2020-01-01

Using Computational Fluid Dynamics And Machine Learning To Predict Sled Profile During High Speed Water Braking At Holloman High Speed Test Track

Jose A. Terrazas
University of Texas at El Paso

USING COMPUTATIONAL FLUID DYNAMICS AND MACHINE LEARNING TO PREDICT SLED PROFILE DURING HIGH SPEED WATER BRAKING AT HOLLOMAN HIGH SPEED TEST TRACK

JOSE ARMANDO TERRAZAS

Master's Program in Computational Science

APPROVED:

Vinod Kumar, Ph.D., Chair

Robert Edmonds, Ph.D., Co-Chair

V.M. Krushnarao Kotteda, Ph.D.

Jorge Munoz, Ph.D.

Art Bronson, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

USING COMPUTATIONAL FLUID DYNAMICS AND MACHINE LEARNING TO PREDICT SLED PROFILE DURING HIGH SPEED WATER BRAKING AT HOLLOMAN HIGH SPEED TEST TRACK

by

JOSE ARMANDO TERRAZAS, MSSE

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

Computational Science Program

THE UNIVERSITY OF TEXAS AT EL PASO

August 2020

Acknowledgements

The work in this thesis was supported by the U.S. Air Force through the Summer Faculty Fellow Program, Air Force Office of Scientific Research (AFOSR). We would like to acknowledge the U.S. Department of Defense (AFOSR Grant Number # FA9550-19-1-0304, FA9550-17-1-0253, FA9550-12-1-0242, FA9550-17-1-0393, SFFP, AFTC, HAFB/HSTT, AFRL, HPCMP), Department of Energy (DE-FE0026220, DE-FE0002407, NETL, Sandia, ORNL, NREL), Systems Plus, and several other individuals at these agencies for partially supporting our research financially or through mentorship. We would also like to thank NSF ((HRD-1139929, XSEDE Award Number ACI-1053575), TACC, DOE, DOD HPCMP, University of Texas STAR program, UTEP (Research Cloud, Department of Mechanical Engineering, Graduate School & College of Engineering) for generously providing financial support or computational resources. Without their generous support it would have been almost impossible to complete the milestones.

I would also like to thank Dr. Kumar, Dr. Munoz, Dr. Kotteda, Dr. Edmonds, and Dr. Gudimetla for their constant patience, support, and mentorship throughout my study, this work would not have been possible without them.

Abstract

Accurate prediction of a rocket sled test profile and water braking phenomena has potential to result in radical changes in designs of specific sleds and provide greater confidence of braking mechanism and recovery of critical infrastructures. Understanding the water's behavior with the sled is critical to predicting how the water could damage the sled and affect recoverability of the sled and determine success of the missions. Traditionally, sled design for the test missions had been guided by empirical/hand calculations to estimate the forces on various components. The calculations involved various approximations to arrive at the force balance law and predict the acceleration/deceleration profile. In partnership with the Holloman High Speed Test Track (HHSTT), we performed preliminary simulations to develop a predictive model for the HHSTT sled tests for various velocity regimes. The CFD results from various geometry configurations for the sled and modeling parameters will be presented. The main goals of the CFD investigations are to improve the accuracy of the predicted profile that often depends on the complexity of the design and operating conditions.

Having an atmospheric turbulence deconvolution model would be of great utility for the Air Force in its mission to maintain space situational awareness. The purpose or goal of this study is to increase US Air Force capability in space situational awareness, through the construction of a deconvolution model that, once trained, can remove turbulent effects and provide clearer images in a small amount of time (expedient), with relatively good accuracy (effective), and with a relatively low computational resource requirement (efficient). To develop a deconvolution model a Conditional Generative Adversarial Network (cGAN) will be used. The cGAN will make use of two Neural Nets (NN) that will be pitted against each other, one will be an image generator and the other an image discriminator. The Generator will be given turbulent

images and will generate turbulent images while converging towards a 'pristine (non-turbulent)' image as it attempts to 'trick' the Discriminator. The discriminator will be given pristine, blurred, and generated (fake) images, its goal will be to distinguish whether images are generated or from the pristine dataset.

By combining CFD with machine learning, cGANs specifically, we can train a net to produce CFD results, eliminating time and resources required to run CFD models.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Advances, accessibility, and increasing computational capability in the past few years has made it possible to model and solve, within small error, complex system of equations that approximate fluid flows with enough accuracy to make the results useful for engineering analysis. These complex systems of equations are known as Navier – Stokes Equations and are used to imitate a wind tunnel and fluid phenomena in a virtual space. The development and solving of these virtual models are what Computational Fluid Dynamics (CFD) is about. By using CFD engineers and scientist can analyze and study fluids in a quick, cost effective, non-intrusive, and feasible manner compared to running real test and experiments. To this end, CFD modeling can be a great tool for the US Air Force (USAF) at HHSTT to engineer, built, and predict pusher sleds braking profiles. By designing and building better sleds, testing capabilities at HHSTT can be improved. Improvement can be in better braking performance and reusability which will decrease the cost of testing to the track and increase the speeds at which items can be tested as braking capability is improved.

Just as computational capability and advances have enabled CFD, they have also made Machine Learning (ML) through Neural Networks (NN) possible. Machine Learning enables statistics to learn how to perform a specific task independently. It has characteristics in several different fields, including artificial intelligence, statistics, and computer science [29].

By combining CFD and ML, studied in this paper, it is hoped that USAF capability to engineer, test, and analyze testing models at HHSTT will be drastically improved. In this study, we will review CFD of water braking of sleds in Holloman High Speed Test Track (HHSTT) and Conditional Generative Adversarial Net (cGAN) used to deconvolute images and discuss the

results of both. We will then conclude with a discussion of possible integration of both tools to improve HHSTT ability to model and predict test profiles for water braking.

Chapter 2: Technical Background

To successfully complete and analyze CFD models in this study we must understand the technical components that make it up. Since we will be braking with water, we will review fluid flows and the equations governed by them. In this section we will go over the technical components required and used to successfully obtain CFD results in the scope of our study.

2.1 Computational Fluid Dynamics (CFD)

By following a group of partial differential equations, it is possible to model fluids virtually. Before CFD was possible, engineers and scientist had to do testing and studies in specially designed testing set ups such as wind tunnels. Although relatively expensive testing set ups are still done, CFD offers many benefits over real world testing. A list of benefits include cost, turnaround time, exact similarity, non-intrusive measuring, analysis, feasibility, and extendibility. A list of disadvantages includes turbulence, accuracy, and error control.

Building, operating, and maintaining a test lab can be expensive, there are so many variables to consider. Size of object being tested, simulating testing conditions, construction of actual test object, number of tests being performed, and staff, just to name a few. CFD provides an opportunity, if done right, to mitigate costs as unique tests with variable conditions can be easily altered with comparatively little cost, time, and effort. Also, to gather data on real world tests, sensors must be used. These sensors can perturb or intrude flows and alter results not to mention that data may only be collected at sensor location, in this too CFD has an advantage. With CFD you do not need intrusive sensors and data can be collected on any point of the testing item not just on selected points which can lead to better analysis and understanding of fluid behavior or impact. Another advantage is that conditions that may not be measured experimentally may be measured virtually, for example gas effects on a missile during reentry.

All the benefits however are not without disadvantages and experimental testing is sometimes still necessary. Turbulence models in 3-D for example involve very small flow features at high Reynolds numbers that even with today's super computers are still impossible to simulate all the details [28]. This in turn, depending on tolerance for error, can make CFD too inaccurate. Errors incurred from the insufficient detail of turbulence models can make the model's accuracy unquantifiable.

Despite the drawbacks CFD is still a key technology that when used right can offer substantial value from its advantages. For some flows CFD will only be able to provide qualitative results but they may provide some insight that may be used for a better experimental setup. The components needed for a CFD solution are a physical model, a mathematical model, a numerical method, the mesh, and pre- and post-processing tools [28].

2.1.1 Physical Model

The physical model is the most basic component to CFD, it is the geometry of the test object and where the space or environment in which it will be modeled is defined. The physical model does require assumptions to make the model simpler and solvable, this includes boundary conditions, continuum assumption, the type of flow description, and conservation principles.

When building a fluid dynamics model, we are usually not concerned with tracking individual particles of molecular or atomic size. The particles are too small, the number of particles to track would be unnecessarily large, and the impact individual particles would have on a body in terms of lift and drag in our study would be negligibly minor. For this reason, an assumption made in this study is the continuum assumption. The effect of this assumption is that we consider the statistical average of the particles bulk flow [28].

When analyzing dynamics there are two ways to do describe them, Eulerian and Lagrangian. In this study we will do it from the Eulerian viewpoint as it allows for the use of a control volume in which conservation of momentum principles apply.

Building on continuum and control volume principles we consider the balance of transported quantities in that control volume. It is then possible to define and analyze at each location in space how much mass, momentum, and energy is entering and exiting each space.

2.1.2 Mathematical Model

Once assumptions and conservation of momentum and mass are established from the Physical Model, mathematical equations based on them can be derived. To examine the conservation of momentum equations let us begin with an example in 1 dimension by assuming constant flow in y and z. By assuming constant flow in y and z we are left with variations in the x direction only. We will derive the equations for mass conservation in 1-D x-coordinate and then apply it to y and z cartesian coordinates, see figure 2.1. First, we will use Taylor's expansion

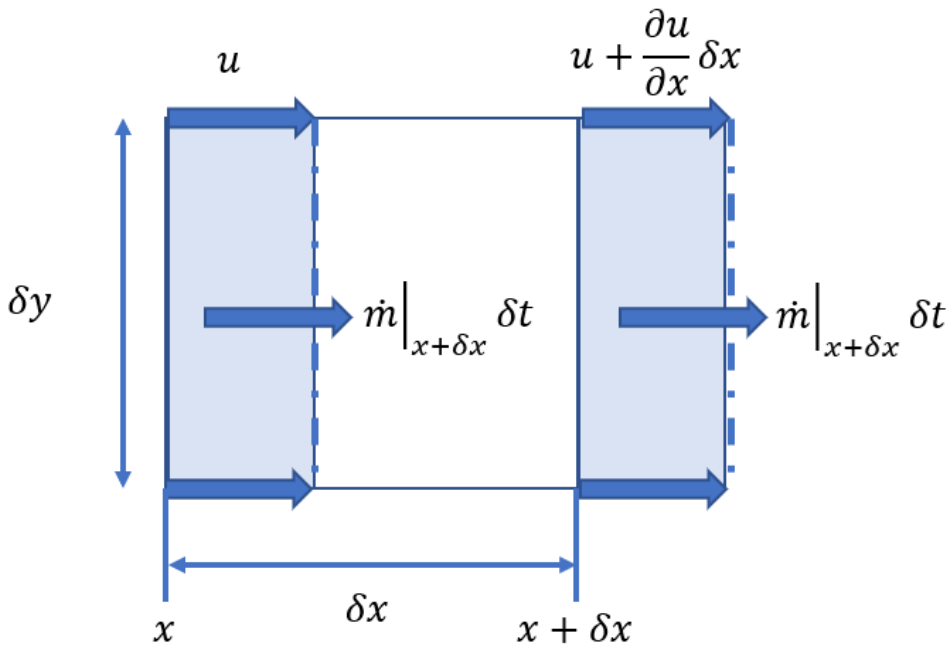


Figure 2.1 Conversation of mass on 1-dimensional fluid element

to approximate the mass flux $(pu)|_{x+\delta x}$ at the right edge of the element $x + \delta x$ in terms of mass flux $(pu)|_x$ at x:

$$(pu)|_{x+\delta x} = (pu)|_x + \frac{\mathrm{d}(\rho u)}{\mathrm{d}x}\delta x + \frac{1}{2}\frac{\mathrm{d}^2(\rho u)}{\mathrm{d}x^2}\delta x^2 + O(\delta x^3) \tag{2.1}$$

From figure 2.2.2.1 we can then see that the fluid entering on the left-hand side at x is in time δt is

$$\dot{m}|_x \delta t = (pu)|_x \delta y \delta z \delta t \tag{2.2}$$

And the amount of fluid exiting on the right-hand side $x + \delta x$ in the same time interval δt is

$$\dot{m}|_{x+\delta x} \delta t = (pu)|_{x+\delta x} \delta y \delta z \delta t \tag{2.3}$$

From these equations we can then get change in mass in the control volume in time δt by subtracting fluid entering $\dot{m}|_x \delta t$ from $\dot{m}|_{x+\delta x} \delta t$ and multiplying it by the time interval δt.

$$\delta m = \dot{m}|_x \delta t \; - \; \dot{m}|_{x+\delta x} \delta t$$

$$\delta m = (pu)|_x \delta y \delta z \delta t \; - \; (pu)|_{x+\delta x} \delta y \delta z \delta t \tag{2.4}$$

Then using Taylor expansion for pu we obtain

$$\delta m = ((\rho u)|_x) \delta y \delta z \delta t \; - \left((\rho u)|_x + \frac{\mathrm{d}(\rho u)}{\mathrm{d}x} \delta x + \frac{1}{2} \frac{\mathrm{d}^2(\rho u)}{\mathrm{d}x^2} \delta x^2 + O(\delta x^3) \right) \delta y \delta z \delta t$$

Which simplifies to

$$\delta m = \; -\left(\frac{\mathrm{d}(\rho u)}{\mathrm{d}x} \delta x + \frac{1}{2} \frac{\mathrm{d}^2(\rho u)}{\mathrm{d}x^2} \delta x^2 + O(\delta x^3) \right) \delta y \delta z \delta t$$

Divide by size of control volume $\delta x \delta y \delta z$ and time interval δt to end up with

$$\frac{\delta \mathrm{m}}{\delta \mathrm{x} \, \partial \mathrm{y} \delta \mathrm{z} \delta \mathrm{t}} = \; \frac{\delta \rho}{\delta \mathrm{t}} = \; - \; \frac{\mathrm{d}(\rho \mathrm{u})}{\mathrm{dx}} \; - \; \frac{1}{2} \frac{\mathrm{d}^2(\rho \mathrm{u})}{\mathrm{dx}^2} \delta \mathrm{x} + \mathrm{O}(\delta \mathrm{x}^2) \tag{2.5}$$

The mass flow gradient $\frac{\mathrm{d}(\rho \mathrm{u})}{\mathrm{dx}}$ can be relatively large but if we make the size of the element δx sufficiently small, lim δx → 0 then δx ≪ 1 and the second and third term on the right side of (2.5) become negligible. We are then left with

$$\frac{\delta\rho}{\delta t} = -\frac{d(\rho u)}{dx} = -\rho\frac{du}{dx} - u\frac{d\rho}{dx} \tag{2.6}$$

Because we will be braking with water we take the case of incompressible flow in which density is constant, $\rho = \text{constant}$, and therefore cannot vary so $\frac{d\rho}{dx} = 0$. This means that we end up with

$$\frac{\delta\rho}{\delta t} = -\rho\frac{du}{dx} \tag{2.7}$$

And if there is no variation in y and z then velocity can not vary so $\frac{du}{dx} = 0$ so

$$\frac{\delta\rho}{\delta t} = 0 \tag{2.8}$$

This becomes important when looking at velocity variations when we couple velocity components in x, y, and z in a continuity equation.

