

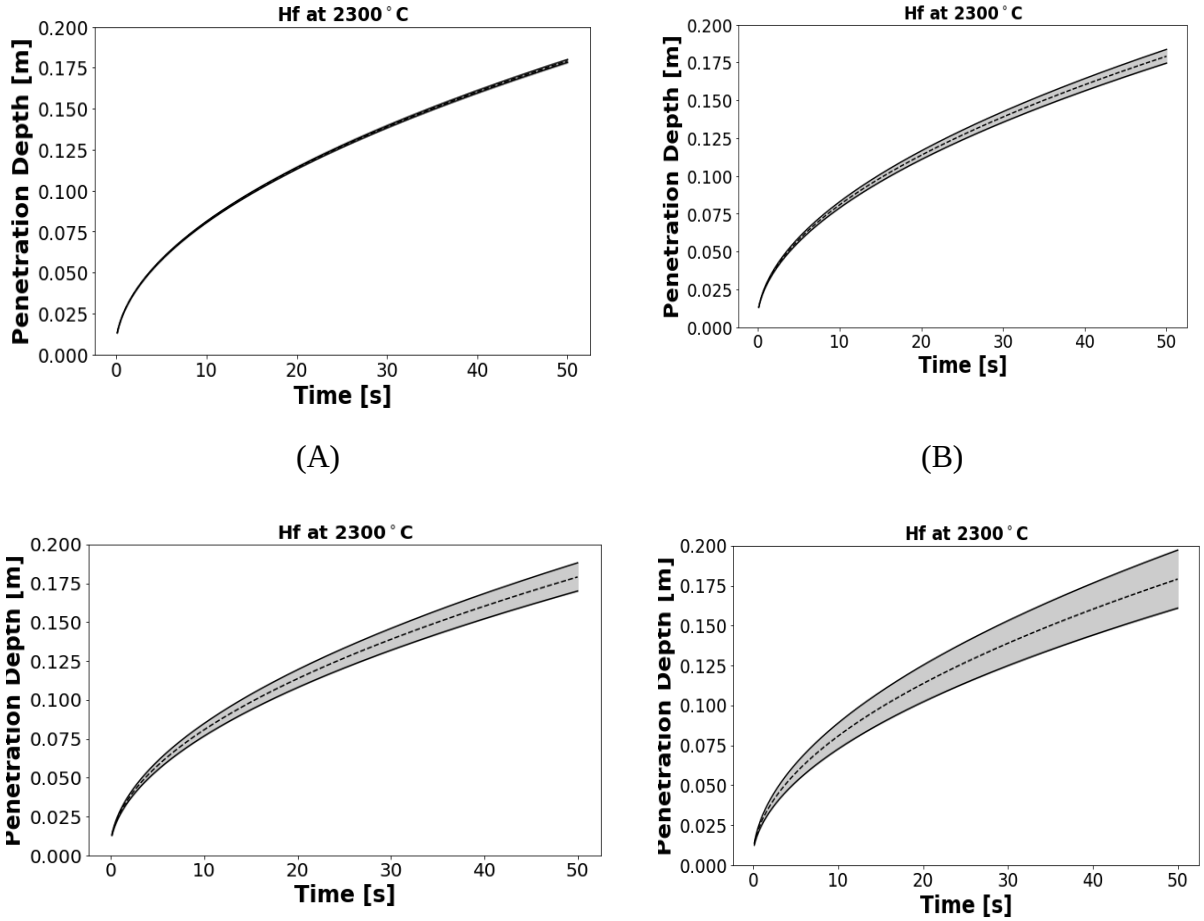

Figure 23: Uncertainty quantification study performed for Hafnium at 2300 Celsius

In the case of Hafnium percolating through a boron carbide packed bed at 2300 °C, it was observed that the uncertainty present in this case when the viscosity standard deviation reaches $\pm 6.69x10^{-5}\frac{Pa}{s}$, there is no uncertainty in the depth of penetration, with a constant depth of 0.175m. A standard deviation of $\pm 3.35x10^{-}\frac{Pa}{s}$ is translated into an output uncertainty of 0.025m. A standard deviation of $\pm 6.69x10^{-4}\frac{Pa}{s}$ produces a standard deviation of around 0.030 m and for $\pm 1.34x10^{-4}\frac{Pa}{s}$, it is noticed that the value of penetration rate can go at a value of up to 0.040m

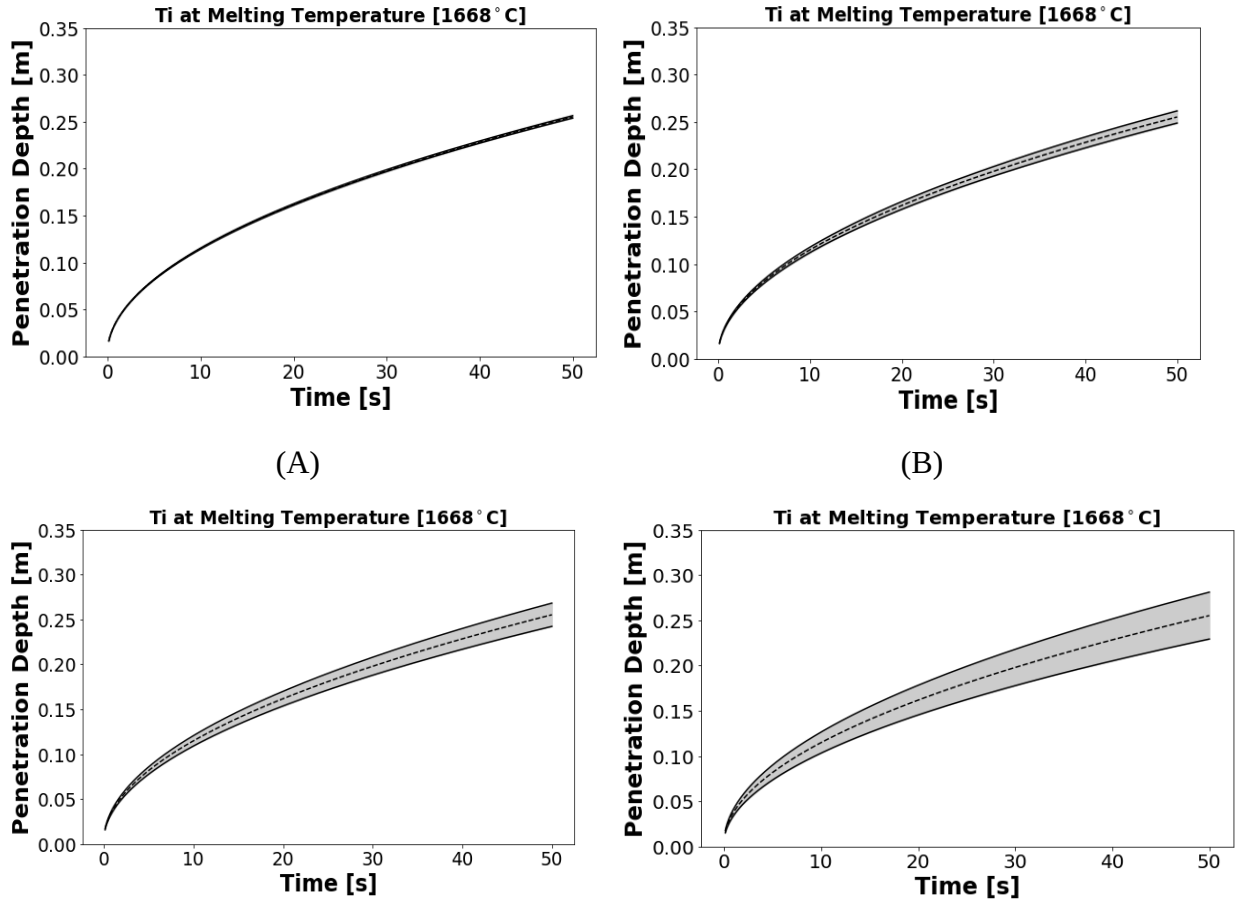

Figure 24: Prediction of Ti percolation at melting temperature

Figure 24 deals with the prediction of titanium at melting temperature inside of the pore packed bed. From here, it can be inferred that when a standard deviation in the viscosity of $\pm 4.03x10^{-5}\frac{Pa}{s}$, the uncertainty is certainly negligible and the depth of penetration remains without a perturbation. When the viscosity has a standard deviation of $\pm 2.015x10^{-4}\frac{Pa}{s}$, the perturbation achieved is of around 0.02m. At a standard deviation of $\pm 4.03x10^{-4}\frac{Pa}{s}$, the depth of penetration reaches disturbances of 0.04m. At the standard deviation of $\pm 8.06x10^{-4}\frac{Pa}{s}$, the difference in the prediction is of about 0.06m.

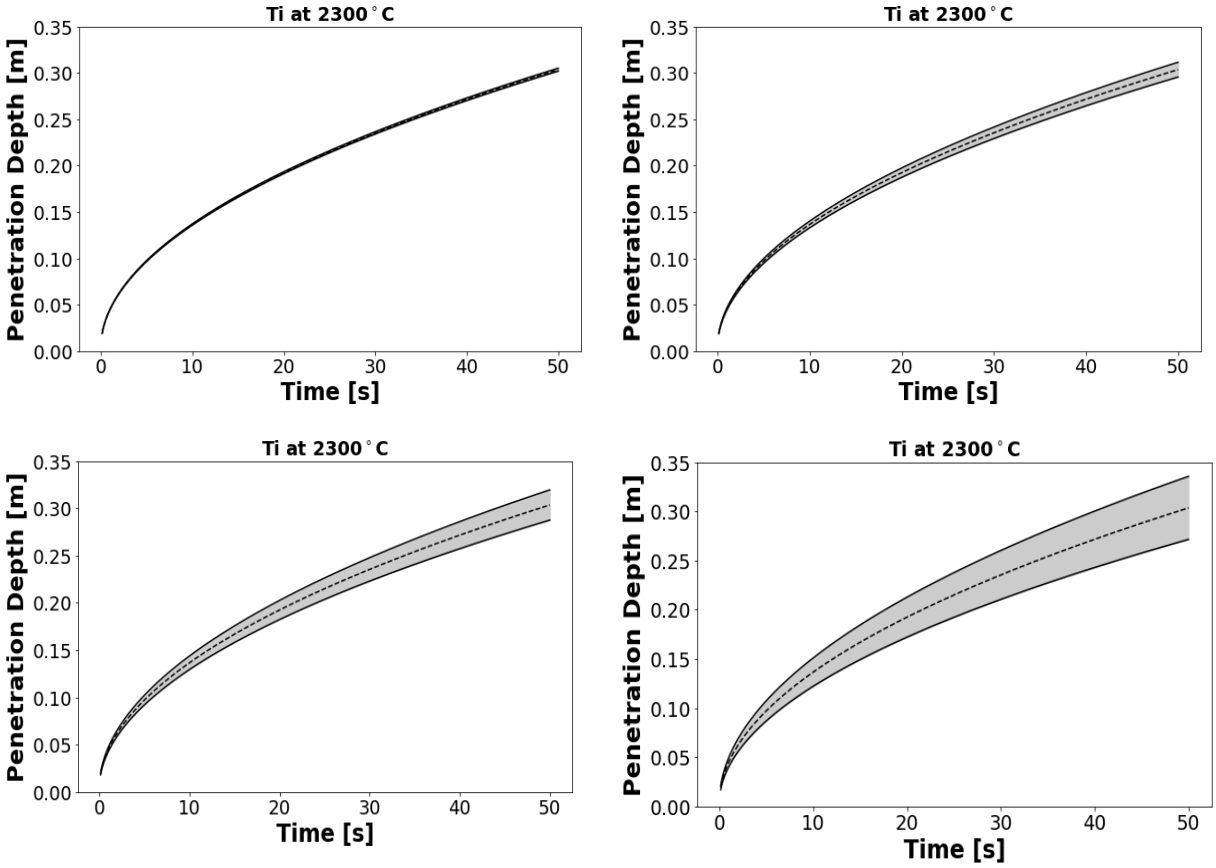

Figure 25: Prediction of Ti being percolated at 2300 Celsius

Based on Figure 25, when a standard deviation is applied to the system with a value of $\pm 3.3x10^{-5}\frac{Pa}{s}$, there is virtually no perturbation to the depth of penetration. However, when a perturbation of $\pm 1.65x10^{-4}\frac{Pa}{s}$ in the viscosity is applied, a perturbation of 0.02m is observed. After a perturbation of $\pm 3.3x10^{-4}\frac{Pa}{s}$ is applied to the input data, a variation of 0.04m, is observed in the depth of penetration after the infiltration time has been completed. Also, when a standard deviation of $\pm 6.6x10^{-4}\frac{Pa}{s}$ is present in the system, the penetration length will vary by approximate 0.011m

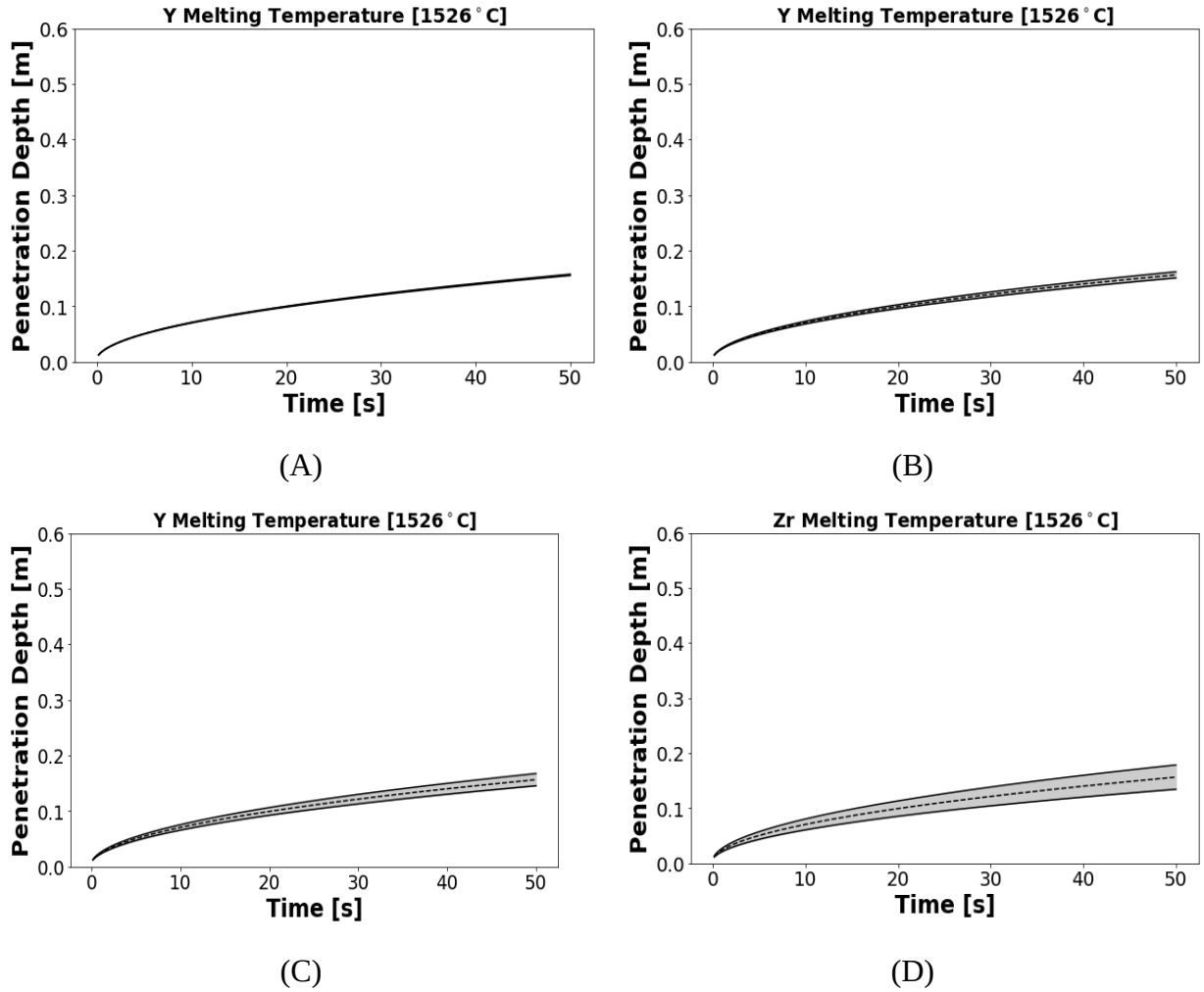

Figure 26: Uncertainty quantification of Yttrium at its melting temperature

After the prediction was generated and a standard deviation of $\pm 4.54x10^{-5}\frac{Pa}{s}$ is applied to the input viscosity, it is observed that this virtually causes no perturbation in the depth of penetration. When the standard deviation $\pm 2.27x10^{-4}\frac{Pa}{s}$ is given to the system, a perturbation of 0.01m, can be appreciated. At a standard deviation of$\pm 4.54x10^{-4}\frac{Pa}{s}$, a change of 0.03m is observed The value of $\pm 9.08x10^{-4}\frac{Pa}{s}$ is related to an uncertainty value of about 0.09 m.

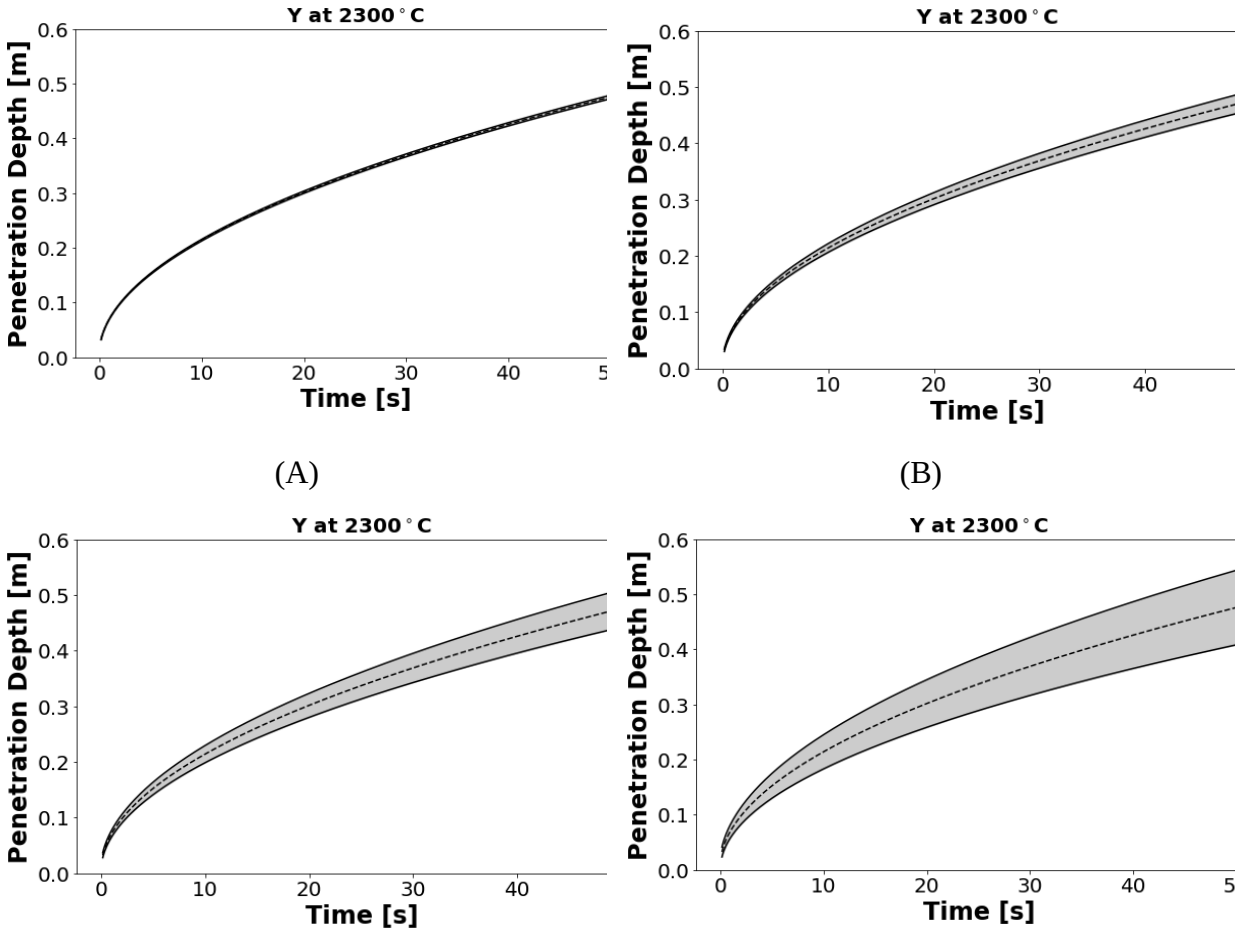

Figure 27: Uncertainty quantification study for Yttrium at 2300 °C

When a standard deviation of $\pm 4.91x10^{-6}\frac{Pa}{s}$ is applied to the system, almost no perturbation is present in the infiltration depth. With a value of $\pm 2.46x10^{-5}\frac{Pa}{s}$, a perturbation of around 0.05 is seen. At a standard deviation of $\pm 4.91x10^{-5}\frac{Pa}{s}$, the observed shaded area is of about 0.01. At a standard deviation of $\pm 9.82x10^{-5}\frac{Pa}{s}$, there is a difference in the depth of penetration of 0.2 m

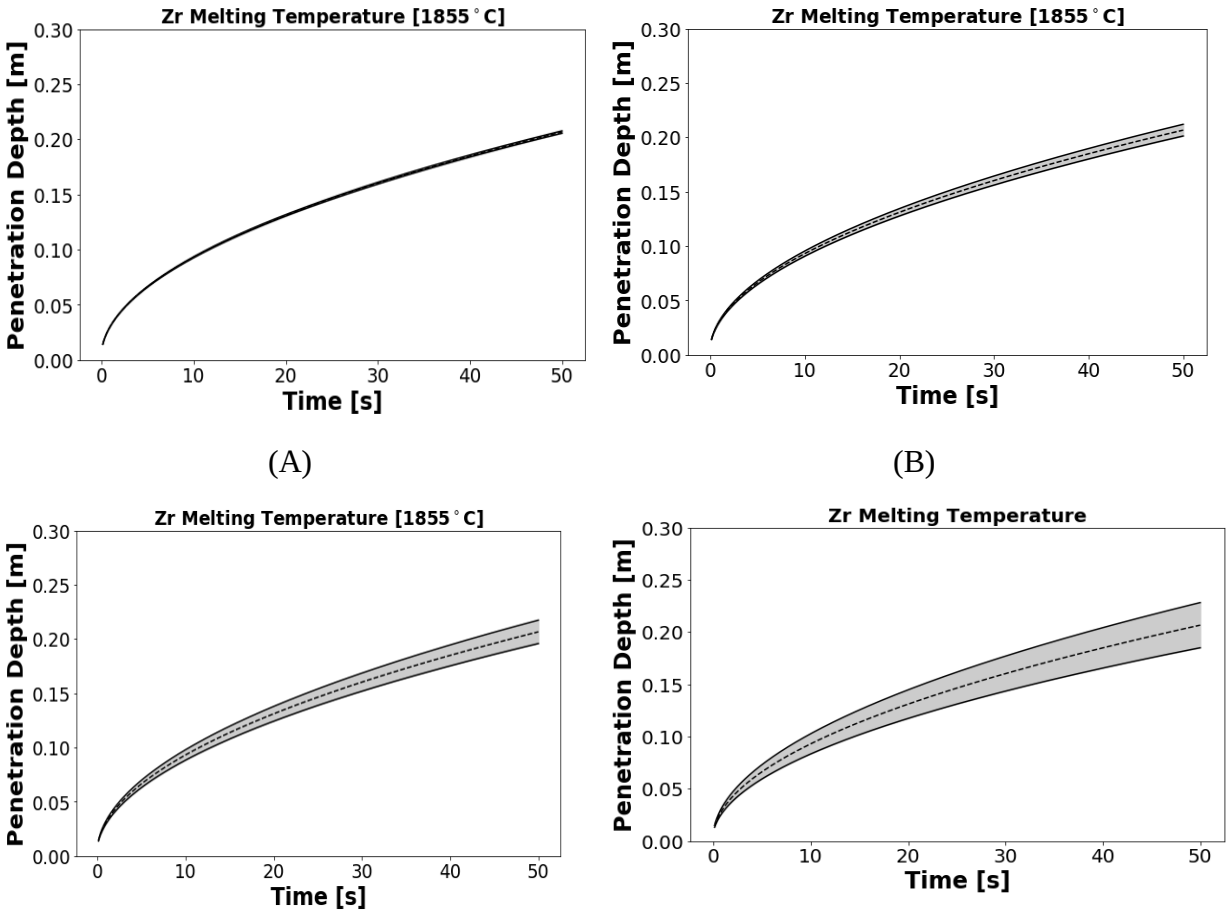

Figure 28: Uncertainty quantification for zirconium at its melting temperature

The first value of standard deviation tested, which is $\pm 4.74x10^{-5}\frac{Pa}{s}$, saw no change in the depth of penetration when the runs were sampled, $\pm 2.37x10^{-4}\frac{Pa}{s}$ leads to a perturbance of 0.04m, the value of $\pm 4.74x10^{-4}\frac{Pa}{s}$ reaches a difference in maximum penetration depth and minimum penetration depth of 0.06 m, and a standard deviation of will $\pm 9.48x10^{-5}\frac{Pa}{s}$ will lead to a depth of penetration of about 0.08m.