In 3-dimensional mass conservation we will start with (2.4) but in all cartesian coordinates.

$$\delta m = \dot{m}|_x \delta t - \dot{m}|_{x+\delta x}\delta t + \delta m = \dot{m}|_y \delta t - \dot{m}|_{y+\delta y}\delta t + \delta m = \dot{m}|_z \delta t - \dot{m}|_{z+\delta z}\delta t \tag{2.9}$$

As a result, we end up with

$$\delta m = -\frac{\mathrm{d}(\rho u)}{\mathrm{d}x}\delta x\delta y\delta z\delta t \; -\frac{\mathrm{d}(\rho v)}{\mathrm{d}y}\delta x\delta y\delta z\delta t \; -\frac{\mathrm{d}(\rho w)}{\mathrm{d}z}\delta x\delta y\delta z\delta t \tag{2.10}$$

Dividing by the control volume and small enough time steps, lim δt → 0, produces

$$\lim_{\delta t\to 0}\frac{\delta m}{\delta x\,\partial y\delta z\delta t} = \lim_{\delta t\to 0}\frac{\delta \rho}{\delta t} = \frac{\partial \rho}{\partial t} \tag{2.11}$$

By assuming continuity and therefore infinite control volume in 3-D we obtain

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho \omega)}{\partial z} = 0 \tag{2.12}$$

Understanding that $\frac{\partial \rho}{\partial t} = 0$ from our one-dimensional analysis, we end up with

$$\frac{\partial(u)}{\partial x} + \frac{\partial(v)}{\partial y} + \frac{\partial(\omega)}{\partial z} = 0 \tag{2.13}$$

For the incompressible continuity equation.

As we cannot have more mass into a control volume once full, the inflow must balance with the outflow. If flow then slows in the x direction the flow entering at x must flow out the y and z directions. This occurance is the divergence constraint and (2.13) is the divergence of the velocity field that will be coupled in a system of equations that will track fluid flow computationally. In (2.10) we can see the conservation of mass, the difference between what is going in and what is going out must be inside the control volume.

For the derivation of the momentum equations of a fuild, force of moving fluid must be balanced. Because force is a vector and we have three coordinates, we will have three momentum equations. The momentum equations will be derived from Newton's second law F=ma. Using a Eulerian viewpoint established at the beginning of the section we can consider

$$\frac{\mathrm{d}(m\vec{v})}{\mathrm{d}t} = \vec{F} \tag{2.14}$$

For all of the three coordinate directions.

Because we are working with high speed turbulent flow we assume high Reynold's Number and neglect viscous shear forces. This means that fluid elements will be able to slide past each other and we can focus on pressure forces and gravity when deriving the equations. Recalling that pressure is a scalar and only acquires direction when acting on a surface, positive pressure will be compressive and pressure on face x will result in a positive force on the x-direction. This means that pressure on the $x + \delta x$ face will be negative, see figure 2.2 to view pressure forces on the faces of a fluid element. The difference in the forces acting on each other will provide us with the net force on the element.

$$F_{p,x} = (p|_x - p|_{x+\delta x})\delta y \delta z \tag{2.15}$$

Using Taylor expansion like in conservation of mass, but this time just in the first order or linear term we get

$$= \left(p|_x - \left(p|_x + \frac{\partial \mathrm{p}}{\partial x}\Big|_x \delta x\right)\right)\delta y \delta z$$

$$= -\frac{\partial_p}{\partial x}\delta x \delta y \delta z = -\frac{\partial_p}{\partial x}\delta V \tag{2.16}$$

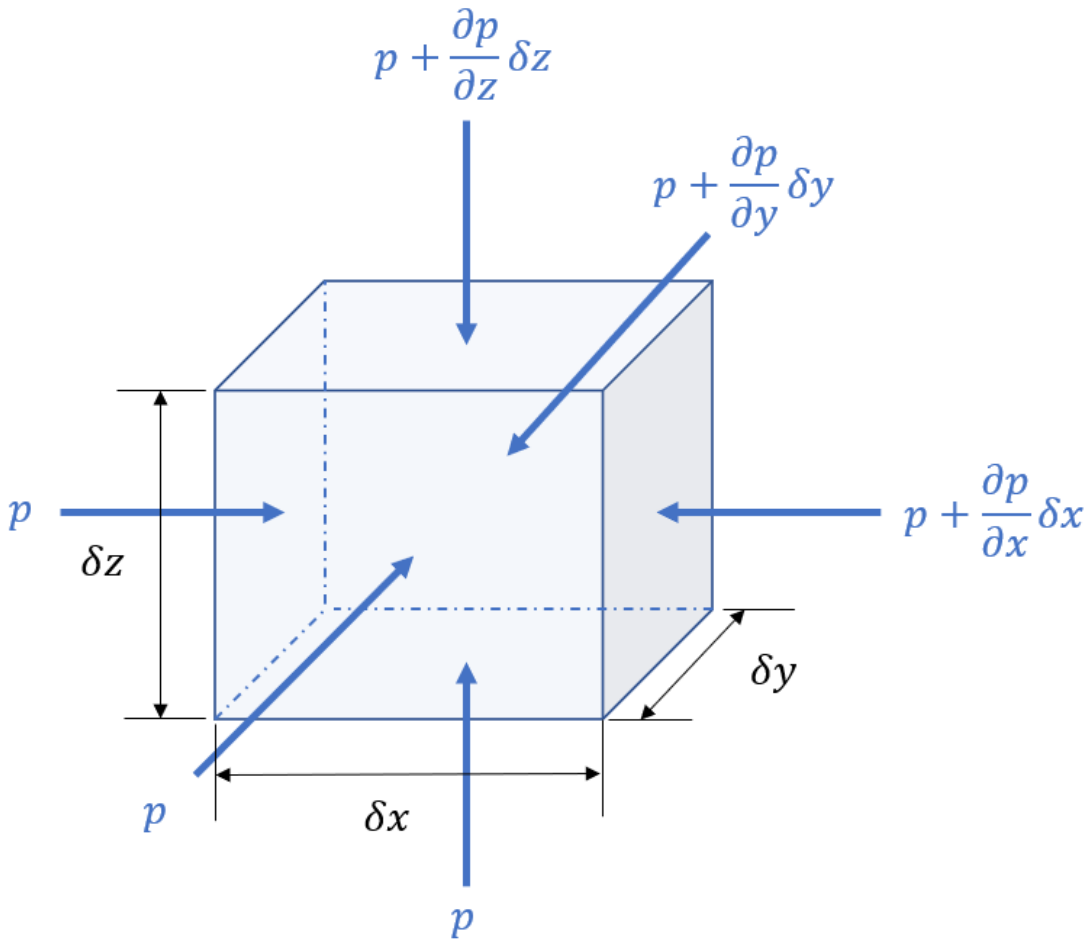


Figure 2.2 Pressure forces on 3-dimensional fluid element

Where δV is the size of the control volume $\delta x \delta y \delta z$. In the same manner as we just derived the Force component in the x direction, we derive the force on the y and z directions.

$$F_{p,y} = -\frac{\partial_p}{\partial y}\delta V \qquad (2.17)$$

$$F_{p,z} = -\frac{\partial_p}{\partial z}\delta V \qquad (2.18)$$

In vector notation the pressure force vector is

$$F_p = \begin{bmatrix} F_{p,x} \\ F_{p,y} \\ F_{p,z} \end{bmatrix}$$

Taking into consideration gravity in the -z direction and the sum of all body forces we end up with

x-direction

$$p\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial \mathrm{y}} + w\frac{\partial u}{\partial \mathrm{z}}\right) = \; -\frac{\partial_p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{2.19}$$

y-direction

$$p\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial \mathrm{y}} + w\frac{\partial v}{\partial \mathrm{z}}\right) = \; -\frac{\partial_p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \tag{2.20}$$

z-direction

$$p\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial \mathrm{y}} + w\frac{\partial w}{\partial \mathrm{z}}\right) = \; -\frac{\partial_p}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) - \rho g \tag{2.21}$$

One more equation to consider is energy balance but it is not always required. For compressible flows yes but inviscid incompressible flows, which is what was used in the study, it is not needed. For those type of flows if it satisfies the momentum equation it will also respect Bernoulli's equation [28].

2.1.3 Numerical Method and Discretization

Having establish the mathematical model for fluid dynamics we now need to identify a numerical model or process by which to solve the system of partial differential equations (PDE) of conservation of mass, momentum, and energy.

Having determined the control volume in which the simulation will take place, we need to establish exactly how the discretization of the volume space will take place. The discretization translates the mass equations, or continuous equations, to the computer using a discrete set of points in the control volume. This discretization, linked to the mesh, is what defines the connectivity of all these points. The mesh and points form the elements in the control volume in which mass, momentum, and energy of fluids are described may then be calculated when provided with initial conditions. An important thing to keep in mind is poor discretization, or mesh, can lead to numerical instability and may cause the model to blow up. For this reason, there are two types of meshes that we must consider, structured and unstructured. While each has its advantages, in general, structured is good for 'non-complex' geometry and unstructured is good with geometry with arcs. The type of mesh selected will impact the solving of the PDEs and can lead to faster or slower CFD run time.

Because the scope consists of arcs we will be using unstructured mesh and the Finite Volume Method (FVM) which is used for unstructured mesh. To mitigate the possibility of numerical instability, and because we are dealing with high speed flows, a small-time step of 1e-6 seconds or less must be used. Time-step, δt, is an initial condition required for solving Navier Stokes.

2.1.4 Post-Processing Analysis

Once the software finishes running the model, we must analyze our results and determine if they are appropriate, do they fall within expectations. For post-processing we can produce contours that provide information of the entire control volume. This information includes velocity vectors, volume fraction, pressure magnitudes, and more.

2.2 MACHINE LEARNING

As computational power has increased, their utility has increased. With the increase in computational capability, tools and methods have emerged to take advantage of this rise. Among these is Machine Learning (ML). Machine learning is the science of a computer so that it may learn from data [29]. By harnessing computational power and coupling with the capability to learn from data, tasks can become automated and productivity increased.

Uses of ML can range from email spam filters, language translation, voice recognition, to image recognition, to more advanced uses that can take advantage of unsupervised learning. Amongst ML practitioners a recent framework has emerged, Generative Adversarial Networks (GANs). To use GANs we need to learn about the algorithms that comprise it, neural networks.

2.2.1 Neural Networks

Neural nets (NN) are inspired from our brain's architecture. In a similar fashion to birds inspiring flight through planes and planes achieving flight differently to birds, NN have become a bit different that its biological inspiration. It is in this difference however that they become versatile, powerful, and scalable with the capability for large and complex tasks such as classifying billions of images, speech recognition, video recommendation on YouTube, and even beating the best Chess players in the world.