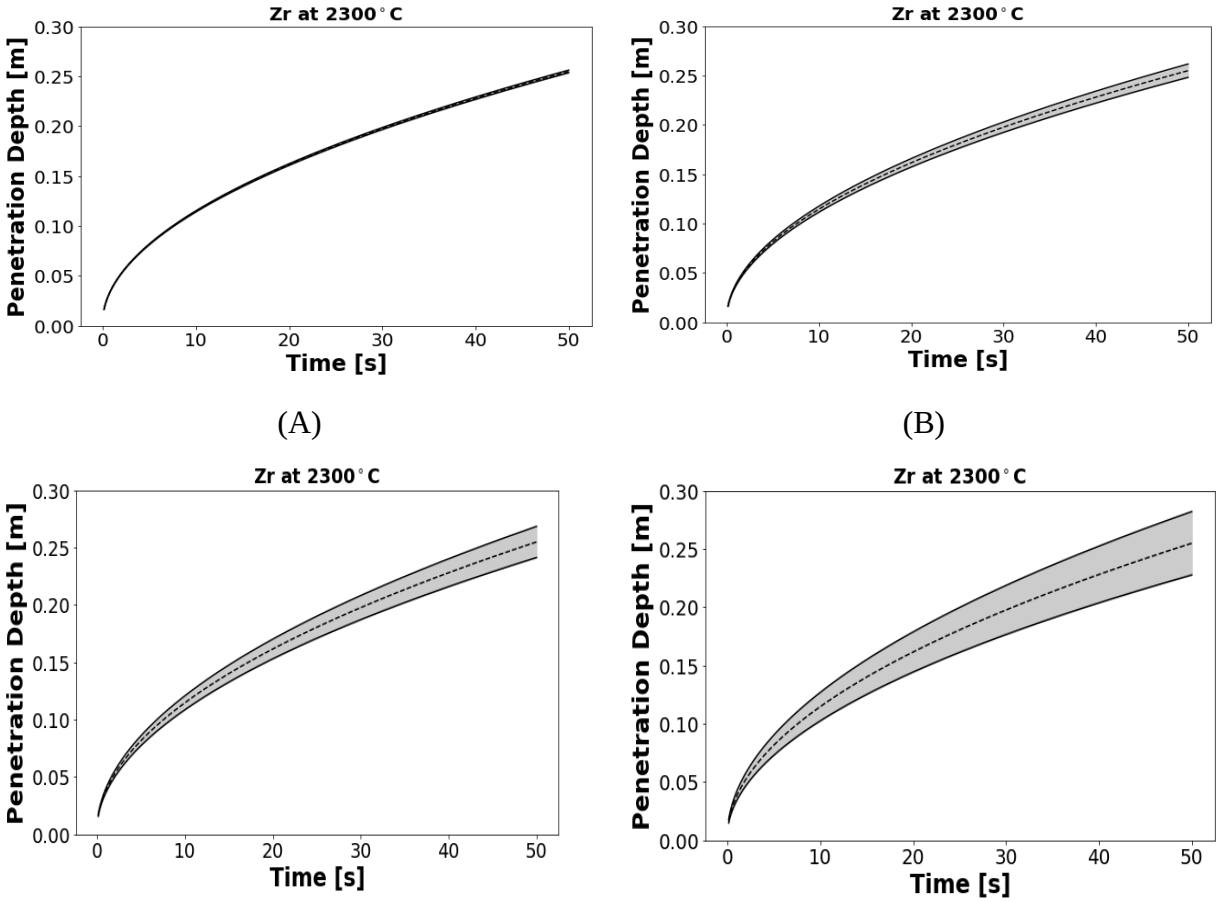

Figure 29: Uncertainty quantification for zirconium percolating at 2300 °C

When the standard deviation is $\pm 3.00x10^{-6}\frac{Pa}{s}$, there is no perturbation to the infiltration of the liquid zirconium at 2,300°C. At a standard deviation of $\pm 1.5x10^{-4}\frac{Pa}{s}$, the variation of depth of penetration is quantified at 0.02m. With a value of $\pm 3.00x10^{-4}\frac{Pa}{s}$, the penetration depth perturbation can be counted as 0.04 m and at $\pm 6.00x10^{-4}\frac{Pa}{s}$, it is possible to observe a perturbation of 0.07m.

6.3 Predicting the Depth of Penetration

Figure 30 through Figure 33 depicts the infiltration predictions. This results were obtained with the DNN algorithm afore mentioned in previous sections. Every image represents the theoretical approximation calculated by using the Semlak-Rhine's equation, along with the prediction obtained with the deep neural network.

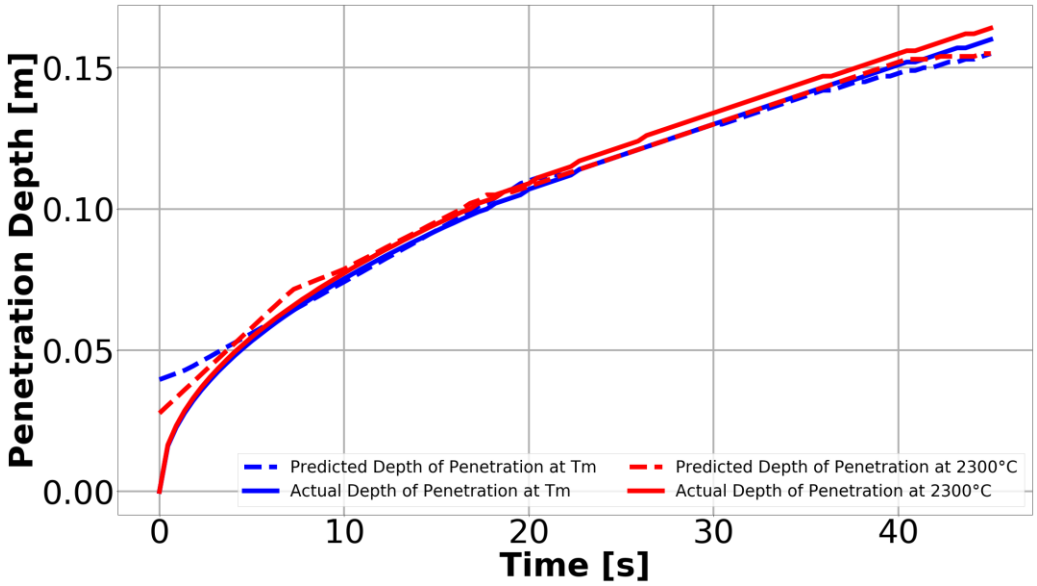

Figure 30: Prediction of infiltration of Hafnium realized with TensorFlow

Looking at Figure 30, it can be appreciated that Hf still is sticking to the behavior observed in the previous chapter section. The maximum depth of penetration that was predicted was calculated analytically happened at 2300 °C, with an estimate depth of penetration of 0.165 m. Also, the change in depths of penetration at different temperatures does not have a significant variation, which is expected, since the melting temperature of hafnium $(2231\ °C)$ is really close to the $2300\ °C$ temperature.

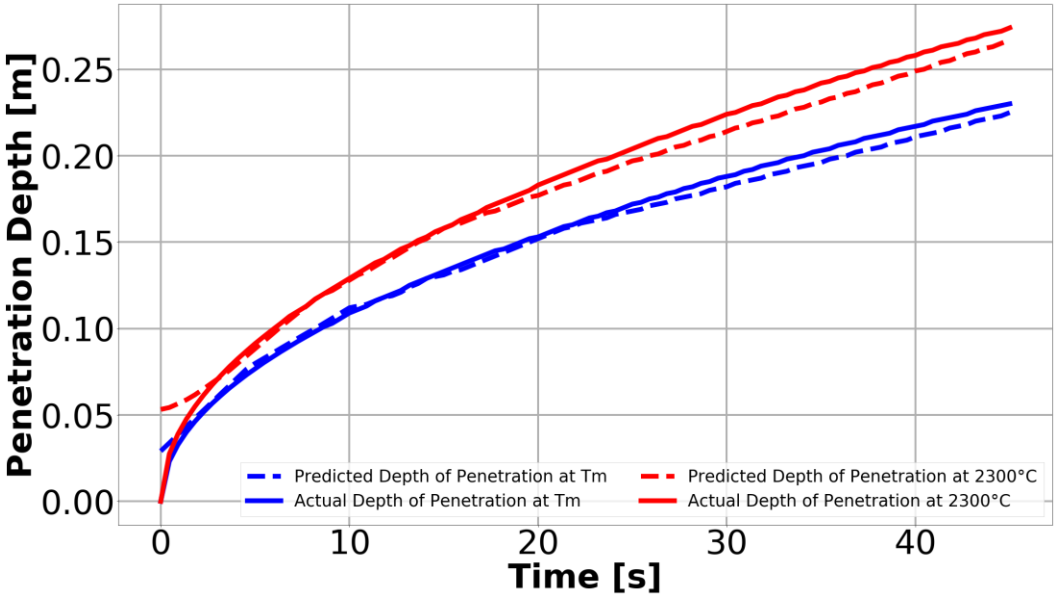

Figure 31:Prediction of infiltration of Titanium realized with TensorFlow

Regarding the infiltration of Titanium shown in Figure 31, it can be seen that the infiltration at melting temperature reached a height of 0.23 m, while the infiltration predicted at a temperature of $2300\ °C$ has reached a height of about 0.26 m.

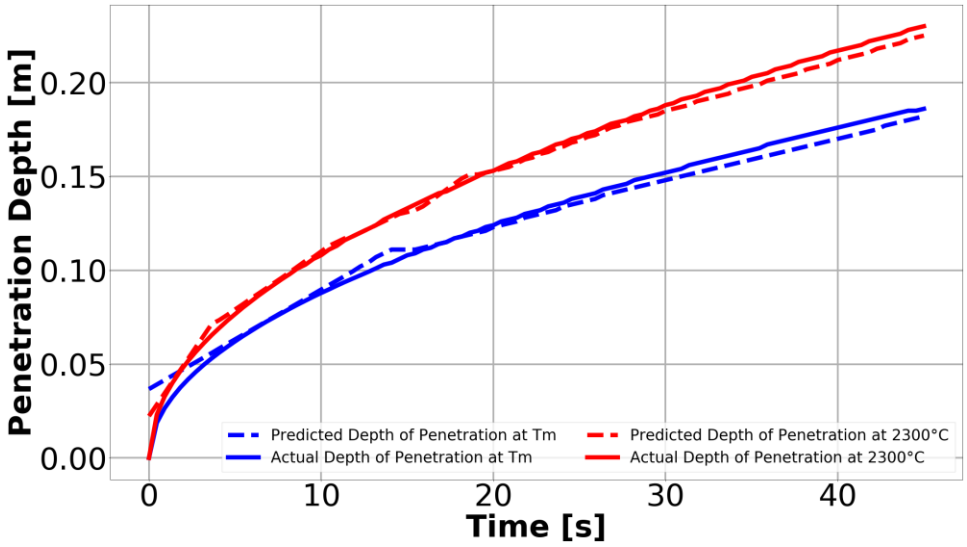

Figure 32: Prediction for the infiltration of zirconium.

By evaluating Figure 32, it is possible to understand the percolation of zirconium. The height reached after 45 seconds of infiltration at melting temperature equals 0.17 m, while the height reached 0.23 m

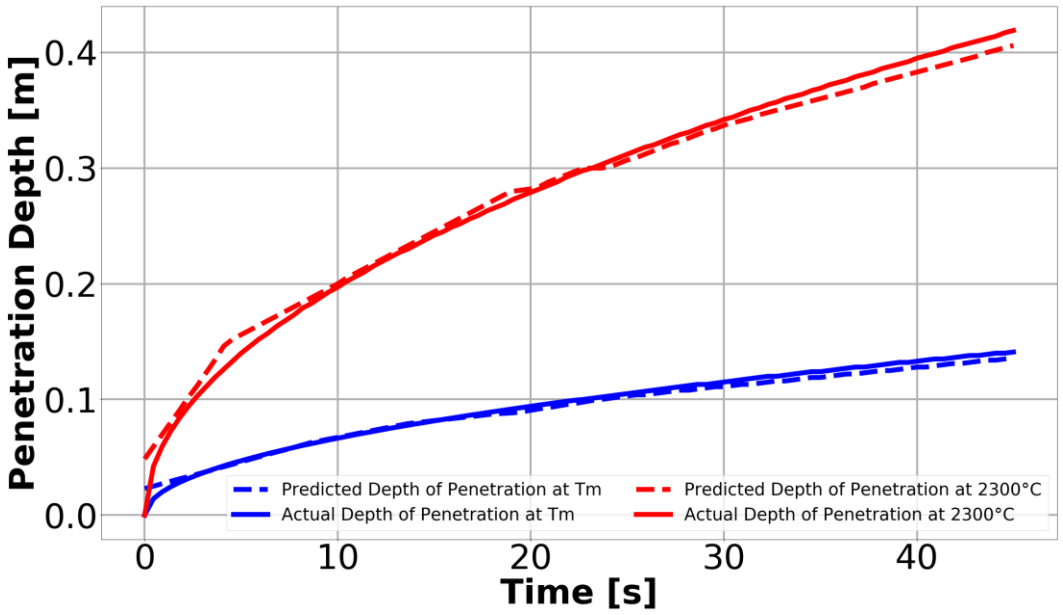

Figure 33: Prediction for the infiltration of Yttrium

Figure 33 shows the prediction of Yttrium. As expected, the values of the depth of penetration of Yttrium at the temperature of 2300 °C and the ones calculated at their melting temperature differs, which follows the overall trend predicted with EXPNS and Dakota. It is observed that the machine learning algorithm imitates the infiltration behavior really nice at melting temperature.

I. Convergence Study

A convergence study was performed in order to understand the number of data points that will give the most accurate prediction. For this, a training set and a validation set was created. The training set is composed of 2,000 data points, always depicting the depth of penetration as a function of time. Also, a test set was created, showing penetration heights at a different time than in the training sets.

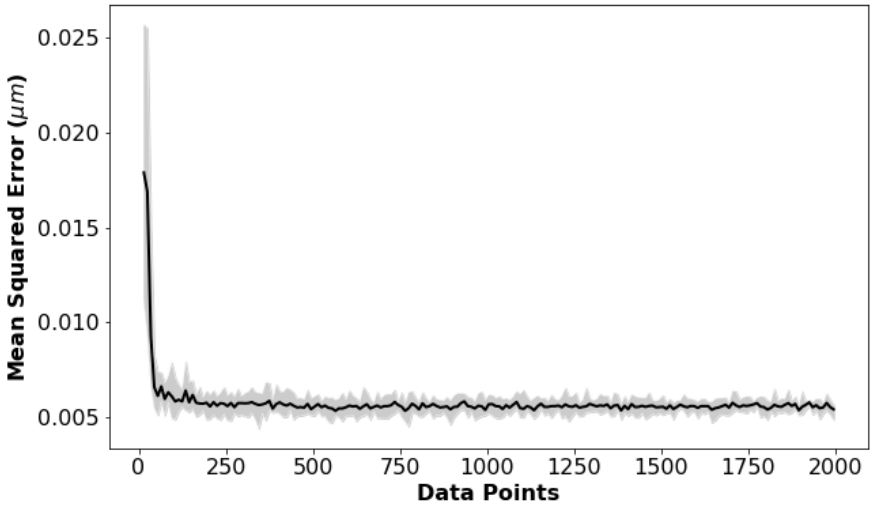

Figure 34: Convergence study for DNN

Figure 34 presents a comparison of the MSE as N increases. Based on this figure, it is concluded that the least amount of error (of around 0. 0.0014 μm) can be achieved by setting N to be 2,000.

6.4 Numerical Simulation of Formation of Melt Jets in Melt Coolant Interactions

Once the geometry and necessary setup were made for the 2-D simulation of the stratified water/liquid metal system, measurements of the melt spikes length were obtained and studied.

Figure 35 shows the formation sequence of the melt jets through time, they look similar to the results in Ciccarelli and Frost [2] that were digitally processed in Figure 36. Since no measurements for the melt spike length were made in [2], we show only qualitative agreement between the results and the simulation.

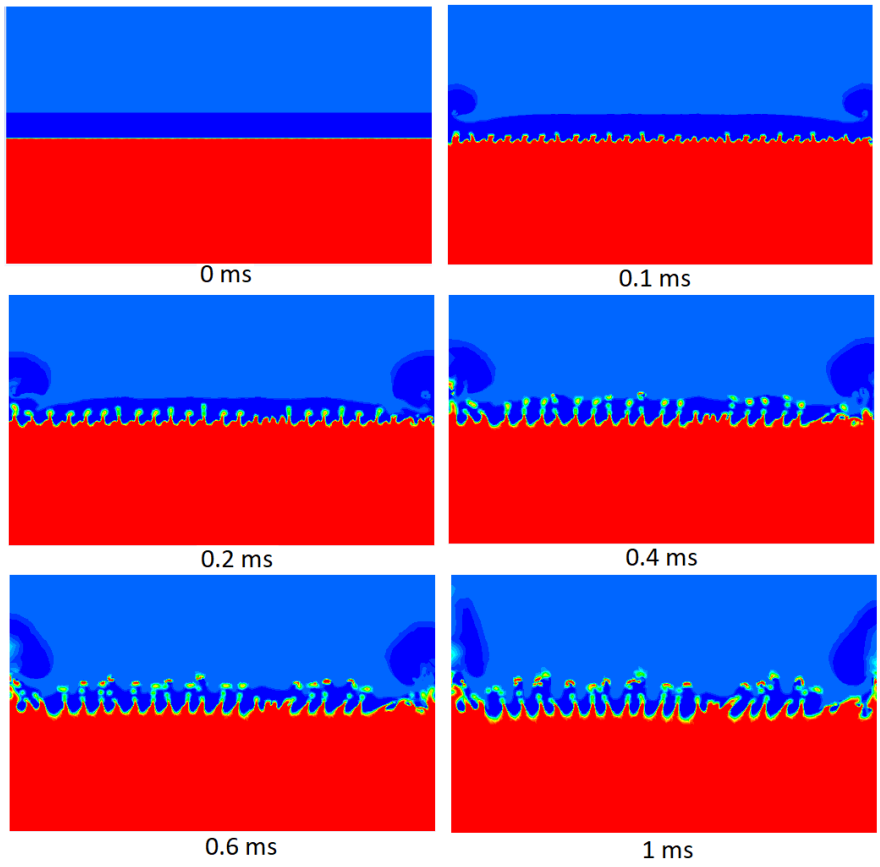

Figure 35:Density contours at various time instants for pressure pulse of 100 MPa.

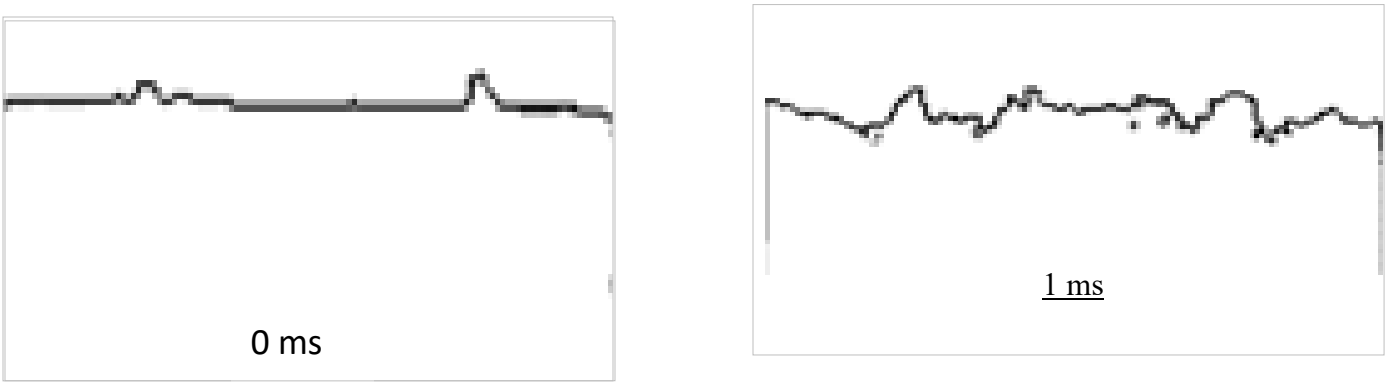

Figure 36: Digital processing of Ciccarelli and Frost [92] results for melt jet formation

In Figure 36, it can be observed that at time= 0 ms the system is at rest and there are no perturbations in the vapor film which is located between the water and liquid tin. The spikes appear in the flow as the time progresses. They grow in the vapor film zone for time ≤ 0.2ms. After 0.2 ms the melt jets get fragmented for pressure pulse of 100 MPa and the density of the melt of 7000 kg/m3. The decrease in the pressure values results in little spikes or ripples of melt for the same density. Figure 10 shows the effect of the melt density at the pressure of 40MPa. It is observed that the increase in density increases the spike height. For density < 10000kg/m3, the spike maximum height does not vary with time. However, the spike height increases with time for ρ = 10000kg/m3. Figure 11 shows the variation of spike maximum height with time for various initial pressures. For this case, the melt density remained constant at 7000 kg/m3. The melt spikes height increases with an increase in the pressure values. The formed craters push the melt upwards. The force that pushes the melt increases with increases in the melt density. This can be seen in Figure 12 where two different densities were simulated. Figure 13 shows the effect of having different initial pressure values in a qualitative way. It is observed that the increase in initial pressure increases the perturbations around the vapor film and increases the spike height. The increase in the density increases the melt spike height as well.