Neural nets are composed of nodes and layers of nodes that are connected to each other. From layer to layer they then create weights from inputs being provided. Using inputs and an activation function, NN are able to produce outputs. These activation functions are usually either a sigmoid function or a Rectified Linear Unit (ReLu), see Figure 2.3. When the activation function

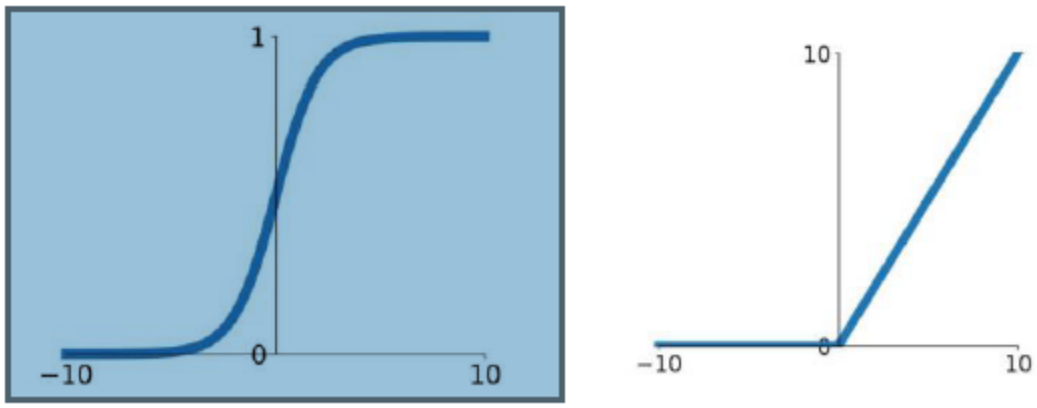


Figure 2.3 – Sigmoid function (left) and ReLu Function (right)

is calculated output is determined, as denoted by h in Figure 2.4. In the equation below, (2.22) we can also see the weight denoted by w for each node. Figure 2.4 represents the NN node for the equation

$$\mathrm{h}_1 = f\left(x_1 w, +\chi_2 \omega_2 + x_{B_3^{\omega}} + b\right) \tag{2.22}$$

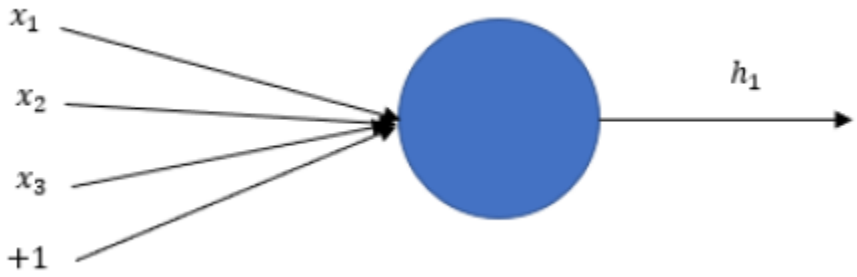


Figure 2.4 – Neural Net Node

Now that a NN node, neuron, has been established let us look at a simple NN and break down the components. A simple NN consists of an input layer, hidden layer, and output layer, see Figure 2.5.

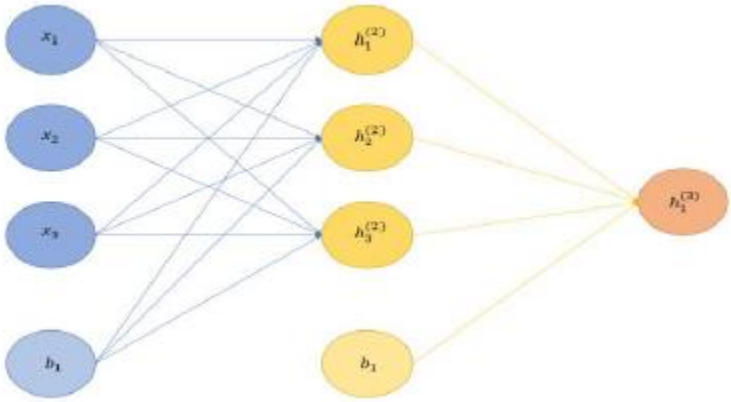


Figure 2.5 – Input layer (blue), Hidden layer (yellow), Output layer (red)

The process by which NN learn is called feedforward. Hidden layer is determined by providing input and output data, determining the hidden layer weights is how the NN learns.

2.2.2 Convolutional Neural Network

Convolutional Neural Networks (CNN) emerged by studying the brain's visual cortex and have been used in a limited capacity since the 1980s [29]. A big limiter of CNNs in the 80s was the absence of today's computational power and data. By studying the brain scientist have come to understand to a certain degree how the human brain processes visual, auditory, and other inputs from our eyes, ears, and other sensors into images, sounds, other information. A CNN is a specific type of NN.

Through experimental studies during the late 1950s on cats and monkeys, crucial insights of the structure of the visual cortex were derived. The authors of the experiments discovered how neurons in the visual cortex of the brain reacted to visual stimuli located on a small area of the visual field. They noticed that patterns in neurons; some react to horizontal lines, others to different line orientation, some have larger receptive fields, some react to complex patterns that are a

combination of other patterns [29]. Those studies inspired neocognition which eventually evolved into convolutional neural networks.

CNNs consist of two types of layers, Convolutional and Pool. Typical CNN architectures are conceptually simple. They have a few convolutional layers, then a pooling layer, then another few convolutional layers, followed by another pooling layer, and so forth. See Figure 2.6 for a visual representation.

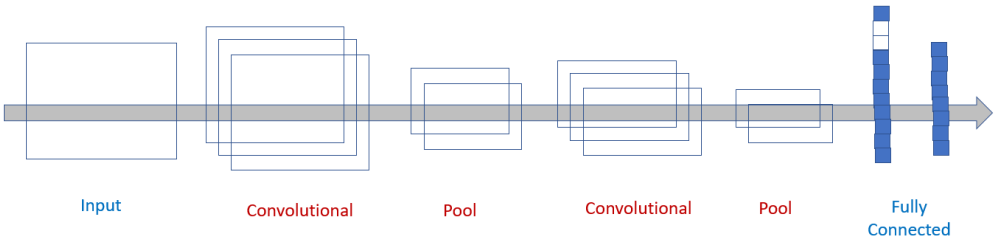


Figure 2.6 - Typical CNN Architecture

The convolutional layers are the most important building block of any CNN. CNNs consist of multiple layers that work in collaboration to improve image recognition. See Figure 2.7.

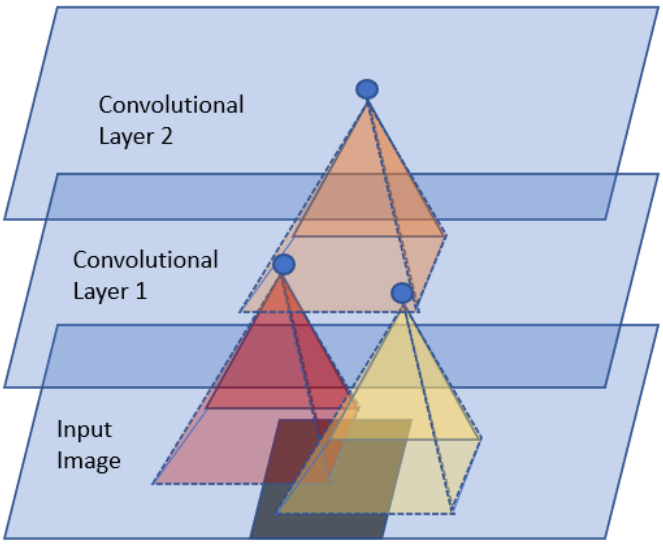


Figure 2.7 – Convolutional Layers

Neurons in the first convolutional layer are only connected to their receptive fields and not every pixel in the input image [29]. The second layer on the otherhand is connected only to neurons in a small rectangle of the first. This design allows the neural network to focus on specific low level features in the first layer and then assemble them into higher level features in the next layer, and so forth. This structure is what allows CNNs to work really well for image recognition.

The weight of a neuron can be represented by a small image the size of a receptive field. Figure 2.8 consists of two possible sets of weights, vertical and horizontal. They are also called filters or convolutional kernels.

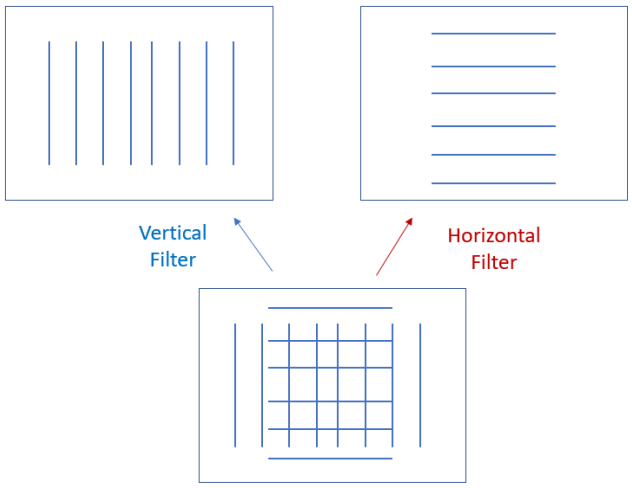


Figure 2.8 – Filters

What the neurons in the filters do is ignore everything in their receptive field (in this case the entire input image) except for the ones they were assigned to (vertical or horizontal). If all neurons are assigned to the vertical filter for example, the outcome would be the top left figure on Figure 2.8. Vice-versa if all are applied to the horizontal filter the outcome would be the top right figure on Figure 2.8.

Having a layer full of neurons using the same filter provides a feature map. A feature map highlights the areas in the image that are very similar to the filter. When a CNN is being trained what the CNN does is to find the most useful filters for its job and combines them into complex patterns. An example would be a cross would be where the vertical and horizontal filter are active.

To understand the concept of convolutional layers only 2D layers have been presented thus far. However, convolutional layers are composed of multiple feature maps of equal sizes [29]. Looking at Figure 2.9 it can be observed that they are more accurately represented in 3D.

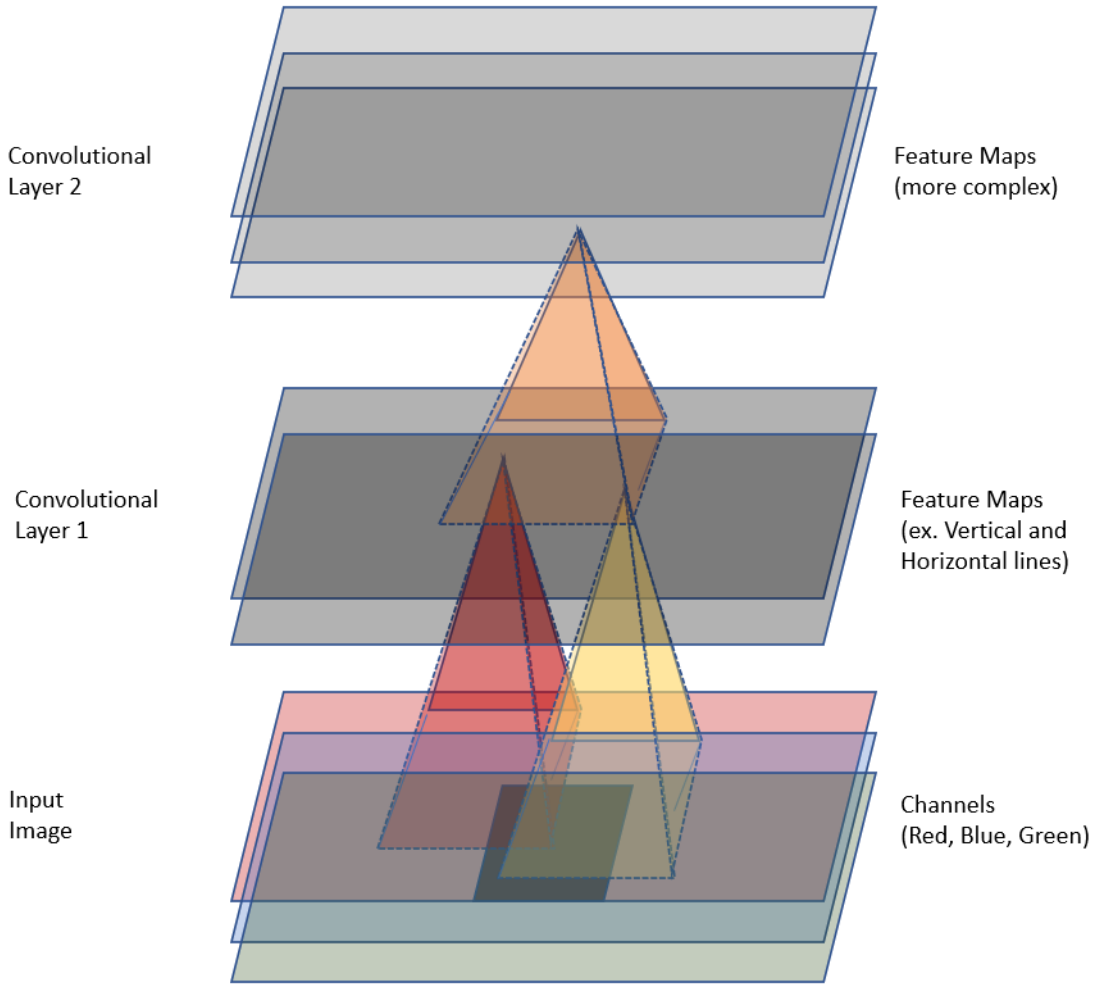


Figure 2.9 – Multiple Feature Maps

Within a single feature map, all neurons share the same parameters, weights, or bias term. When stacked with different feature maps the convolutional layer end up having different parameters Neuron's receptive field in the layer extends to all previous feature maps [29]. Due to the stacking of multiple feature maps a convolutional layer can simultaneously apply multiple filters to its inputs and therefore making it able to detect multiple features anywhere in the input provided. Furthermore, input images are composed of multiple sublayers. Examples include color channels (red, green, blue or RGB) and satellite images capturing different light frequencies.

A problem with CNNs hinted earlier as the amount of computing power required lies a memory problem. Convolutional layers require a large amount of Readily Available Memory (RAM). This is especially true during the training of the CNN "because the reverse pass of backpropagation requires all the intermediate values computed during a forward pass [29]. When making a prediction for a new instance the RAM occupied by a layer can be released as soon as the next layer is computed. You only need as much RAM as needed by two consecutive layers. The problem is when the CNN needs to be trained, when everything computed during the forward pass needs to be preserved for the reverse pass. The minimum amount of RAM needed is the total amount of RAM needed by all layers. A way to mitigate the memory problem is to reduce the mini-batch size or try removing a few layers. Also, the stride, distance between two consecutive receptive fields, can be reduced or the bit size of the floats, numbers, can be reduced from 32 to 16. Alternately all can be done.