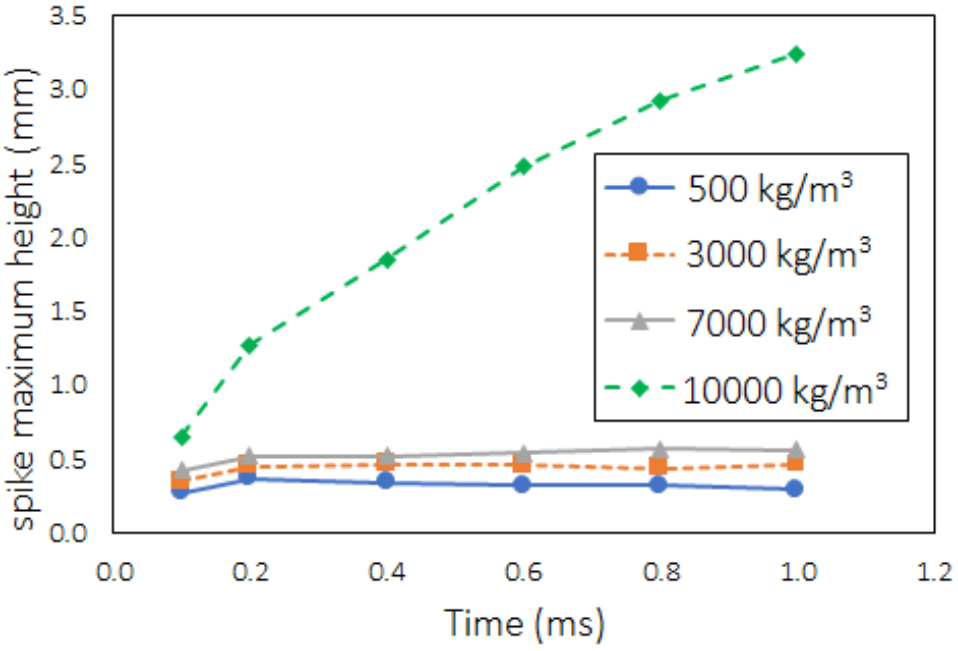

Figure 37: Variation of jet length with time for various melt densities when initial pressure is 40 MPa.

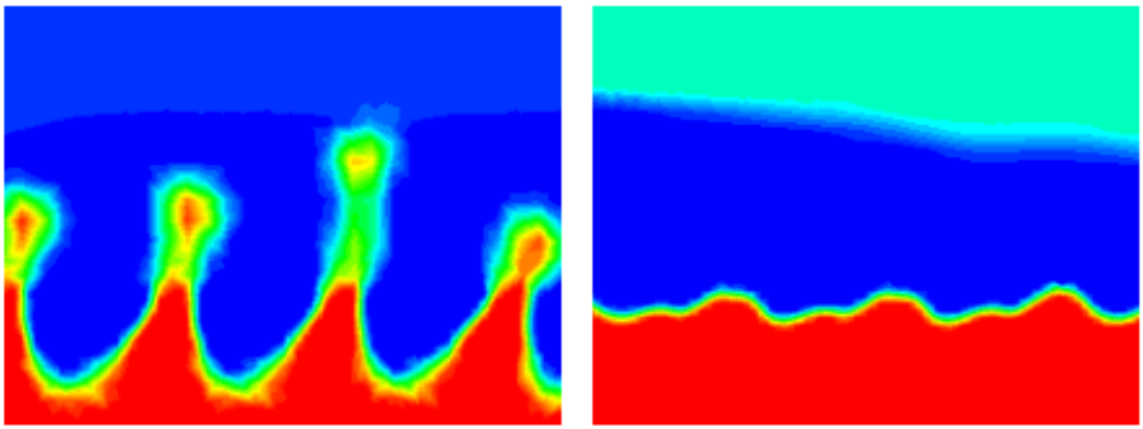

Figure 38 Density contour with a pressure of 40 MPa and a density of (a) 10000 kg/m^3 and (b) 3000 kg/m^3, both at 1 ms

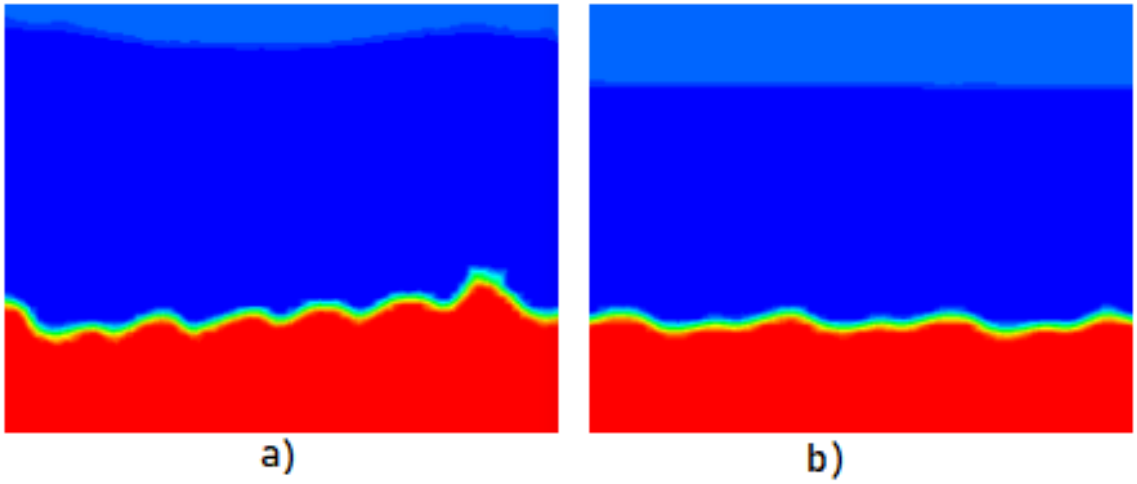

Figure 39: Density contour with a density of 7000 kg/m^3 and an initial pressure of (a) 40 MPa and (b) 10 MPa, both at 1 ms.

I. Machine Learning Results

In order to supervise the machine learning training, the DNN was validated by using the same training data as a validation set.

The behavior of the formation of the melt jets was analyzed with deep neural network machine learning algorithm which results are presented in Figure 40 and Figure 41.

Figure 40 depicts the machine learning prediction for the number of spikes produced when a pulse pressure of 100 MPa is applied to the system. The values predicted with the deep neural network neural algorithm are same as the actual values. And, the mean square error is 0.0 and the maximum absolute error is 1 spike.

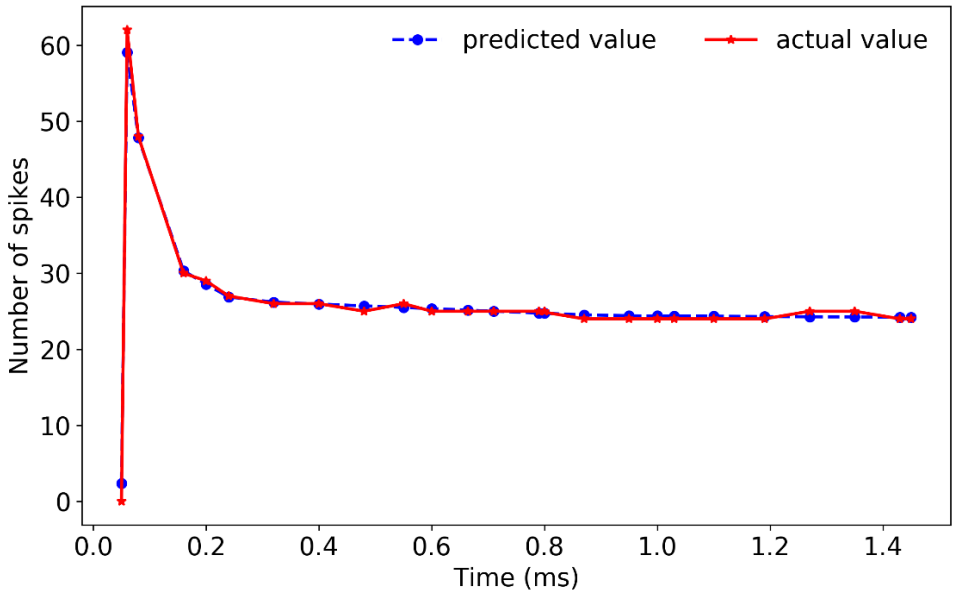

Figure 40: Comparison of actual number spikes and the value predicted by the machine learning algorithm.

On the other hand, Figure 15 represents a comparison of the actual and predicted maximum spike length at various times for the pulse pressure of 100 MPa. The predicted values are in good agreement with the actual values. And, the max error between the actual and predicted value is 0.6 mm.

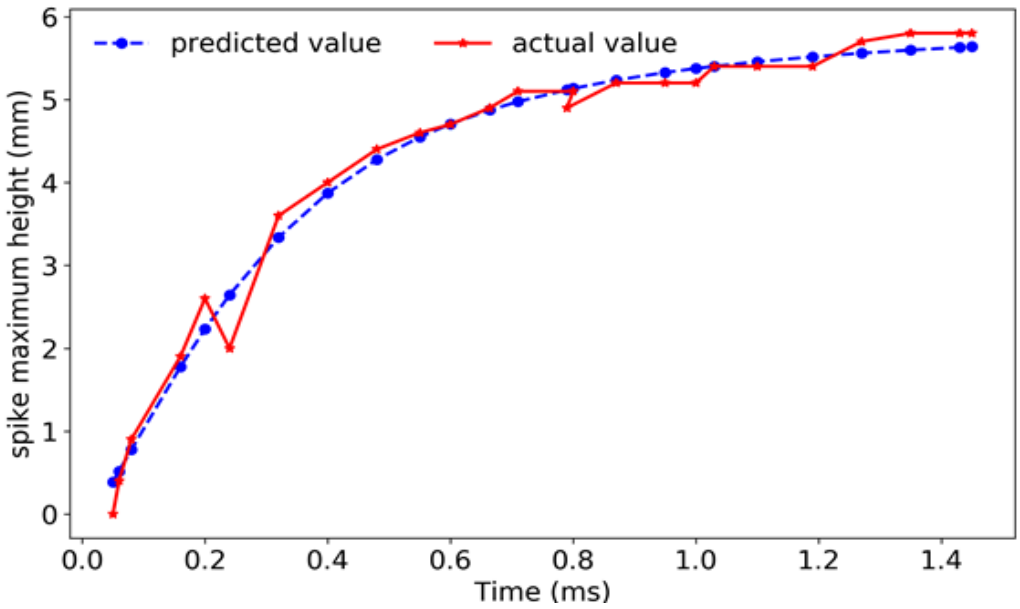

Figure 41 :Comparison of actual maximum spike length and predicted maximum spike length

6.5 Machine Learning Approach to Predict the Flow Rate for An Immiscible Two-Phase Flow at Pore Scale for Enhanced Oil Recovery Application

I. Multivariate Linear Regression

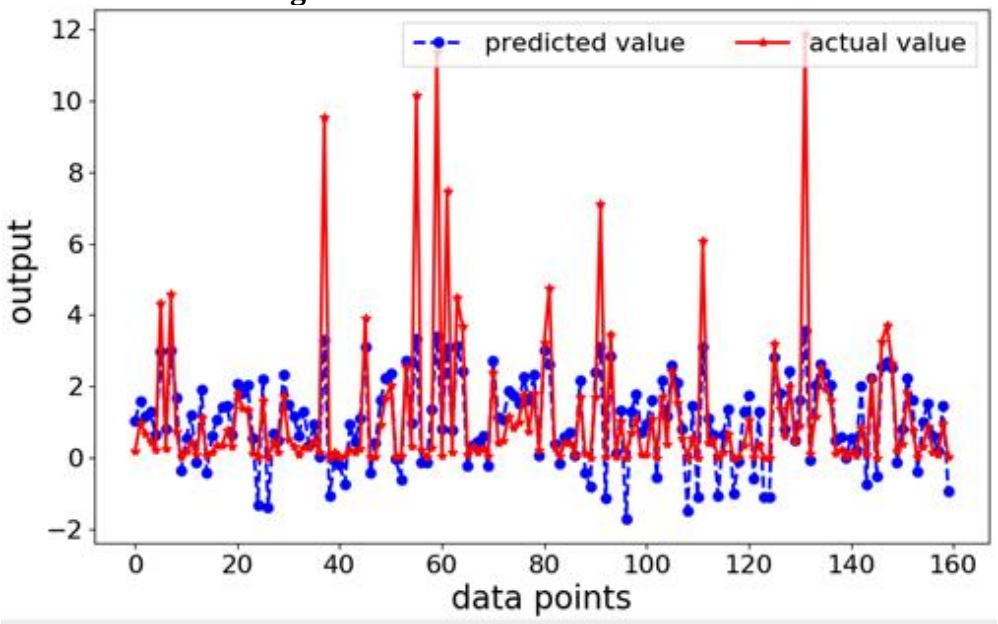

Figure 42: Multivariate linear regression prediction.