The pooling layer is the second and final common building block of CNN, the first one being convolutional layer. The goal of the pooling layer is to subsample the input image in order to reduce the computational load, memory used, and number of parameters. Pooling layers can be

quite destructive as even taking a 2x2 kernel with a stride of 2 can result in a loss of 75% of input values.

Like in convolutional layers, neurons in pooling layers are connected to outputs of a limited number of neurons from the previous layer. Unlike a convolutional layer however, it has no weights. All it does is aggregate inputs using an aggregate function, like max or mean. For max only max input values in each kernel make it to the next layer.

Many variants of this architecture have been developed over the years, with increasingly reducing error rates. The advantage of CNN is that all neurons in a feature map sharing the same parameters drastically reduces the number of parameters in the model. This means that for the CNN once it has been trained and it has learned how to recognize a pattern in one location, it can recognize it in any other location. By contrast the DNN can only recognize a pattern in the location where it recognized it.

CNNs are great tool to take advantage of machines computing power and will only become more useful as computing power increase. They will play a big part of future technologies in automation of many systems such as self-driving cars, language recognition, even customer service as it recognizes search patterns and preferences of customers. Although they currently there are a few nuances such as requiring many hyperparameters (number of filters, their height and weight, etc.) and requiring substantial amounts of RAM, their architecture and building blocks are simple to understand. Like any recent technology however it will develop and improve as tools are designed to make it easier to design and use. New CNN architectures are being developed every year and the field is moving very rapidly.

2.2.3 Conditional Generative Adversarial Network

A GAN is a system of two NN, in our case CNN for computer vision. The two NN are the Discriminator and the other the Generator. The objective with having two of them is that one will be trained to recognize 'real' data and the other will generate data and try to trick the Discriminator, or the one trained to recognize real data. In a cGAN the Generator is also trained from images. The Generator will be generating fake data and through backpropagation eventually receive enough feedback to be able to trick the Discriminator. Figure 2.10 describes the architecture of the system.

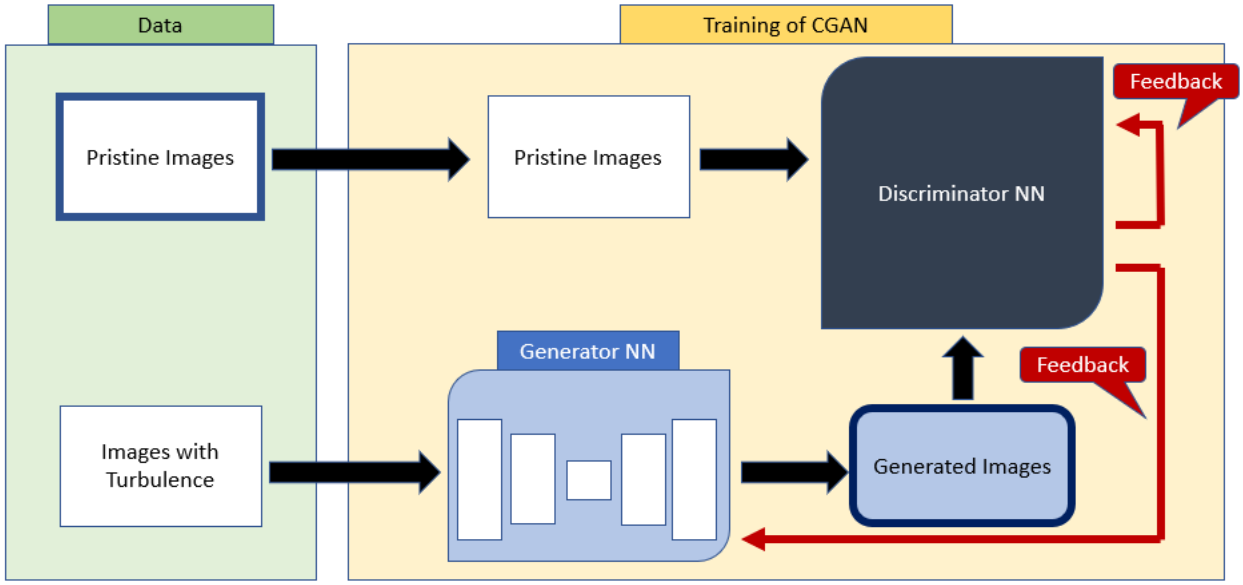


Figure 2.10 – cGAN Architecture

Chapter 3: Modeling Water Braking

Operated by the 846th Test Squadron (TS), the Holloman High-Speed Test Track (HHSTT) at the Holloman Air Force Base (HAFB), New Mexico is the world's premier facility for testing full-scale high-speed flight vehicles. The TS plans and executes world-class rocket sled tests, HHSTT is one of the unique Air Force Test Centers (AFTC) that provides cutting-edge Test and Evaluation (T&E) capabilities in support of AF critical weapon system technology and development. The HHSTT conducts a diversity of tests for assessing the performance of various aerospace systems. Examples include bomb and fuse function and survivability at impact (Figure 1), weapon lethality, ejection seat performance (Figure 2), guidance system accuracy, material erosion and ablation, rain effects on the survivability of aerospace materials, aerodynamic characteristics of aerospace system designs, and others [1–3]. It is used for conducting hypersonic (beyond Mach 8) [2,4–8] impact tests using rocket-propelled sleds that run on a 9.63 miles steel rail track. The rocket sleds are held on the rails by steel shoes/slippers [9,10].

Operational flight speeds at the HHSTT are achieved using rail-mounted rocket-propelled sleds. A typical test will have the test article attached to a forebody sled that is driven by one or more pusher sleds that accelerates the forebody to the desired velocity (Figure 1). Often, the test article is launched off the end of the track as shown in Figure 1. However, many tests require recovery of the test article on the rail for post-test analysis. Therefore, some braking mechanism is needed to stop the sleds before running out of track. By recovering the sled from a test, the cost and time requirements for future tests may also be reduced. Retrieving a sled means that it can be used or modified for future trials. The methods to stop/recover the sleds include curving the track, water braking, spoilers as air brakes, or hooks and cables [11]. Traveling at high speeds with the condition to recover sleds, the test requires that items be brought to rest quickly and safely

sometimes from hypersonic speeds without damaging the sled for reusability purposes. Moreover, most trials have multiple sleds in a sled train, some or all of which must be recovered on the rail. Many recovered sleds are decelerated to a stop through a combination of aerodynamic drag (sled coast-out) followed by entry into water braking. Water braking works to stop the vehicle through a momentum exchange, where the water is diverted vertically or laterally by sled-mounted scoops, wedges, or other features.

There are several methods for deploying water braking depending on whether the sled is dual rail or monorail, location on the track in which braking is initiated, and even speed at which braking is initiated.

For dual rail pusher sleds [2,12,13], water is deployed in the trough between the rails at increasing depths. Water depth is controlled using frangible dams that are inserted into the channel at planned locations. The sled braking scoop encounters and diverts the water in the trough, which provides a drag force and consequently slows down the sled. Dual rail forebody sleds may be stopped by rail-top braking. For rail-top braking, plastic tubes are filled with water taped to the top of the rail. The water bags are encountered by sleds-mounted scoops or a sled-mounted wedge. Increasing braking forces are applied using larger diameter water bags. There are two primary water braking methods for monorail sleds: rail-top water bags as just described, and temporary water troughs deployed along both sides of the rail.

Testing items and systems designed to operate in a flight environment under realistic conditions can be expensive and difficult to measure accurately. Force magnitudes on the sled due to interaction with water during braking are unknown. This results in an inability to precisely predict the sled stop point along the rail or to optimize sled designs and trajectory profiles. Understanding water braking phenomenon through CFD modeling [14–18] will improve

the T&E capability at HHSTT on several fronts, including reductions in test costs and test risk as well as an improved ability to achieve customer-desired velocity and acceleration profiles.

3.1 METHODOLOGY

The simulations are carried out with the Finite Volume Method (FVM) approximations. FVM provides a powerful numerical approach to perform CFD simulations. Multiphase (two-phase/water-air) flow dynamics will be modeled using the Volume of Fluid (VOF)[19–21]. The VOF model can model two or more immiscible fluids by solving a single set of momentum equations and tracking the volume fraction of each of the fluids throughout the domain. Typical applications include the prediction of a jet breakup, the motion of large bubbles in a liquid, the motion of liquid after a dam break, and the steady or transient tracking of any liquid-gas interface. The VOF formulation is generally used to compute a time-dependent solution, but for problems in which you are concerned only with a steady-state solution, it is possible to perform a steady-state calculation. For example, the flow of water in a channel with a region of air on top and a separate air inlet can be solved with the steady-state formulation. The VOF formulation relies on the fact that two or more fluids (or phases) are not interpenetrating. For each additional phase that we add to our model, a variable is introduced: the volume fraction of the phase in the computational cell. In each control volume, the volume fractions of all phases sum to unity. The phases share the fields for all variables and properties and represent volume-averaged values if the volume fraction of each of the phases is known at each location. Thus, the variables and properties in any given cell are either purely representative of one of the phases, or representative of a mixture of the phases, depending upon the volume fraction values. Based on the local value of q^{th} fluid's volume fraction

in the cell (α_q), the appropriate properties and variables will be assigned to each control volume within the domain (Figure 2).

The tracking of the interface(s) between the phases is accomplished by the solution of a continuity equation (Eq 3.1) for the volume fraction of one (or more) of the phases. For the q^{th} phase, this equation has the following form:

$$\frac{1}{\rho_q}\left[\frac{\partial}{\partial t}\left(\alpha_q \rho_q\right) + \nabla.\left(\alpha_q \rho_q \mathbf{v}_q\right) = S_{\alpha_q} + \sum_{p=1}^{n}\left(\dot{m}_{pq} - \dot{m}_{qp}\right)\right] \tag{3.1}$$

where $\dot{m}_{qp}$ is the mass transfer from phase q to phase p and $\dot{m}_{pq}$ is the mass transfer from phase p to phase q. By default, the source term on the right-hand side of the equation (S_{α_q}), is zero, but you can specify a constant or user-defined mass source for each phase. The volume fraction equation will not be solved for the primary phase; the primary-phase volume fraction will be computed based on the unity constraint for the sum of the volume-fraction of all the phases.

The properties appearing in the transport equations are determined by the presence of the component phases in each control volume. In a two-phase system, if the phases are represented by the subscripts 1 and 2, and if the volume fraction of the second of these is being tracked, the density (ρ) in each cell is given by $\alpha_2 \rho_2 + (1 - \alpha_2)\rho_1$. Similarly, all other properties such as viscosity (μ) and pressure (p) are computed. Momentum equations for each phase are solved throughout the domain, and the resulting velocity field is shared among the phases. The momentum equation (Eq. 3.2) is dependent on the volume fractions of all phases through the properties ρ and μ.

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla.(\rho\mathbf{v}\mathbf{v}) = -\nabla p + \nabla.[\mu(\nabla\mathbf{v} + \nabla\mathbf{v}^T)] + \rho\mathbf{g} + \mathbf{F} \tag{3.2}$$

One limitation of the shared-fields approximation is that in cases where large velocity differences exist between the phases, the accuracy of the velocities computed near the interface can be adversely affected. The energy equation (Eq. 3.3) also shared among the phases.

$$\frac{\partial}{\partial t}(\rho E) + \nabla.\left(\mathbf{v}(\rho E + p)\right) = \nabla.\left(k_{eff}\,\nabla T\right) + S_h \tag{3.3}$$

The VOF model treats energy ($E = \frac{\sum_{q=1}^{n} \alpha_q \rho_q E_q}{\sum_{q=1}^{n} \alpha_q \rho_q}$), and temperature ($T$) as mass-averaged variables where E_q for each phase is based on the specific heat of that phase and the shared temperature. The properties ρ and effective thermal conductivity (k_{eff}) are shared by the phases. The source term (S_h) contains contributions from any other volumetric heat sources.

Figure 3.1: Impact of test – an article on a non-recovered forebody sled with a recovered pusher sled (left) and water bag braking (right).

Table 3.1: Description of the cases to be simulated

Cases	Description
2D Model (4 Cases)	Simplistic scoop design, Low speed (50-100m/s), the Learning curve
2D Model	Domain and mesh refinement increased over previous, High speed (up to Mach ~0.9), Validated with hand calculation
3D- J.Model	Simplistic design, Different angle variations, Small domain for faster conversion
3D-Scoop. Model	Scoop CAD model, added proper trough dimensions, large domain size
3D-Model	Need to add rails, front sled, and pusher sled (future studies)

To accurately represent the fluid dynamics, a proper boundary condition must be imposed on the outer boundaries of the fluid domain and the inner surfaces representing fluid-structure interfaces. Two types of boundary conditions are available, i.e., Dirichlet boundary conditions where specify the value of the function at the boundary, and Neumann boundary conditions where specify the gradient normal to the boundary of the function at the boundary. Sometimes, a mixed type of boundary conditions is used to solve the fluid dynamics problem. In the numerical simulation, it is impossible to solve a problem without defining a boundary condition. The numerical simulation also needs to consider the physical process in the boundary region. Results may vary while applying different boundary conditions. Improper boundary conditions may provide nonphysical influences on the simulation results. That means boundary conditions play an important role in accurate simulation result.

The study was split into five different stages listed in Table 3.1. In green we have the completed stages, yellow we have currently ongoing, and blue is future. The decision to split the study into different stages was made early on to progress with confidence. The objective was to properly identify the correct initial and boundary conditions first before proceeding to more complex models. After having identified those conditions, the geometry could be taken from 2-D

to 3-D. The difference with the 3-D models is that the 3D-J.Model is a simple scoop geometry with a small domain used to test initial conditions, boundary conditions, and set up before modeling the 3D-Scoop.Model (actual scoop used in some sleds for water braking at the HHSTT).