II. Random Forest Algorithm

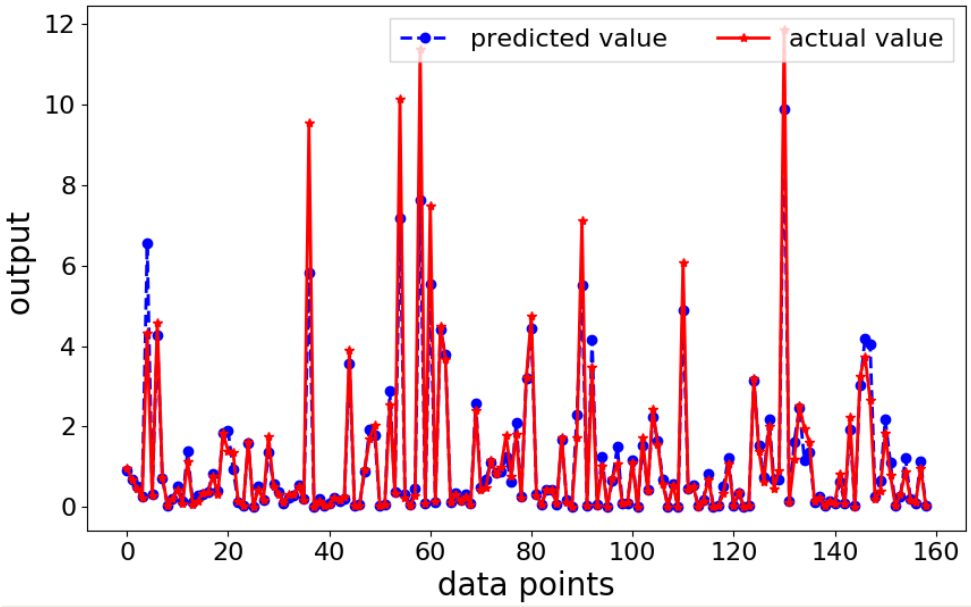

Figure 43: Volumetric flow rate variables values for Random Forest.

III. Deep Neural Network

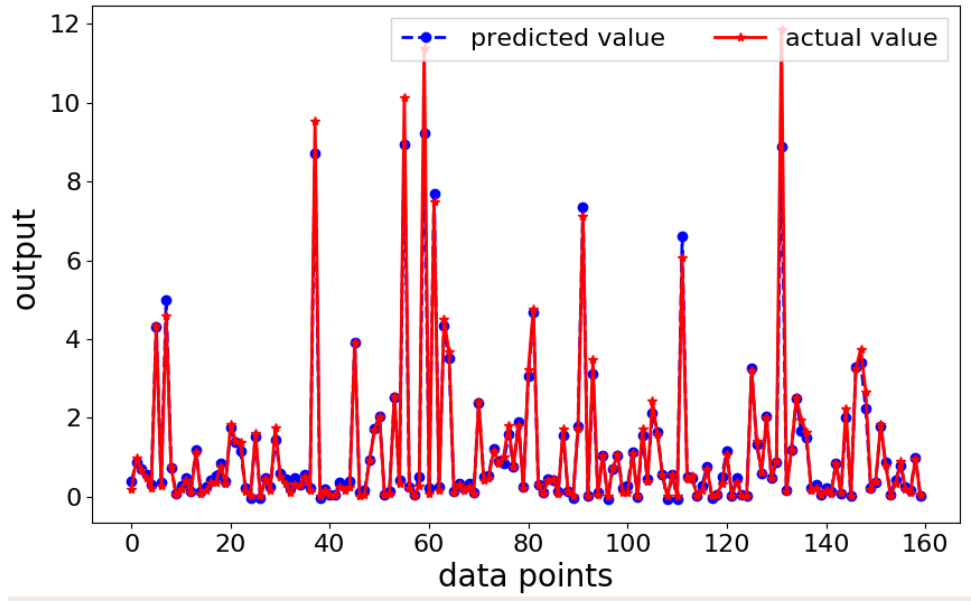

Figure 44 Volumetric Flow Rate Values for DNN

IV. Statistical Analysis and Results

Variance is the knowledge of knowing the probability arrangement of choosing a value that is around the mean center of the distribution. In a better explanation is that is the square standard deviation, meaning that is the squared distance distribution from the mean. By squaring the

standard deviation, we can clarify that the variance cannot be a negative number. Having a high variance would mean that the most weight is in the high probability values. For example, if you have a high variance would imply that the values are more likely to be chose if they have more weigh close to the actual values in the distribution[112]. The mean square error is the calculation of the error between the predicted and actual values square. Meaning that as the mean squared error gets smaller, the error is less in the prediction.

Table 6: Statistical Analysis of the three main TF Models

	Multivariate Linear Regression	Random Forest	Deep Neural Network
Mean Square Error	2.1	0.38	0.104
Variance	0.49	0.91	0.97

6.5 Summary

The physics predicted by TensorFlow and EXPNS has been described in this chapter. For the sensitivity study of the flow of molten metal performed with EXPNS and Dakota, two predictions have been generated. One is based in the microscopic features that define the porous media, and the second one encompasses the variation of the molten metal properties, specifically the potential change in viscosity being present in the real-life scenario. Regarding the machine learning side of this study, eight predictions have been made for the infiltration of molten metal through the microstructure, two predictions for the coolant explosions have been proposed, and a third prediction set was generated when trying to predict the flow rate for enhanced oil extraction.

CHAPTER 7: CONCLUSION

In this thesis, two data driven tools for the prediction of complex multi-physics engineering problems were developed, implemented and used in order to generate accurate and high fidelity predictions for the depth of penetration of molten metal into a powder packed bed, an approach to predict the length and number of molten metal spikes as a function of time as been tested and implemented, along with preliminary predictions that could potentially benefit the oil industry by predicting the volume flow rate for enhanced oil recovery techniques

7.1 Surface Tension- Viscosity in Simulating Molten Hf, Ti, Y and Zr Infusion into a B4C Packed-Bed at the Microscopic Scale

This study presents the numerical simulations of liquid metal flow through a porous structure representing a B_4C packed bed at both their melting points and a higher operating temperature. We have simulated the flow for a stochastic porous structure where the radii follow a uniform random distribution. It has been observed that owing to the randomness in the geometry, there is a presence of cross flow inside the porous structure that perturbs the moving front of the liquid metal and we have quantified the average penetration depth and penetration rate for the both the temperature and all four metals. Following these simulations, we have conducted Monte Carlo simulations, considering the viscosity of the liquid metal and boundary conditions as uncertain variables and quantified the variation of the average penetration depth with the variation of both viscosity and boundary conditions. All the simulations have been carried out a massively parallel in-house software that had been developed for exa-scale simulations and uncertainty quantification.

7.2 Predicting the Depth of Penetration of Molten Metal into A Pore Network Using TensorFlow

In this study, a deep neural network algorithm was developed and used to predict the deep of penetration for the infiltration of molten metal inside of a B_4C ceramic bed. A convergence study was performed in order to understand the relationship between N, and n. It is found that after 500

data points, the mean square error starts to reach asymptotic behavior and that it will be at its lowest point at 2,000 data points.

The maximum theoretical depth of penetration has been calculated and compared with the predictions obtained with the DNN algorithm for the four molten metals used in this study

Based on the results obtained, the deep neural network has proven to be a robust tool for the modeling of molten metal infiltration, since it has the capability of analyzing the infiltration process all the way to the micro-scale.

7.3 Numerical Simulation of Formation of Melt Jets in Melt Coolant Interactions

We have shown that molten metal fragmentation in the presence of water can be observed through computational fluid dynamics techniques. A simplified two-dimensional simulation was able to capture the formation of metal spikes under conditions (e.g., high pressure) similar to those reported previously. Water penetration was not observed but the chance cannot be discarded. Hence, the 3D simulation will be carried out in the future.

Machine learning has proven to be a very useful tool to analyze this problem since it provides good predictions that provide useful insight into this problem. For future work, it is considered to incorporate a computer vision algorithm to automatically process the images obtained from simulations and experiments.

7.4 Machine Learning Approach to Predict the Flow Rate for an Immiscible Two-Phase Flow at Pore Scale for Enhanced Oil Recovery Application

In this study, an immiscible two-phase flow through porous media has been analyzed through the Hagen-Poiseuille analytical method using a hybrid version of it because the problem is assumed to be a pipe void analysis problem. In the hybrid version of the Hagen-Poiseuille equation, there is a need for a coupling between the pressures of the two fluids, while introducing an effective viscosity quantity. The Hagen-Poiseuille equation applies to this problem because the problem is

Newtonian, laminar, incompressible, and internal. Incorporated viscous fingering effects are shown in the literature, which applies to this problem and can present a way of expecting an effect during porous media experimental and simulation results. For predicting the volumetric flow rate using machine learning tool; TensorFlow, to generate three models to predict data without going through the analytical solution. TensorFlow is incredible machine learning tool that can run in small processing, while also used to solve big data problems using parallel clusters. The method of TensorFlow presented in this paper is generalized in supervised learning algorithms because they are the proven to be efficient algorithms on which data is validated. From the data collected from the three models, the one that predicts the data the best is the Deep Neural Network, due to its low mean square error, high variance, and proven results. Compared with two other models, Random Forest leaning towards the second-best method, while Multivariant to be the worst method to predict values. This is due to the non-linear approach used in Random Forest and linear approach for the Multivariant method.

CHAPTER 8: FUTURE WORK

8.1 Surface Tension- Viscosity in Simulating Molten Hf, Ti, Y and Zr Infusion into a B4C Packed-Bed at the Microscopic Scale

There are several features that show potential for improvement in EXPNS. The first improvement that should be addressed consists approaching the addition of a third-dimension parameter, in order to be able to predict the rate and depth of penetration in the *z* direction. The second improvement consists in the incorporation of a conductance formulation for irregular geometries, since perfect regular geometries are rarely found in real world applications. Another possible improvement would be the incorporation of image processing software, in order to incorporate the micro-structure features of genuine porous media for different applications.

8.2 Predicting the Depth of Penetration of Molten Metal into A Pore Network Using TensorFlow

Regarding the prediction of depth of penetration by using TensorFlow, it is necessary to better capture real data for molten metal infiltration process; this could be achieved by performing the actual infiltration under laboratory conditions, and help to quantify the properties of the molten metal in real time. Currently, the equipment is being installed and configured to manufacture the UHTC. As mentioned in the previous section, a very important improvement will consist in the incorporation of computer vision algorithms in order to describe the micro-structure features that have a very important influence in the depth of penetration achieved. Also, it is important to work on the development of an unsupervised machine learning algorithm. The first step to develop this unsupervised algorithm consists in establishing the reward mechanism that the algorithm will use to predict the conductivity of the system.

8.3 Numerical Simulation of Formation of Melt Jets in Melt Coolant Interactions

Future research ventures for this study will consists of the addition of a bigger data set size, which could be potentially obtained through in-depth examination. Ideally, the data should come

from experimentation, however, it is possible to train the algorithm from synthetic data obtained from CFD simulations. Another important addition for this algorithm will be the incorporation of a deep learning model instead of a machine learning algorithm, which will help to reduce the human error in the measurements and give some new insights into the fine composition of the formation of metal spikes in the furnace. Another improvement that should be implemented for this project consists in the implementation of 3D modeling with the purpose of obtaining the trajectories of the molten metal droplets.

8.4 Machine Learning Approach to Predict the Flow Rate for an Immiscible Two-Phase Flow at Pore Scale for Enhanced Oil Recovery Application

Future enhancements for this project would consist in improving the quality of the data used in this investigation. Instead of producing synthetic data, the possibility of using training sets coming from real oil extraction sites must be examined. Providing more training sets to the algorithms will result in better predictions for the oil recovery application.

BIBLIOGRAPHY

[1] Askeland, D. R., Fulay, P. P., and Wright, W. J., 2011, The Science and Engineering of Materials.