3.2 IMPORTANT CFD STEPS

The proposed software used for the CFD simulations is Fluent. To perform CFD simulations with the software, the following steps are required:

- Acquire the approximated physical model and create the geometry from a Computer Aided Design (CAD) software (NX or Solidworks).
- Create the mesh by importing the CAD model into a mesh generator software (ANSYS Design modeler) and removing sharp edges/corners.
- Choose the appropriate mathematical models governing equations of multiphase fluid flow. These equations are also called Navier-Stokes equations which are derived from conservation laws for the mass, momentum, and energy in a control volume and are a coupled system of five non-linear PDEs for 3D in space and time.
- Choose a numerical method used to discretize the equations being solved.
- Analyze the results with post-processing tools to visualize the results such as contour plots of velocity, temperature, and pressure; drag and lift profiles, and 3D time evolution of eddies structures.

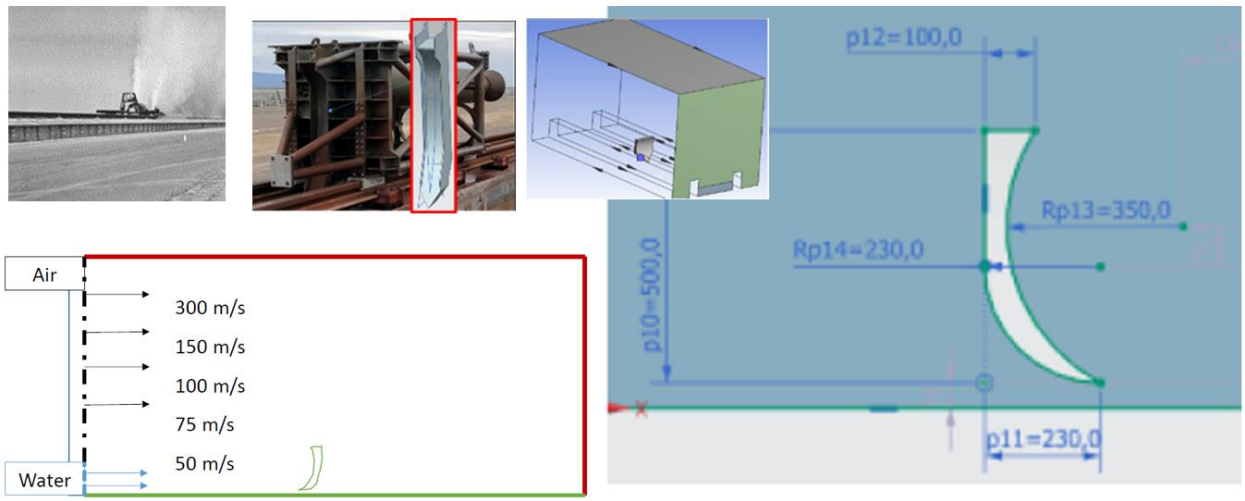


Figure 3.2: Preliminary results (SFFP 2018): Water braking instigation with CFD

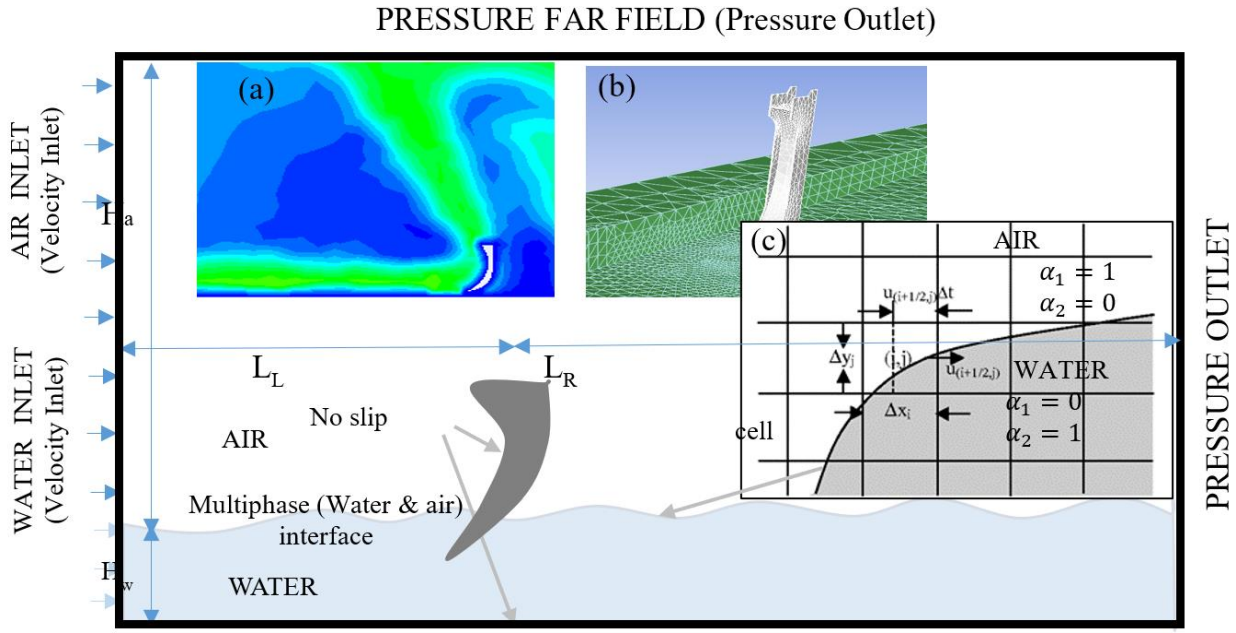


Figure 3.3: Schematic of important parameter and BCs in CFD setup and simulations with example of simulated results (a), mesh (b), and two-phases VOF description (c).

3.3 Modeling Approximations

In water braking mechanism, water at increasing elevation is used to hit scoop attached with the sled providing a drag force and consequently slowing down the sled. Methods for

deploying water braking in dual rail scoop mechanism will be studied following a step-by-step process.

- Step 1: Define the geometry (boundaries and scoop) of the system as a simple representation of water braking scoop mechanism, depicted in Figure 3.2. It will inform us about forcing mechanism and magnitude that water level has on the scoop and estimate its impact on the scoop.
- Step 2: Define boundary conditions (anything not defined in Figure 3.3 is a wall, no slip) for the system, shown in Figure 3.3.
- Step 3: Assume a step-by-step increase in water level and velocity profile as used in the dual rail pusher sleds where water is deployed in the trough between the rails at increasing depths, and the depth is controlled using frangible dams that are inserted into the trough at planned locations.

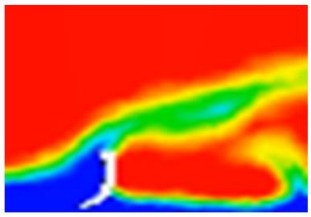


Figure 3.4: 2D Volume fraction Contour at 100 m/s

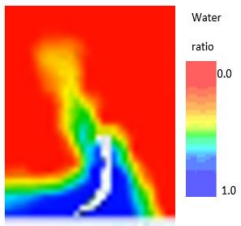


Figure 3.5: 2D Volume fraction at 300 m/s

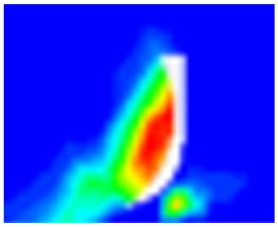


Figure 3.6: 2D Static pressure at 100 m/s (blue – Low PRESSUR; Red – High PRESSURE

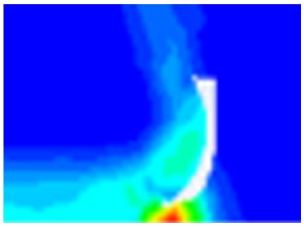
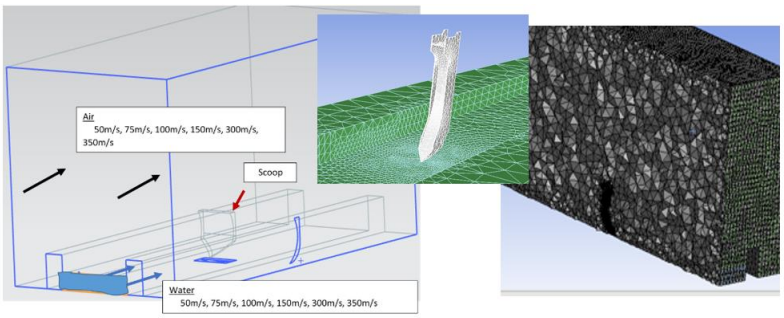


Figure 3.7: 2D Dynamic pressure at 100 m/s (blue – Low pressure; Red – High PRESSURE)



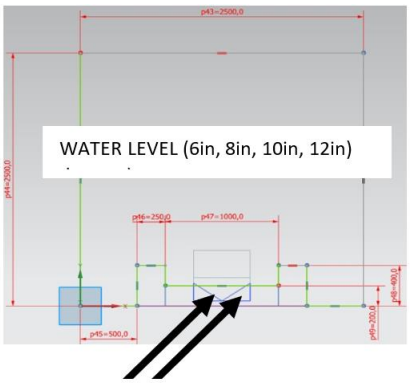


Figure 3.8: 3D Model

3.4 Results and Discussion

The 2D model in Figure 3.2, illustrates the control volume (domain) and fluid flow. 2D models were run with different initial velocity vectors for both air and water inlet while everything else remained the same. The magnitudes were 50 meters/second (m/s), 75 m/s, 100 m/s, 150 m/s, and 300 m/s. The 2D model reviewed in this section is the second iteration of the 2-D model. Both air and water are always run at the same speed in all models. The geometry created is just a general scoop with no correlation to any scoops used on the track. As previously stated, the purpose of the 2D model was to establish initial conditions and boundary conditions. Also, the 2D model was also used to validate models' calculations with hand calculations at 300 m/s.

Force (by hand) = 1.35e7 N/m Force (2D model) = 1.4e7 N/m

The simplicity of the design allowed for hand calculations to be done and compared to results from the 2D model in Fluent. Having validated the 2D model, we could proceed to the more sophisticated 3D model. The mesh was refined for reduced skewness and quality was improved around the scoop. The mesh grid is also mostly structured for faster conversion. BCs are Two

inlets, one air the other water, One outlet for pressure, Six walls (4 for the scoop, one for the top wall, and one for the bottom wall). The mesh consisted of 41,591 cells. The set-up is illustrated in Figure 2. Gravity was enabled at -9.81 m/s^2. The solver used was pressure-based transient allowing for a Multiphase Volume of Fluid (VOF) type of model with time steps. Making it pressure based allowed for multiphase flow modeling of air and liquid water. The VOF consisted of 2 Eulerian phases, air and water, and the k-ε viscous model was used for turbulence. Energy equation was accounted for when inlet velocities were equal to or higher than 100 m/s, the density of air becomes compressible an ideal gas can be used to control the density. Water is assumed to be incompressible with constant density. To simulate the sled's scoop entering water braking, air and water were used as velocity inlets at the desired velocity of sled entering water braking. For boundary condition to simulate the real-world situation, the top and right side (red) are pressure outlets, bottom, as well as four scoop boundaries (green), are walls with no slip, and finally, in dashed blue and black lines, we have velocity inlet of air and water at varying velocities.

Figures 3.4, 3.5, 3.6, 3.7 are for qualitative purposes. The volume fraction contour displays the water (blue) hitting the scoop and going over. We will study the sensitivity of the results to the computational mesh. Red is air, while light blue and green represent a mixture of air and water, water is sprayed back. At 100m/s, water does not have enough velocity to be pushed forward. The volume fraction contour displays the water (blue) hitting the scoop and going over, Figure 3.4. Red is air, while light blue and green represent a mixture of air and water, water is sprayed back. At 100m/s, water does not have enough velocity to be pushed forward. Water was accelerated to 300 m/s or about Mach 0.87. At this velocity, we can see that the water no longer flows over the scoop (Figure 3.5). It is reflective of what had been observed in an actual test of sleds entering at velocities close to Mach 1. Most interesting is the effect of the water behind

scoop, having water jump up as such with still significant velocity can cause damage to the sled behind the scoop. In Figure 6, High static pressure is produced in the frontal area of the scoop and is responsible for water building up and going over at 100 m/s. This pressure on the scoop is also accountable for creating drag and acting as a braking force on the scoop which is transitioned to the sled. Dynamic pressure builds up can be observed; this creates lift that can be dangerous for a sled as it can cause it to lift from the rails (Figure 3.7).

The 2D model initial conditions and boundary conditions were validated though drag calculations and visual inspection of volume fraction contours. Having proper understanding and conditions established, work on the 3D model could proceed. Several cases of 2D were run, different velocity and water height. The slight difference in the calculation, from one of the cases, between the 2D CFD model and the one by hand is the slight arc in the geometry at the top. For simplicity, the hand calculation was taking as if the scoop was deflecting water straight up. In the CFD case, however, there is a slight arc that pushes water towards the front causing slightly higher braking force.

The 3D Test model was built as a precursor to the scoop of an actual sled and to study specific geometry effects on the water (Figure 3.8). The point of attack of the scoop and the geometry of the scoop remained the same as in the 2-D model. The difference was the cuts on the front of the sled. The trough for the water was added to the geometry.

The 3D Test model was built as a precursor to the scoop of an actual sled and to study specific geometry effects on the water (Figure 3.8). The point of attack of the scoop and the geometry of the scoop remained the same as in the 2-D model. The difference was the cuts on the front of the sled. The trough for the water was added to the geometry. From Figure 3.8, looking at the right side of the 3-D Geometry is the 2-D model used earlier. Actual environmental geometry

was added such as the trough where the water is. The angle at the tip of the scoop to its sides varied between 0, 5, 35, and 60 degrees as well as the level of the water. The different water heights for the water entry were 6in, 8in, 10in, 12in. The scoop geometry remained the same with those exceptions. Cases were created for all combinations of degrees and water height. The geometry had a small domain for quicker initial results. The model geometry is not yet representative of the actual track. The mesh for the 3-D Test Model was considerably larger even as the domain was kept small. It had a total of 493,806 elements (3D polygons through which flow is modeled).

The boundary conditions are like the 2D with the exception that sections were coupled together into larger sections. Mostly remained the same, except instead of the boundaries being 1D they became 2D. The boundary sections at the desired velocity of the sled is entering water braking. To converge the initial mesh time step sizes had to be 1e-7. The domain is 5 meters long meant that at 100 m/s it would take 500,000 time-steps for air and water to make it from one side of the control volume to the other. For accuracy, 50 iterations per time step would be taken bringing the total number of iterations required for a single model to go from one end to the other to 25 million iterations. Even parallelized depending on the number of cores it would take a day. Due to a large amount of time to complete, because of the number of elements and time steps, the mesh was redone with a much smaller mesh and time step.

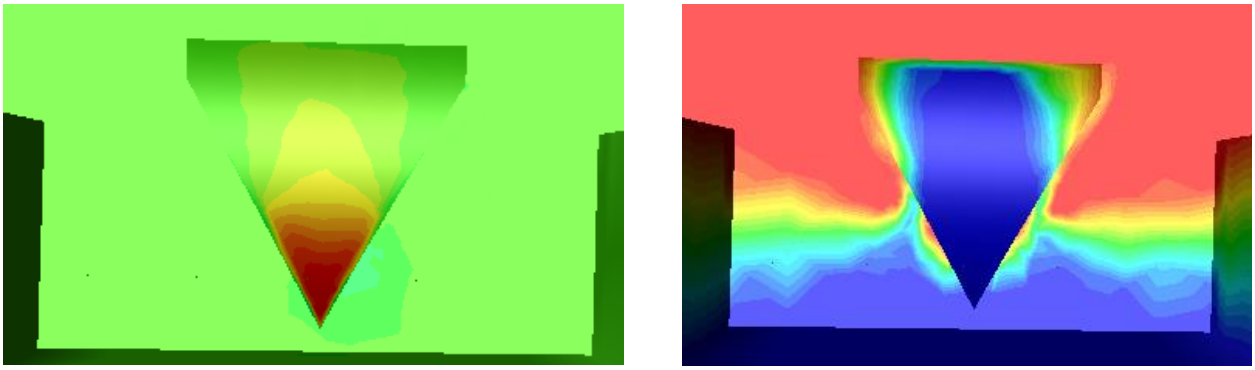


Figure 3.9: 3-D Scoop Model Static Pressure and Volume Fraction Contour at 50m/s

Scoop area was water makes contact has high static pressure build-up (Figure 3.9), like 2-D. The volume fraction contour on the left is interesting as there is no water shown near the bottom sides. A mesh with higher resolution will be required to identify if there is truly no water there or if it is an error being propagated by the mesh elements being too large. However, it may make sense if there were no water there as the scoop cuts the water.

A total of 96 cases are planned. Different angles, water heights, and velocities. Cases were not able to be completed in time, due to the small-time step size required for converting the number of time steps needed had to be very large. Contours of cases shown in results were acquired using a very coarse mesh that could provide a glimpse of water behavior phenomena. Without a finer mesh, it will be challenging to get accurate results. Results may be affected by large element size, therefore, may be misleading. Cases must be run with much higher mesh density. Large amounts of HPC hours are required.

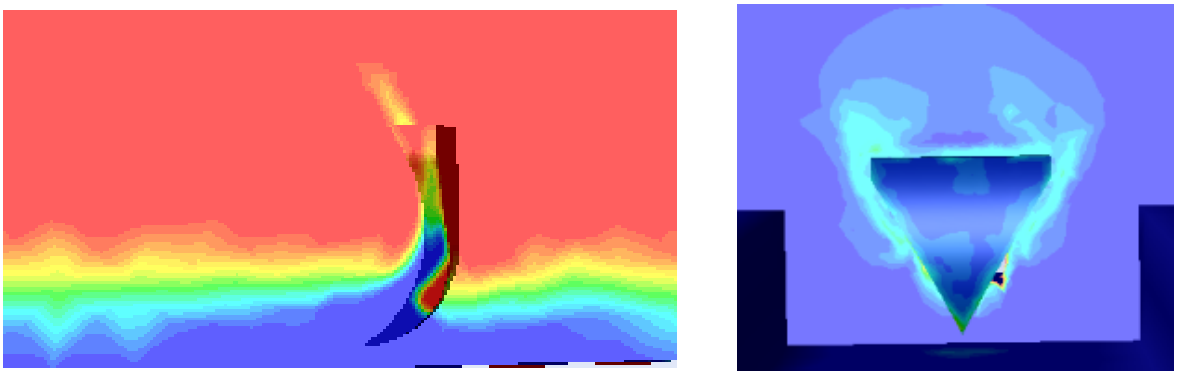


Figure 3.10: 3D Model Volume Fraction and Vorticity at 50 m/s

3D Scoop Model is the same as 3D Test Model except for a much larger domain and the scoop geometry. Everything else stayed the same. The cases created for it are based only on water height and velocity of fluids (air and water). Water heights planned are the same as in 3D Test

Model, 6in, 8in, 10in, 12in. The geometry of the new scoop is much more complex, the arc at the top allows it to provide much higher braking force as it shoots water to the front. Another notable difference is the created were air inlet, water inlet, scoop walls, a bottom wall, top outlet, back outlet, left outlet, and right outlet. To simulate the sled's scoop entering water braking, air and water were used as velocity inlets "guards" on the sides that keep some of the water from escaping to the sides. This allows for more water to be thrown forward, increasing braking force. The mesh consists of 464,423 elements with refinement near the scoop and on the bottom wall where water will run. In similar fashion to the 3-D test model, a total of 16 cases are planned. Different water heights and velocities. Cases were not able to be completed in time due to the small-time step size required for converting the number of time steps needed had to be very large. Contours of cases shown in results were acquired using very coarse mesh and much smaller time steps. This, however, caused the model to fail to converge soon after the results shown in Figure 10. Without finer mesh, it'll be challenging to get analyzable results. Cases have been created to run, but summer ended before being able to run them. Cases must be run with much higher mesh density. Large amounts of HPC hours are required. Even with an original mesh of over 464,000 the mesh elements seem too large due to the large control volume.

3.5 Verification and Validation (V&V) Framework (Future Work)

In the last decades or so, the CFD simulations has undergone a strong development and has become a powerful tool both for the analysis and understanding of fluid dynamics phenomena, and similarly for the design of AF systems such as the sled test, made possible due to the advent of faster and faster supercomputers with increasing memory capabilities and advanced computing algorithms. However, the CFD results are strongly affected by the numerical methodology employed (i.e., accuracy, efficiency, geometrical modeling, etc.) and computer resources for real-

life problems. The limitations of CFD need to be addressed due to the lack of understanding and/or inappropriate physical modeling as well as limited experimental information. The credibility of CFD simulations can only be established by a rigorous process of verification and validation[22–25].

Over several decades, the HHSTT has conducted several high-speed sled tests with water braking mechanism and collected data of predicted profile with varying degree of success, confidence, and fidelity. This along with tremendous knowledge built around the braking mechanism would provide a rich dataset for the team to perform validations of predicted CFD results with a high degree of confidence. We propose to follow the established guidelines for assessing the credibility of the CFD Modeling and Simulation (M&S) results [22,26,27]. A possible roadmap to build a validation strategy is described in Table 3.2. The two main principles that are necessary for assessing the results are Verification and Validation.

Table 3.2: A roadmap to build a verification & validation strategy

Step 1	Step 2	Step 3
• Consider simple flow features with simple scoop design. • Identify test cases for which analytic or hand-calculation are available/can be estimated with a high degree of confidence. • Assess the accuracy, convergence, and capability of the CFD results.	• Consider simple/moderate flow physics. • Establish grid requirements for various geometry & operating conditions. • Assess model assumptions through comparison with benchmark experiments.	• Consider full Multiphysics physics flow to assess accuracy by comparing the high fidelity CFD results with experiments. • Evaluate model limitations and determine grid requirements. • Carry out the parametric analysis with through comparison with test data.

Verification is the process of determining how accurate a computational simulation represents a given conceptual model. The verification process will be subdivided into two major components: code verification and calculation verification. Since we propose to use ANSYS Fluent, commercial software that has gone through rigorous code development test, we propose to

do testing for simple 2D geometries mimicking hand calculation and compare the results with the HHSTT hand calculated results to establish confidence in the computer program. In the calculation verification, we propose to estimate and develop the basis for numerical errors due to discretization approximations. Validation determines how accurate a simulation (i.e., the conceptual model) represents the phenomenon to be investigated.

In general, the verification must precede validation. The need for validation experiments and the associated accuracy requirements for computational model predictions are based on the intended use of the model and should be established as part of V&V activities. Validation of an HHSTT braking phenomena will be pursued hierarchically from the component level to the system level starting with hand-calculation to check-out runs and sled tests with varying degree of component complexity and operating speeds. The validation strategies will include the predictive capability of the model in the physical realm of interest by addressing uncertainties that arise from both simulation results and experimental data.

3.6 CONCLUSION

We developed 2D and 3D models for the scoop system of the sled. The CFD calculations were validated for drag calculations using simple momentum balance on the sled. Boundary conditions were ensured by visual inspection of volume fraction contours and conservation laws. The slight difference in the calculation, from one of the cases, between the 2-D CFD model and the one by hand is the small arc in the geometry at the top. For simplicity, the hand calculation was taken as if the scoop was deflecting water straight up. In the CFD case, however, there is a slight arc that pushes water towards the front causing slightly higher braking force. Several cases of 2D geometries were run to develop basic understandings of drag profile for different velocity

and water height. Mesh independency for the 3D model has not been done, the 3D model is still a work in progress.

Chapter 4: Turbulence Deconvolution using a Conditional Generative Adversarial Net

4.1 Introduction

Atmospheric turbulence varies on many factors including air direction, temperature, speed, and density. It is responsible for the blurring effects seen on images taken of objects in space. In Figure 1, the image on the left illustrates the turbulence phenomena. Turbulence phenomena are mostly modeled by Kolmogorov's theory of turbulence, it will be used to simulate turbulence in non-provided generated images. The research objective is to create a model that can reverse turbulent effects, in images, caused by atmospheric conditions (see Figure 4.1) to identify objects, and its features, in space with greater 'clarity'. Priority number one is to achieve a high degree of image quality when compared to a 'pristine' pre-turbulence image (greater than 90 percent accuracy to the original). A secondary objective is to reduce the time required to remove turbulence on an image to 3 seconds, while still maintaining enough image quality to identify objects.

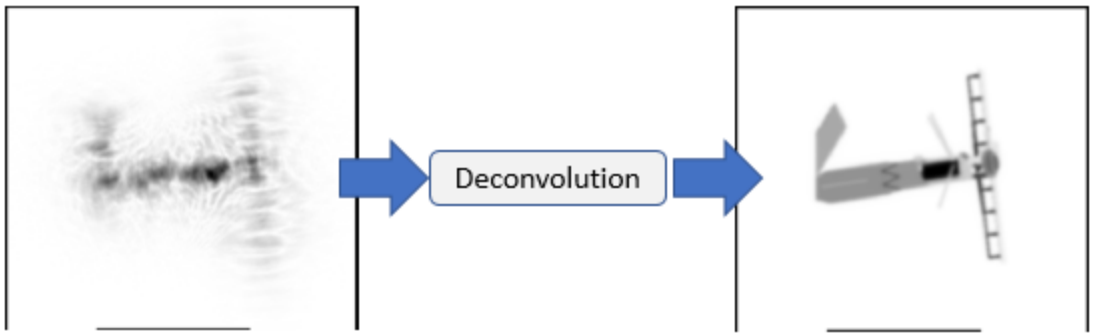


Figure 4.1: Satellite turbulence deconvolution

4.2 Methods

For removing turbulent effects on images, a Conditional Generative Adversarial Net (cGAN) will be used. A Generative Adversarial Net (GAN) consists of two neural nets being trained simultaneously as they are pitted against each other [31]. One acting as a generative model

(Generator) and the other as a discriminative model (Discriminator), the Generator is tasked with 'tricking' the Discriminator into thinking that the images it will provide it are from a 'real' dataset. The Discriminator on the other hand is tasked with determining if an image is 'pristine' or not.

The Generator will work by generating images from turbulent images given, the Generator will initially build images of objects in turbulence but will slowly converge towards images without turbulence as it 'learns' from feedback provided by the Discriminator. As the Discriminator rejects the images being given to it by the Generator as fake, the Generator receives a 'generated output' from a generator loss function that the generator then optimizes to reduce and hence converges towards the minimization of the loss function output. By minimizing the loss, the Generator gets closer to tricking the Discriminator. In this process the Generator learns to apply 'deconvolution' to turbulent images as it attempts to trick the Discriminator into believing that the image it gives it are 'real' or non-turbulent. See Figure 4.3.