[2] Jinnapat, A., and Kennedy, A., 2012, "Characterisation and Mechanical Testing of Open Cell Al Foams Manufactured by Molten Metal Infiltration of Porous Salt Bead Preforms: Effect of Bead Size," Metals (Basel)., 2(4), pp. 122–135.

[3] Zhang, G.-J., Guo, W.-M., Ni, D.-W., and Kan, Y.-M., 2009, "Ultrahigh Temperature Ceramics (UHTCs) Based on ZrB 2 and HfB 2 Systems: Powder Synthesis, Densification and Mechanical Properties," J. Phys. Conf. Ser., 176, p. 012041.

[4] Gasch, M. J., Ellerby, D. T., and Johnson, S. M., 2005, "Ultra High Temperature Ceramic Composites," Handbook of Ceramic Composites, N.P. Bansal, ed., Springer US, Boston, MA, pp. 197–224.

[5] Liu, G. W., Muolo, M. L., Valenza, F., and Passerone, A., 2010, "Survey on Wetting of SiC by Molten Metals," Ceram. Int., 36(4), pp. 1177–1188.

[6] Castrodeza, E. M., Mapelli, C., Vedani, M., Arnaboldi, S., Bassani, P., and Tuissi, A., 2009, "Processing of Shape Memory CuZnAl Open-Cell Foam by Molten Metal Infiltration," J. Mater. Eng. Perform., 18(5–6), pp. 484–489.

[7] Lopez-Esteban, S., Saiz, E., Moya, J. S., and Tomsia, A. P., 2005, "Spreading of Viscous Liquids at High Temperature: Silicate Glasses on Molybdenum," Langmuir, 21(6), pp. 2438–2446.

[8] Saiz, E., Tomsia, A. P., Rauch, N., Scheu, C., Ruehle, M., Benhassine, M., Seveno, D., de Coninck, J., and Lopez-Esteban, S., 2007, "Nonreactive Spreading at High Temperature: Molten Metals and Oxides on Molybdenum," Phys. Rev. E, 76(4), p. 41602.

[9] Rrcou, R., and Viws, C., 1982, "LOCAL VELOCITY AND MASS TRANSFER MEASUREMENTS IN MOLTEN METALS USING AN INCORPORATED MAGNET PROBE," Inr. J. Heat Mass Transf., 25(IO), pp. 1579–1588.

[10] Blunt, M. J., 2001, "Flow in Porous Media - Pore-Network Models and Multiphase Flow," Curr. Opin. Colloid Interface Sci., 6(3), pp. 197–207.

[11] Kumar, V., Harris, C. K., Bronson, A., Shantha-Kumar, S., and Medina, A., 2016, "High-Temperature Liquid Metal Infusion Considering Surface Tension-Viscosity Dissipation," Metall. Mater. Trans. B Process Metall. Mater. Process. Sci., 47(1), pp. 108–115.

[12] Nishida, Y., and Ohira, G., 1999, "Modelling of Infiltration of Molten Metal in Fibrous Preform by Centrifugal Force," Acta Mater., 47(3), pp. 841–852.

[13] Delgado, P., and Kumar, V., 2015, "A Stochastic Galerkin Approach to Uncertainty Quantification in Poroelastic Media," Appl. Math. Comput., 266, pp. 328–338.

[14] Saiz, E., and Tomsia, A. P., 2004, "Atomic Dynamics and Marangoni Films during Liquid-Metal Spreading," Nat. Mater., 3(12), pp. 903–909.

[15] De Gennes, P. G., 1985, “Wetting: Statics and Dynamics,” Rev. Mod. Phys., 57(3), pp. 827–863.

[16] Joekar-Niasar, V., and Hassanizadeh, S., 2008, “Insights into the Relationships among Capillary Pressure, Saturation, Interfacial Area and Relative Permeability Using Pore-Network Modeling,” Transp. Porous Media.

[17] Gostick, J., Ioannidis, M., and Fowler, M., 2007, “Pore Network Modeling of Fibrous Gas Diffusion Layers for Polymer Electrolyte Membrane Fuel Cells,” J. Power Sources.

[18] Glantz, R., and Hilpert, M., “Pore Network Modeling,” columbia.edu.

[19] Hinebaugh, J., Fishman, Z., and Bazylak, A., 2010, “Unstructured Pore Network Modeling with Heterogeneous PEMFC GDL Porosity Distributions,” J. Electrochem.

[20] Valvatne, P. H., and Blunt, M. J., 2004, “Predictive Pore-Scale Modeling of Two-Phase Flow in Mixed Wet Media,” Water Resour. Res., 40(7), pp. 1–21.

[21] Laroche, C., and Vizika, O., 2005, “Two-Phase Flow Properties Prediction from Small-Scale Data Using Pore-Network Modeling,” Transp. Porous Media.

[22] Spiteri, E. J., Juanes, R., Blunt, M. J., and Orr, F. M., 2008, “A New Model of Trapping and Relative Permeability Hysteresis for All Wettability Characteristics,” SPE J., 13(3), pp. 277–288.

[23] Kuttanikkad, S. P., Prat, M., and Pauchet, J., 2011, “Pore-Network Simulations of Two-Phase Flow in a Thin Porous Layer of Mixed Wettability: Application to Water Transport in Gas Diffusion Layers of Proton Exchange Membrane Fuel Cells,” J. Power Sources, 196(3), pp. 1145–1155.

[24] Blunt, M. J., 1998, “Physically-Based Network Modeling of Multiphase Flow in Intermediate-Wet Porous Media,” J. Pet. Sci. Eng., 20(3–4), pp. 117–125.

[25] Van Der Linden, J., Tordesillas, A., and Narsilio, G., 2017, “Testing Occam’s Razor to Characterize High-Order Connectivity in Pore Networks of Granular Media: Feature Selection in Machine Learning,” EPJ Web Conf., 140, pp. 1–4.

[26] Chauhan, S., Rühaak, W., Khan, F., Enzmann, F., Mielke, P., Kersten, M., and Sass, I., 2016, “Processing of Rock Core Microtomography Images: Using Seven Different Machine Learning Algorithms.,” Comput. Geosci., 86, pp. 120–128.

[27] Van Der Linden, J. H., Narsilio, G. A., and Tordesillas, A., 2016, “Machine Learning Framework for Analysis of Transport through Complex Networks in Porous, Granular Media: A Focus on Permeability,” Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., 94(2), p. 1.

[28] Gholami, R., Moradzadeh, A., Maleki, S., Amiri, S., and Hanachi, J., 2014, “Applications of Artificial Intelligence Methods in Prediction of Permeability in Hydrocarbon Reservoirs.,” J. Pet. Sci. Eng., 122, pp. 643–656.

[29] Khanian, M., and Riahi, M. A., 2012, “Secondary Porosity Characterization in Carbonate Reservoirs and the Consequences in Permeability Forecasting,” Pet. Sci. Technol., 30(11),

pp. 1163–1175.

[30] Çengel, Y. A., and Cimbala, J. M., 2006, Fluid Mechanics: Fundamentals and Applications, McGraw-HillHigher Education.

[31] Semlak, K. A., and Rhines, F. N., 1958, "The Rate of Infiltration of Metals," Trans. TMS-AIME, 212, pp. 325–331.

[32] Kumar, V., Harris, C. K., Bronson, A., Shantha-Kumar, S., and Medina, A., 2016, "High-Temperature Liquid Metal Infusion Considering Surface Tension-Viscosity Dissipation," Metall. Mater. Trans. B Process Metall. Mater. Process. Sci., 47(1), pp. 108–115.

[33] Adams, B., Bohnhoff, W., and Dalbey, K., 2009, "DAKOTA, a Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis:," Sandia Natl. Lab., (December 2009), p. 1047.

[34] Sagiroglu, S., and Sinanc, D., 2013, "Big Data: A Review," 2013 International Conference on Collaboration Technologies and Systems (CTS), IEEE, pp. 42–47.

[35] IBM, Zikopoulos, P., and Eaton, C., 2011, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Osborne Media.

[36] Yang, X., Wu, C., Lu, K., Fang, L., Zhang, Y., Li, S., Guo, G., and Du, Y., 2017, "An Interface for Biomedical Big Data Processing on the Tianhe-2 Supercomputer," Molecules, 22(12), pp. 1–11.

[37] Ulfenborg, B., Karlsson, A., Riveiro, M., Ameen, C., Akesson, K., Andersson, C. X., Sartipy, P., and Synnergren, J., 2017, "A Data Analysis Framework for Biomedical Big Data: Application on Mesoderm Differentiation of Human Pluripotent Stem Cells.," PLoS One, 12(6), p. e0179613.

[38] Ye, Z., Tafti, A. P., He, K. Y., Wang, K., and He, M. M., 2016, "SparkText: Biomedical Text Mining on Big Data Framework.," PLoS One, 11(9), pp. 1–15.

[39] Navale, V., and Bourne, P. E., 2018, "Cloud Computing Applications for Biomedical Science: A Perspective," PLOS Comput. Biol., pp. 1–15.

[40] Ghofrani, F., He, Q., Goverde, R. M. P., and Liu, X., 2018, "Recent Applications of Big Data Analytics in Railway Transportation Systems: A Survey.," Transp. Res. Part C, 90, pp. 226–246.

[41] Xiao, X., and Zhou, C., 2017, "Application and Challenges of Big Data in Quality Monitoring of Highway Engineering.," AIP Conf. Proc., 1820(1), pp. 1–6.

[42] Wei, S., Yuan, J., Qiu, Y., Luan, X., Han, S., Zhou, W., and Xu, C., 2017, "Exploring the Potential of Open Big Data from Ticketing Websites to Characterize Travel Patterns within the Chinese High-Speed Rail System," PLoS One, 12(6), pp. 1–14.

[43] Govindan, K., Cheng, T. C. E., Mishra, N., and Shukla, N., 2018, "Big Data Analytics and Application for Logistics and Supply Chain Management.," Transp. Res. Part E, 114, pp. 343–349.

[44] Wu, K.-J., Liao, C.-J., Tseng, M.-L., Lim, M. K., Hu, J., and Tan, K., 2017, "Toward

Sustainability: Using Big Data to Explore the Decisive Attributes of Supply Chain Risks and Uncertainties.," J. Clean. Prod., 142, pp. 663–676.

[45] Bagheri, B., Siegel, D., Zhao, W., and Lee, J., 2015, "A Stochastic Asset Life Prediction Method for Large Fleet Datasets in Big Data Environment," Volume 14: Emerging Technologies; Safety Engineering and Risk Analysis; Materials: Genetics to Structures, p. V014T06A010.

[46] Xiang, F., Yin, Q., Wang, Z., and Jiang, G. Z., 2018, "Systematic Method for Big Manufacturing Data Integration and Sharing.," Int. J. Adv. Manuf. Technol., 94(9–12), pp. 3345–3358.

[47] Nakata, K., Orihara, R., Mizuoka, Y., and Takagi, K., 2017, "A Comprehensive Big-Data-Based Monitoring System for Yield Enhancement in Semiconductor Manufacturing.," IEEE Trans. Semicond. Manuf., 30(4), pp. 339–344.

[48] Liu, Z., Wang, Y., Cai, L., Cheng, Q., and Zhang, H., 2016, "Design and Manufacturing Model of Customized Hydrostatic Bearing System Based on Cloud and Big Data Technology," Int. J. Adv. Manuf. Technol., 84(1–4), pp. 261–273.

[49] Angelova, A., Krizhevsky, A., and Vanhoucke, V., 2015, "Pedestrian Detection with a Large-Field-Of-View Deep Network," Proceedings - IEEE International Conference on Robotics and Automation, pp. 704–711.

[50] Gonzalez-Dominguez, J., Lopez-Moreno, I., Moreno, P. J., and Gonzalez-Rodriguez, J., 2015, "Frame-by-Frame Language Identification in Short Utterances Using Deep Neural Networks," Neural Networks, 64, pp. 49–58.

[51] Saripalli, P., and McGrail, P., 2002, "Semi-Analytical Approaches to Modeling Deep Well Injection of CO2 for Geological Sequestration," Energy Convers. Manag., 43(2), pp. 185–198.