The Discriminator works by training it on images with pristine, blurred (turbulent), and generated images simultaneously. It learns to classify images as turbulent or non-turbulent, assigning probability of the ladder through a sigmoid function. Its role in the GAN is to train the Generator by telling it how 'close' the images it generates are to the non-turbulent images. If training is successful, the Discriminator's classification of generated images as non-turbulent will be around 50%. Anything below 40% and the Generator model would not be a successful deconvolution model, it may still be improved. Anything more than 60% and the Generator has become too good, its potential to get better decreases as the Discriminator is determined to struggle classifying turbulent and non-turbulent images. It may signify Discriminator is not capable of determining non-turbulent images and therefore providing the Generator bad feedback, meaning the Generator model may not be very good. A well-trained GAN will achieve an equilibrium of

about 50% accuracy for the Discriminator, the Generator is good enough to trick the Discriminator, but the Discriminator is good enough to keep improving the Generator. As the purpose of this report is to address the deconvolution of Turbulence in images and not GANs, GANs are only briefly explained. For greater detail on GANs please refer to [31,32].

The cGAN, that would remove turbulence, started with a GAN that would generate images from random noise. This simpler GAN is shown in Figure 4.2, below. The objective is to add components and capability to this simpler GAN to eventually evolve into something that would remove turbulence.

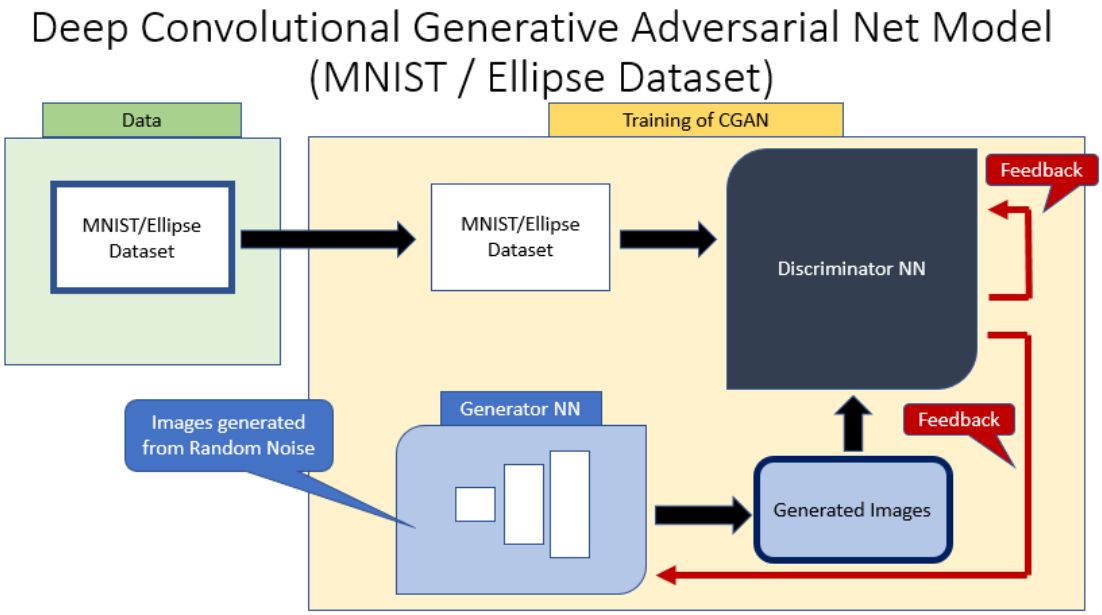


Figure 4.2: Deep Convolution GAN, used for MNIST and Ellipse dataset

After successful training and testing of GAN with MNIST and random Ellipse data, another component to the GAN is to be added. Shown below in Figure 4.3, turbulent images are added along with the pristine images in the training of the GAN. It is a model similar in function to the one used in [33] to recover degraded images.

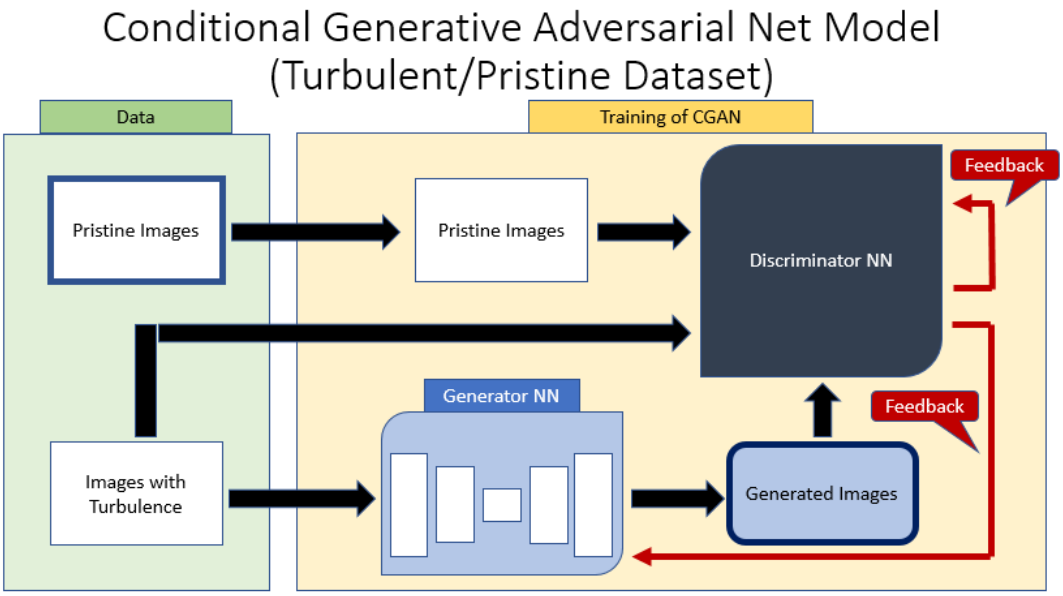


Figure 4.3: Conditional GAN, planned for use with generated Turbulent and Pristine Ellipse dataset as well as generated Satellite Dataset

4.3 EXPERIMENTS

All current results are qualitative. The Generator NN for the DCGAN in Figure 4.2 was initiated with random noise. The dataset used to train the DCGAN is data from MNIST. It includes 60,000 samples of written integers, Figure 4.4.

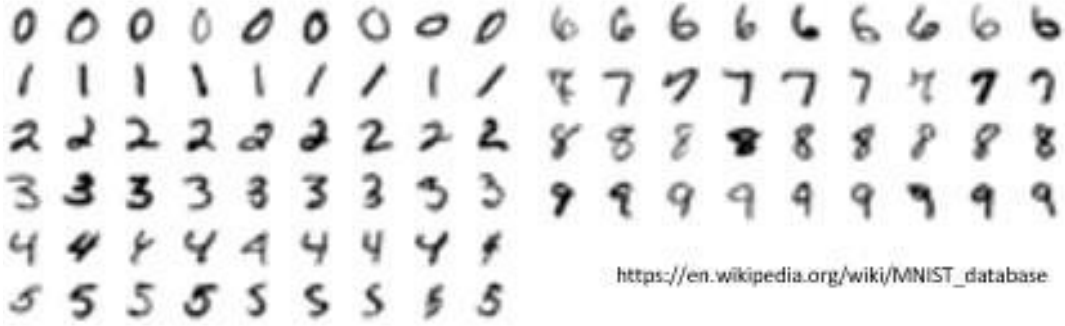


Figure 4.4: MNIST dataset sample

After 50 epochs or cycles, the GAN managed to produce some realistic results for some digits. They can be viewed in the figure below, Figure 4.5.

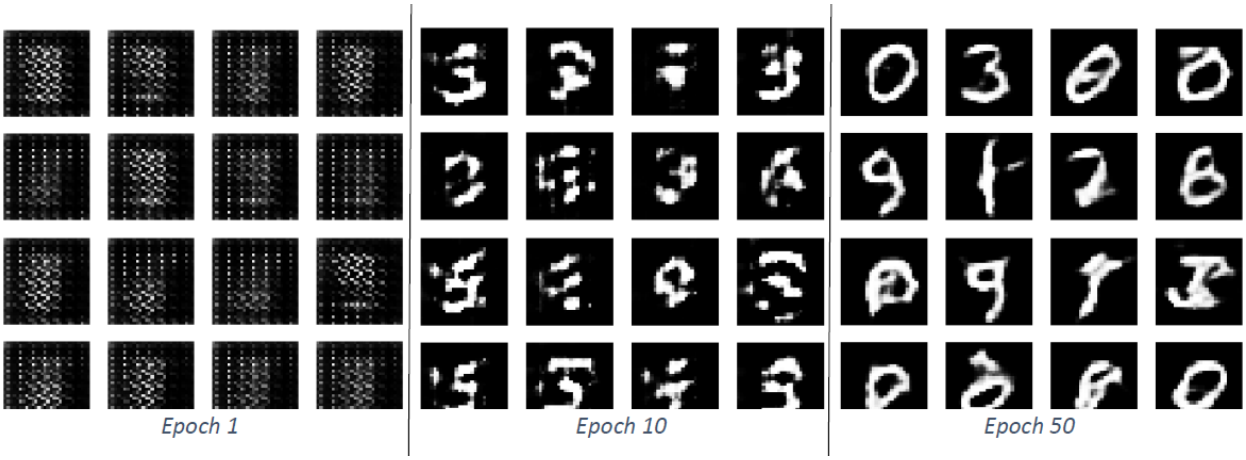


Figure 4.5: DCGAN MNIST results

For the Ellipse dataset, 200 random ellipses were generated using python. All 200 have different sizes, major and minor radius lengths, orientation, and location. Figure 4.6 illustrates three potential samples from the dataset. For the DCGAN (same GAN as MNIST) only ellipses as in Figure 4.6 were used. The next training, after MNIST, was to teach the GAN to 'create' ellipses in a similar manner to how it learned to 'write' integers in Figure 4.5.

Figure 4.6: Sample of random Ellipses in Ellipse dataset

The result of the cGAN are shown below in Figure 4.7. The GAN again starts from random noise and slowly begins to converge into what can qualitatively be classified as learning to create ellipses of different size.

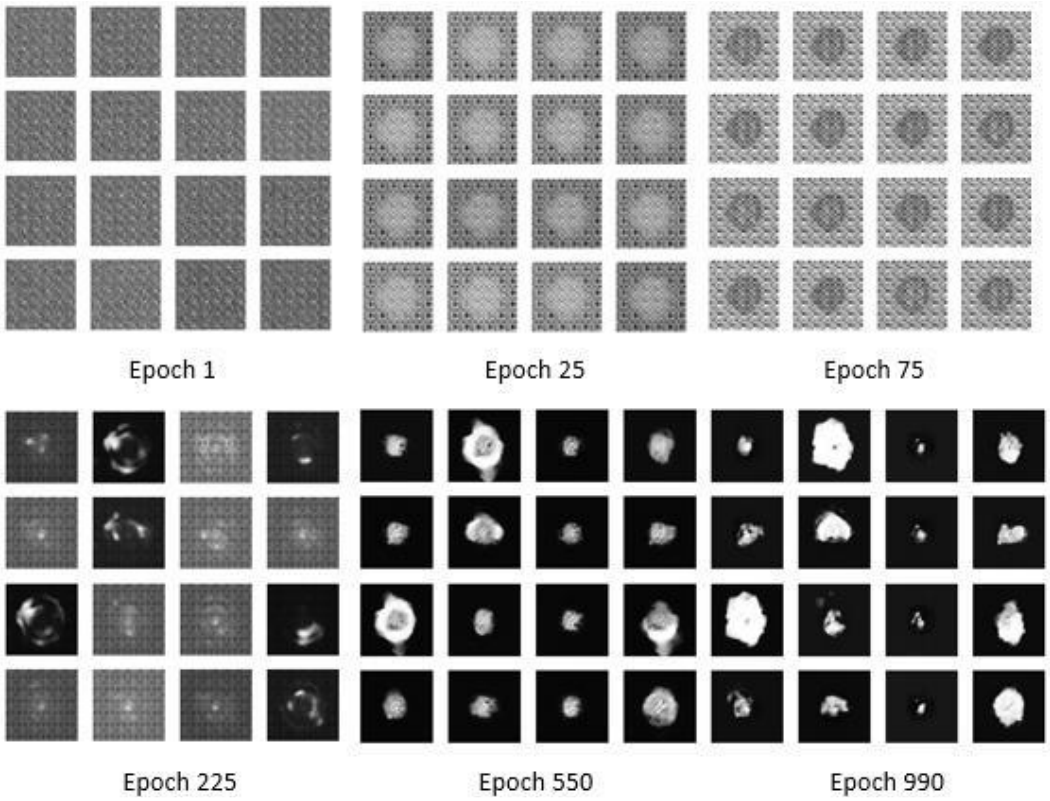


Figure 4.7: DCGAN learning to draw ellipses based on Ellipse Dataset

After the successful training of the Ellipse drawing GAN, the ellipse from the dataset undergoing turbulence were to be incorporate. At the point of incorporation of turbulent images, the GAN in Figure 4.3 is used instead of GAN in Figure 4.2. The turbulent ellipses are created by applying turbulent filters onto the images, simulating what the image would look like at turbulent levels desired. Turbulence filters are created by applying a filter based on Kolmogorov's equation. These same filters are applied to generated satellite images shown in Figure 4.8. At the writing of this report the CGAN with turbulent images is still a work in progress.

Figure 4.8 illustrations satellite data in which the Conditional GAN will be trained on at the conclusion of the study. Currently, the study is not far along to remove turbulence from blurred figures and produce the pristine images (shown in Figure 4.8).

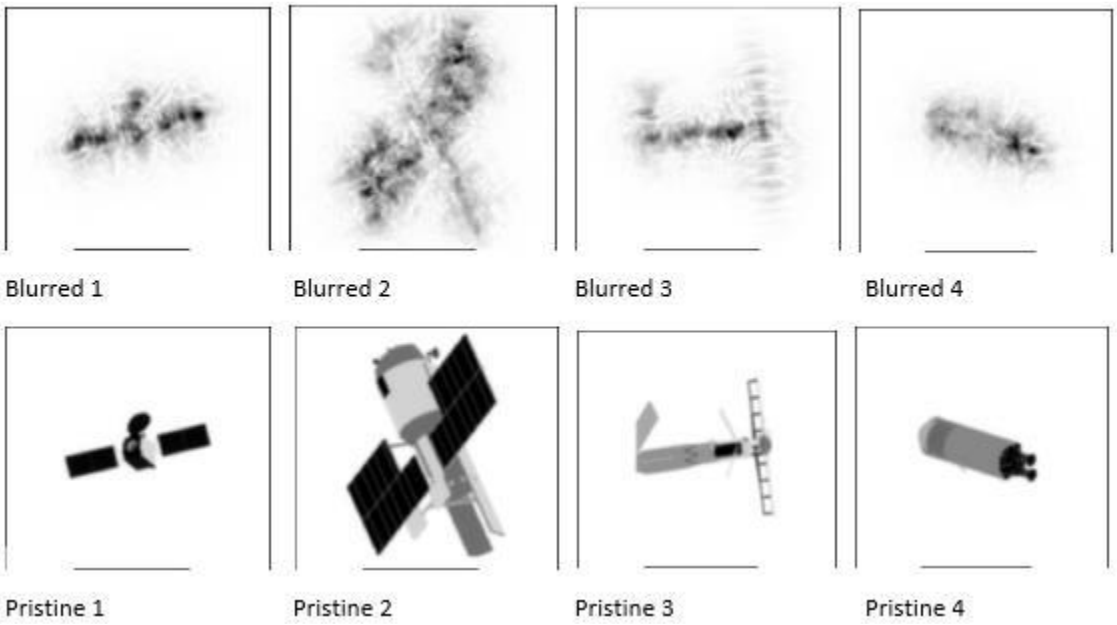


Figure 4.8: Generated Blurred (turbulent) and Pristine (non-turbulent) Satellite images

4.4 DISCUSSION

Preliminary results show promise in the DCGAN. Qualitatively, the Generator is approaching the 'real' images provided to the Discriminator. The next phase of the research is to replace random noise initialization for the Generator with turbulent Ellipse images and have it train towards non-turbulent images. The idea is that the Generator model would then be able to provide deconvolution to images with turbulence.

An advantage of using generated images is having exact images without turbulence, something not possible in the real world. This allows the deconvolution training to converge towards a 'perfect' image. A disadvantage however is that turbulent images are generated and not an exact reflection of turbulence, they approximate atmospheric conditions using Kolmogorov's

equation for Turbulence. Another limitation is that a cGAN would have to be trained for 'categories' of turbulence, as one cGAN model fits all would not provide accurate results. The dataset would have to be partitioned and trained on similar turbulent levels. The result would be many cGAN model corresponding accordingly to turbulence category.

For future work, the goal is to complete the research and use the large declassified dataset of generated satellite images to train different turbulence models according to turbulence category. After turbulent models are trained for each specific turbulence category, a third neural net would have to be integrated to act as a turbulence classifier to assign the appropriate turbulence model to a provided turbulent image. In addition, perhaps Reinforcement Learning could be included during the training of the GANs to improve learning capability.

4.5 Conclusion

The research is to be conducted in 5 stages, see Figure 4.9. First, the Generative Adversarial Net (GAN) and python Turbulence models, which will act as filters on images, were developed in parallel. Second, GAN was tested using MNIST dataset and images of ellipses, of different sizes and orientation, were created with and without a turbulence filter. Third (current stage), the ellipse turbulent and pristine images are to be fed into the GAN to test the loading and training of the GAN with generated images. Fourth, test ellipse model on blurred (turbulent) satellite images provided. Fifth, train the GAN with generated satellite data provided and quantify the quality of the deconvolution of images, from blurred to pristine.

Figure 4.9: Research plan in stages

At the conclusion of 7 weeks the research is in stage 3, see Figure 4.9. Stages 1 and 2 were successfully completed as demonstrated in the Experiments section. At Stage 3 the integration of a Variational Auto-Encoder (VAE) as the Generator is not yet complete. By using a VAE the objective is to be able to process and manage turbulent images as input into the Generator and begin the training of a deconvolution model.

At the moment of writing this report, all results are qualitative. The datasets previous to the generated satellites, MNIST and Ellipse, were used to build up towards the GAN being able to process different data structures. They were used as proof of concept.

RESEARCH MILESTONES

1) Initial GAN (MNIST Test) ----------> COMPLETE

- Had GAN generate handwritten numbers from data
- Qualitative results

2) GAN (Ellipse Test, see Ellipse Dataset in Supporting Graphics) ----------> COMPLETE

- generated 200 random ellipses as dataset to test GAN
- Qualitative results

3) GAN (Ellipse; Blurred and Pristine Training) ----------> CURRENT

4) GAN Ellipse Model Test (w/ Generated Satellite Data) ----------> INCOMPLETE

5) GAN Train and Test (using Generated Satellite Data) ----------> INCOMPLETE

Chapter 5: Conclusion

As defined by the roadmaps provided for each project, Modeling Water Braking and Turbulence Deconvolution using a Conditional Generative Adversarial Network, there is still work stages to be completed. Proof of concept for both has been established, however, and that is enough to get a glimpse of the potential impact of an integration of both.

5.1 Putting it Together

By completing the complex 3D CFD models laid out, we will have and be able to produce data for the cGAN to learn from. The main goal of the integration is to improve testing capability at HHSTT by melding strengths of CFD and cGANs. As advantages as their strengths are, we must not forget that their shortcoming will only magnify when put together. As CFD is inherently with errors due to the inability to capture unsteady fluid phenomena that occurs at high Reynolds numbers, training a cGAN with errors which itself will not be entirely correct because it is also only trying to replicate real data will increase the error margins. This perhaps will make the use of a cGAN with CFD unpalatable if error control and tolerance are a concern. There is hope however that at the very least there will be qualitative data good enough to warrant combining two. Qualitative, though not as useful as quantitative, can be of used for general understanding of fluid mechanics during specific test that can help identify areas of high importance, focus, and potential concern. This then still becomes advantages as the number of CFD analysis can be reduced and therefore meeting the established objective of reducing cost, computational resources requirements, and increasing testing.

5.2 FUTURE WORK

After the completion of a cGAN that can help determine qualitative results of water breaking, the next step is to use the expanded capability for the development and engineering of new sleds that would allow better braking at HHSTT. The new sleds could then do more with less by being able to expand on real world testing capabilities. By using CFD and cGAN a sled with more optimal braking potential can be built. New sled designs can be analyzed without the risk of failure in real world or without taking too much computational resources. This can allow for the conception of a sled with greater and safer braking potential at a relatively lower cost of what it would be now. This would in turn help increase the speed at which test can be made and still allow for recoverability using water braking.

References

1. Hooser, M., Schwing, A.: Validation of dynamic simulation techniques at the Holloman High-Speed Test Track. In: 38th Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics
2. Turnbull, D., Hooser, C., Hooser, M., Myers, J.: Soft Sled Test Capability at the Holloman High-Speed Test Track. In: U.S. Air Force T&E Days 2010. American Institute of Aeronautics and Astronautics
3. Kennan, Z.A.: Determination of the constitutive equations for 1080 steel and vascomax 300. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT (2005)
4. Minto, D.: CTEIP Funded Advances in Hypersonic Testing at The Hollman High-Speed Test Track. In: 24th AIAA Aerodynamic Measurement Technology and Ground Testing Conference. American Institute of Aeronautics and Astronautics
5. Marren, D., Lu, F.: Advanced Hypersonic Test Facilities. American Institute of Aeronautics and Astronautics (2002)
6. Minto, D.: The Holloman High-Speed Test Track Hypersonic Upgrade Program. In: 22nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference. p. 3034
7. Minto, D.: Recent increases in hypersonic test capabilities at the Holloman High-Speed Test Track. In: 38th Aerospace Sciences Meeting and Exhibit. p. 154 (2000)
8. Minto, D.W., Bosmajian, N.: Hypersonic test capabilities at the Holloman high-speed test track. Prog. Astronaut. Aeronaut. 198, 499–530 (2002)
9. Cinnamon, J.D., Palazotto, A.N.: Analysis and simulation of hypervelocity gouging impacts for a high-speed sled test. Int. J. Impact Eng. 36, 254–262 (2009).

10. Factor, L.: Holloman High-Speed Test Track Design Manual. Res. Summ. 1, 2 (2005)
11. Maker, D., Meyer, C.: Reusable thrust-powered sled mounted on an inclined track for launching spacecraft and airborne vehicles at supersonic speeds, (2006)
12. RASMUSSEN, H.: Captive flight testing by means of rocket sleds. In: 3rd Aerodynamics Testing Conference. p. 364 (1968)
13. Uncapher, W.L., Hohnstreiter, G.F.: Radioactive material package testing capabilities at Sandia National Laboratories. Sandia National Labs., Albuquerque, NM (United States) (1995)
14. Bower, W., Kibens, V., Cary, A., Alvi, F., Raman, G., Annaswamy, A., Malmuth, N.: High-frequency excitation active flow control for high-speed weapon release (HIFEX). In: 2nd AIAA Flow Control Conference. p. 2513 (2004)
15. Schetz, J.A.: Aerodynamics of high-speed trains. Annu. Rev. Fluid Mech. 33, 371–414 (2001)
16. Hong-Qi, T.: Study evolvement of train aerodynamics in China [J]. J. Traffic Transp. Eng. 1, 1–9 (2006)
17. Weinstein, L.M., Minto, D.: Focusing schlieren photography at the Holloman high-speed test track. In: 22nd International Congress on high-speed photography and photonics. pp. 865–874. International Society for Optics and Photonics (1997)
18. Nakata, D., Yajima, J., Nishine, K., Higashino, K., Tanatsugu, N., Kozu, A.: Research and development of high-speed test track facility in Japan. In: 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. p. 928 (2012)
19. Gidaspow, D.: Multiphase flow and fluidization. (1994)
20. Kotteda, V.M.K., Kumar, V., Spotz, W.: Dakota Integrated with MFiX for UQ Analysis:

Sensitivity of particle size on the pressure in a fluidized bed DEM simulations. In: 2018 Workshop on Multiphase Flow Science. National Energy Technology Laboratory (2018)

21. Delgado, P., Chen, F., Kumar, V., Harris, C., Katta, K.: Simulation Of Single And Two-Phase Newtonian Flow In Carbon Capture And Storage Processes Using Variational Methods.
22. ASME: Guide for Verification and Validation in Computational Solid Mechanics. Am. Soc. Mech. Eng. (2006).
23. Oberkampf, W.L., Trucano, T.G.: Verification and validation in computational fluid dynamics. Prog. Aerosp. Sci. (2002).
24. Sargent, R.G., Balci, O.: History of verification and validation of simulation models. In: Proceedings - Winter Simulation Conference (2018)
25. Kleijnen, J.P.C.: Verification and validation of simulation models. Eur. J. Oper. Res. (1995). doi:10.1016/0377-2217(94)00016-6
26. Jayanti, S.: Computational Fluid Dynamics for Engineers and Scientists. (2018)
27. CFD-Committee: Guide: Guide for the Verification and Validation of Computational Fluid Dynamics Simulations. (2002)
28. Muller, Jens-Dominik (2016) *Essentials of Computational Fluid Dynamics*. Boca Raton, FL: Taylor & Francis Group.
29. Geron, Aurelien. (2017). *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. Sebastopol, CA: O'Reilly.
30. D. A. L. Jimenez, "Implementing Large Eddy Simulation to Numerical Simulation of Optical Wave Propagation," 2018.
31. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville

A., Bengio Y., “Generative Adverserial Nets”, NIPS, 2014.

32. Radfor A., Metz L., Chintala S., “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, arXiv:1511.06434v2, 2016.

33. Schawinski K., Zhang C., Zhang H., Fowler L., Santhanam G.K., “Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit”, *Monthly Notices Royal Astronomical Soc.: Lett.*, vol. 467, no. 1, pp. L110-L114, May 2017.

34. Terrazas, J., Kotteda, V. M. Krushnarao, Kumar, V., Edmonds R. & Zeisset M., The CFD Modeling of the Water Braking Phenomena for the Holloman High-Speed Test Track, presented at the ASME-JSME-KSME 2019 8th Joint Fluids Engineering Conference in San Francisco, CA.

35. L.F.R. Sanchez, “Machine Learning Analysis to Characterize Phase Variations in Lasre Propagation Through Deep Turbulence,” 2020.

Vita

Jose Terrazas received his B.S. in Mechanical Engineering, M.S. in Systems Engineering, and a Graduate Certificate in Big Data Analytics from The University of Texas at El Paso (UTEP). He is currently pursuing a second M.S. in Computational Science and PhD in Mechanical Engineering at UTEP under Dr. Vinod Kumar.

Areas of research include multi-phase modeling in water braking phenomena during high speed braking using Computational Fluid Dynamics (CFD) and Machine Learning (ML) for the deconvolution of turbulence in images. He has received recognition by the Office of Research and Sponsored Projects at UTEP and was recognized for demonstrated excellence and skill in Project and Systems Engineering Management. In addition, Jose was a 2018 and 2019 Air Force Research Lab Summer Student Fellow.