[52] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Li, F. F., 2014, "Large-Scale Video Classification with Convolutional Neural Networks," Proc. IEEE CVPR, pp. 1725–1732.

[53] Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., Legg, S., Mnih, V., Kavukcuoglu, K., and Silver, D., 2015, "Massively Parallel Methods for Deep Reinforcement Learning," arXiv:1507.04296, p. 14.

[54] Sutskever, I., Vinyals, O., and Le, Q. V, 2014, "Sequence to Sequence Learning with Neural Networks," Adv. Neural Inf. Process. Syst., pp. 3104–3112.

[55] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1–9.

[56] Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P., 2009, "Exploring Strategies for Training Deep Neural Networks," J. Mach. Learn. Res., 1, pp. 1–40.

[57] Lavin, A., and Gray, S., 2015, "Fast Algorithms for Convolutional Neural Networks."

[58] Le, Q. V, Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y., 2011, "Building High-Level Features Using Large Scale Unsupervised Learning," Int. Conf. Mach. Learn., p. 38115.

[59] Schapiro, A. C., Rogers, T. T., Cordova, N. I., Turk-Browne, N. B., and Botvinick, M. M., 2013, "Neural Representations of Events Arise from Temporal Community Structure," Nat. Neurosci., 16(4), pp. 486–492.

[60] Michalski, R. S., Bratko, I., and Bratko, A., eds., 1998, Machine Learning and Data Mining; Methods and Applications, John Wiley & Sons, Inc., New York, NY, USA.

[61] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V, Mao, M. Z., Ranzato, M. A., Senior, A., Tucker, P., Yang, K., and Ng, A. Y., "Large Scale Distributed Deep Networks," pp. 1–11.

[62] Cui, H., Zhang, H., Ganger, G. R., Gibbons, P. B., and Xing, E. P., "GeePS : Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server."

[63] Chen, T., Li, M., Cmu, U. W., Li, Y., Lin, M., Wang, N., Wang, M., Xu, B., Zhang, C., Zhang, Z., and Alberta, U., "MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," pp. 1–6.

[64] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viégas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., 2016, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," CoRR, abs/1603.0.

[65] Boussinesq, J., 1905, "JOURNAL DE MATHÉMATIQUES PURES ET APPLIQUÉES Calcul Du Pouvoir Refroidissant Des Fluides," J. Math., 1, pp. 285–332.

[66] Bishop, C., and Nasrabadi, N., 2006, "Pattern Recognition and Machine Learning," Pattern Recognit., 4(4), p. 738.

[67] Gostick, J. T., Ioannidis, M. A., and Fowler, M. W., 2007, "Pore Network Modeling of Fibrous Gas Diffusion Layers for Polymer Electrolyte Membrane Fuel Cells," J. Power Sources.

[68] Stallman, R. W., 1965, "Steady One-dimensional Fluid Flow in a Semi-infinite Porous Medium with Sinusoidal Surface Temperature," J. Geophys. Res.

[69] Fischer, G. J., and Paterson, M. S., 1992, "Measurement of Permeability and Storage Capacity in Rocks during Deformation at High Temperature and Pressure," Int. Geophys.

[70] Kelemen, P. B., and Whitehead, J. A., 1995, "Experiments on Flow Focusing in Soluble Porous Media, with Applications to Melt Extraction from the Mantle," J. Petrol., 100, pp. 475–496.

[71] Valvatne, P. H., and Blunt, M. J., 2004, "Predictive Pore-Scale Modeling of Two-Phase

Flow in Mixed Wet Media," Water Resour. Res., 40(7).

[72] Li, B., Rong, L., Li, Y., and Gjunter, V., 2000, "Synthesis of Porous Ni–Ti Shape-Memory Alloys by Self-Propagating High-Temperature Synthesis: Reaction Mechanism and Anisotropy in Pore Structure," Acta Mater.

[73] Sahimi, M., 1995, "Reply to 'Comment on Fractal and Superdiffusive Transport and Hydrodynamic Dispersion in Heterogeneous Porous Media,'" Transp. Porous Media.

[74] Bakke, S., and Øren, P. E., 1997, "3-D Pore-Scale Modelling of Sandstones and Flow Simulations in the Pore Networks," Spe J.

[75] Zhu, Y., Fox, P. J., and Morris, J. P., 1999, "A Pore-scale Numerical Model for Flow through Porous Media," Int. J. Numer.

[76] Mukherjee, P., Kang, Q., and Wang, C., 2011, "Pore-Scale Modeling of Two-Phase Transport in Polymer Electrolyte Fuel Cells—progress and Perspective," Energy Environ. Sci.

[77] Morris, J., Zhu, Y., and Fox, P., 1999, "Parallel Simulations of Pore-Scale Flow through Porous Media," Comput. Geotech.

[78] Shalf, J., Dosanjh, S., and Morrison, J., 2010, "Exascale Computing Technology Challenges," High Perform. Comput. ….

[79] Bergman, K., Borkar, S., and Campbell, D., 2008, "Exascale Computing Study: Technology Challenges in Achieving Exascale Systems," Off. (DARPA IPTO ….

[80] Heroux, M., Bartlett, R., and Howle, V., 2005, An Overview of the Trilinos Project.

[81] Edwards, H., and Trott, C., 2013, "Kokkos: Enabling Performance Portability across Manycore Architectures," 2013 Extrem. Scaling Work. (xsw.

[82] Christie, M., Demyanov, V., and Erbas, D., 2006, "Uncertainty Quantification for Porous Media Flows," J. Comput. Phys.

[83] Dostert, P., Efendiev, Y., and Hou, T., 2008, "Multiscale Finite Element Methods for Stochastic Porous Media Flow Equations and Application to Uncertainty Quantification," Comput. Methods Appl. Mech.

[84] Adams, B., Bohnhoff, W., and Dalbey, K., 2009, "DAKOTA, a Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis:," Sandia Natl.

[85] Akhmatskaya, E., Todd, B., and Daivis, P., 1997, "A Study of Viscosity Inhomogeneity in Porous Media," J.

[86] Riviere, B., 2008, Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation.

[87] Helmig, R., and Huber, R., 1998, "Comparison of Galerkin-Type Discretization Techniques for Two-Phase Flow in Heterogeneous Porous Media," Adv. Water Resour.

[88] Hou, T., and Wu, X., 1997, "A Multiscale Finite Element Method for Elliptic Problems in

Composite Materials and Porous Media," J. Comput. Phys.

[89] Ashesh Chattopadhyay, et al., 2016, "A Generic Hybrid-Parallel Programming Model for a Poro-Elastic Network Simulator Using the Trilinos-Kokkos Framework," Int. J. High Perform. Comput. Appl.

[90] Benzi, M., Golub, G., and Liesen, J., 2005, "Numerical Solution of Saddle Point Problems," Acta Numer.

[91] Kim, B., and Corradini, M. L., 1988, "Modeling of Small-Scale Single Droplet Fuel/Coolant Interactions," Nucl. Sci. Eng., 98(1), pp. 16–28.

[92] Ciccarelli, G., and Frost, D. L., 1994, "Fragmentation Mechanisms Based on Single Drop Steam Explosion Experiments Using Flash X-Ray Radiography," Nucl. Eng. Des., 146(1), pp. 109–132.

[93] Zhou, Y., Zhang, Z., Lin, M., Minghao, Y., and Xiao, Y., 2013, "Numerical Simulation of Fragmentation of Melt Drop Triggered by External Pressure Pulse in Vapor Explosions," Ann. Nucl. Energy, 57, pp. 92–99.

[94] Zhong, M., Li, Y., Lin, M., Yuan, M., and Yang, Y., 2014, "A Numerical Analysis Research on Earlier Behavior of Molten Droplet Covered with Vapor Film at the Stage of Triggering and Propagation in Steam Explosion," Sci. Technol. Nucl. Install., 2014.

[95] Thakre, S., Ma, W., and Li, L., 2013, "A Numerical Analysis on Hydrodynamic Deformation of Molten Droplets in a Water Pool," Ann. Nucl. Energy, 53, pp. 228–237.

[96] Shantha-Kumar, S., 2015, "Reaction of Liquid Aluminium- Samarium Alloys With B 4 C At Ultra High Temperatures," The University of Texas at El Paso.

[97] Grimson, M. J., Rickayzen, G., and Richmond, P., 1980, "Short Range Solvation Forces in Fluids," Mol. Phys., 39(1), pp. 61–73.

[98] Pertsin, A. J., and Grunze, M., 2003, "Erratum: Long-Ranged Solvation Forces in a Fluid with Short-Ranged Interactions (Journal of Chemical Physics (2003) 118 (8004))," J. Chem. Phys., 119(17), p. 9322.

[99] Karanikas, S., Dzubiella, J., Moncho-Jordá, A., and Louis, A. A., 2008, "Density Profiles and Solvation Forces for a Yukawa Fluid in a Slit Pore," J. Chem. Phys., 128(20).

[100] Bobra, M., "By EE-130 May 1992," (May 1992).

[101] Chang, R., 2010, "Chemistry," Chemistry (Easton)., pp. 1062–1084.

[102] Tsuji, T., Jiang, F., and Christensen, K. T., 2016, "Characterization of Immiscible Fluid Displacement Processes with Various Capillary Numbers and Viscosity Ratios in 3D Natural Sandstone," Adv. Water Resour., 95(Supplement C), pp. 3–15.

[103] T.Dorothy, 2014, "A Circuit-Based Algorithm for Two-Phase Flow at the Pore Scale in Porous Media."

[104] Thesis, M., Physics, R., and Folles, N., 2012, "Fluid Displacements during Multiphase Flow Visualized at the Pore Scale Using Micromodels," (June).

[105] Chassagne, R., Dörfler, F., Guyenot, M., and Harting, J., “Modeling of Capillary-Driven Flows in Axisymmetric Geometries,” pp. 1–24.

[106] Hibbeler, R. C., 2016, Dynamics.

[107] Homsy, G. M., 1987, “Viscous Fingering in Porous Media,” Annu. Rev. Fluid Mech., 19(1), pp. 271–311.

[108] Cuthiell, D., Kissel, G., Jackson, C., Frauenfeld, T., Fisher, D., and Rispler, K., “Viscous Fingering Effects in Solvent Displacement of Heavy Oil.”

[109] Zhang, C., Oostrom, M., Wietsma, T. W., Grate, J. W., and Warner, M. G., 2011, “Influence of Viscous and Capillary Forces on Immiscible Fluid Displacement : Pore-Scale Experimental Study in a Water-Wet Micromodel Demonstrating Viscous and Capillary Fingering,” pp. 3493–3505.

[110] Nelson, P. H., 2009, “Pore-Throat Sizes in Sandstones , Tight Sandstones , and Shales,” 3(3), pp. 329–340.

[111] Buckley, J. S., Edwards, J., and Fordham, E., “Fundamentals of Wettability,” pp. 44–61.

[112] Orloff, J., and Bloom, J., “Variance of Discrete Random Variables.”

CURRICULUM VITA

Arturo Schiaffino Bustamante was born in Cuauhtémoc, Chihuahua, Mexico. In 2008, he won a national science and prototyping competition held at Monterrey, Nuevo Leon, Mexico and participated as a Mexican Delegate in the 24th Mostratec in Novo Hamburgo, Brazil, the second largest science fair in the world.

He was admitted into the University of Texas at El Paso, where he got involved in several academic, professional and extracurricular activities, obtaining a Bachelor's in Science in Mechanical Engineering with a minor in Mathematical Sciences in May of 2016. After receiving is Bachelor's degree, Arturo joined the graduate program at UTEP, where he held the positions of research assistant, teaching assistant and academic advisor trough the length of his graduate studies. During his graduate studies, he worked with in several research projects and collaborated with Instituto Tecnologico de Monterrey (ITESM) as computational fluid dynamics instructor and a research contributor.

Arturo received an offer letter from Freeport-McMoRan Inc after the conclusion of his Master's degree. He accepted the offer, and currently works as a mechanical engineer at Morenci, Arizona.

Contact Information: aschiaffino9228@gmail.com

This thesis/dissertation was typed by: Arturo Schiaffino Bustamante