



Remote Sensing and Imaging Physics: Developing new metrics for deep turbulence effects on laser propagation through long path

Vinod Kumar
UNIVERSITY OF TEXAS AT EL PASO

09/10/2020
Final Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTB1
Arlington, Virginia 22203
Air Force Materiel Command

DISTRIBUTION A: Distribution approved for public release.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Executive Services, Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.					
1. REPORT DATE (DD-MM-YYYY) 17-09-2020		2. REPORT TYPE Final Performance		3. DATES COVERED (From - To) 15 May 2019 to 14 May 2020	
4. TITLE AND SUBTITLE Remote Sensing and Imaging Physics: Developing new metrics for deep turbulence effects on laser propagation through long path				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-17-1-0253	
				5c. PROGRAM ELEMENT NUMBER 61102F	
6. AUTHOR(S) Vinod Kumar				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSITY OF TEXAS AT EL PASO 500 UNIV ST ADMIN BLDG 209 EL PASO, TX 79968-0001 US				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AF Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/AFOSR RTB1	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-AFOSR-VA-TR-2020-0166	
12. DISTRIBUTION/AVAILABILITY STATEMENT A DISTRIBUTION UNLIMITED: PB Public Release					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT There is an expanding interest in characterizing atmospheric turbulence effects on laser beams propagating over a long path for directed energy applications. For optical systems, lasers propagating long distances lose their coherence due to the fluctuations in the refractive index. A dominant cause for the refractive index fluctuations, can be attributed to cumulative effect of turbulent eddies present in the atmosphere along the laser beam propagation path. Current analysis of atmospheric turbulence effects is based on theoretical and statistical models [1][3] derived from the Kolmogorov approach. Supporting experimental evidence [4][6] helped validate the classical theory of optical propagation. Inaccuracies develop when implementing this model for various conditions like long range propagation, deep atmospheric turbulence, or a combination of both showing experimental results that deviate from the theory. While Kolmogorov approach is still used for long path propagation, further investigation is needed to comprehend its limitations and determine parameters that can describe deep turbulence effects.					
15. SUBJECT TERMS atmospheric physics					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF	19a. NAME OF RESPONSIBLE PERSON YAKES, MICHAEL
a. REPORT	b. ABSTRACT	c. THIS PAGE			
Standard Form 298 (Rev. 8/98) Prescribed by ANSI Std. Z39.18					

DISTRIBUTION A: Distribution approved for public release.

Unclassified	Unclassified	Unclassified	UU	PAGES	19b. TELEPHONE NUMBER <i>(Include area code)</i>
--------------	--------------	--------------	----	-------	--

FINAL REPORT
for the
Research Project
(AFOSR Award Number – FA9550-17-1-0253)

Project Title:
Remote Sensing and Imaging Physics: Developing new metrics for deep turbulence effects on
laser propagation through long path

By
Principal Investigator:
Vinod Kumar, Ph.D.
Department of Mechanical Engineering
The University of Texas, El Paso (UTEP)
500 W. University Ave El Paso, TX 79968

June, 2020

Table of Contents

1	Summary	3
2	Introduction.....	4
3	Research Progress	4
3.1	Methodology	4
3.2	Optical Turbulence and Associated Parameters.....	5
3.3	Prior research.....	6
3.4	Simulations.....	6
3.5	Discussion	8
3.6	Conclusion.....	11
4	Accomplishments.....	11
4.1	Students	11
4.2	Awards, Fellowship & Collaboration.....	11
4.3	Scholarly contributions	11
4.4	AFRL Relevant Capabilities Development.....	12
5	Acknowledgement	12
6	Bibliography	12

1 Summary

This project seeks to develop metrics to effectively characterize the impact of deep turbulence on laser beam propagation over a long path through critical examination of turbulent eddies' structures in spatial, temporal, and spectral domains. The study employs expertise from complementary field of computational fluid dynamics. The investigation is expected to result in better understandings of power-spectra for the refractive index fluctuations leading to improved adaptive optics technologies.

The advance in the research carried out during the period Aug 2018 Jun 2019 is focused on using a linear optimization technique called Branch and Bound on a Deep Learning Platform (Tensor Flow), to estimate the phase variations of a laser beam that propagates through a turbulent atmosphere. These estimations have been made comparing the calculations of the phase variations that are obtained in analytical form by statistical estimations of the Kolmogorov turbulence as well as equations of phase deviations based on Fresnel propagation in a vacuum and atmospheric turbulence vs the calculations using the Branch and Bound algorithm that is feed into TensorFlow. The results obtained are very encouraging given that the estimates made using Deep Learning techniques are 2,608% away from the analytical estimates.

The broader impact lies in advancing the state-of-the-art of the laser based remote sensing and imaging science which plays important role in the Air-borne laser (ABL) and ground based directed energy (DE) technologies. Future DE technologies can be used to effectively engage and destroy airborne missiles and/or track enemy's air vehicles (e.g. satellites) which involve the beam propagation through a long atmospheric path. Findings from the proposed basic research can be integrated into advanced adaptive optics (AO) technologies.

2 Introduction

There is an expanding interest in characterizing atmospheric turbulence effects on laser beams propagating over a long path for directed energy applications. For optical systems, lasers propagating long distances lose their coherence due to the fluctuations in the refractive index. A dominant cause for the refractive index fluctuations, can be attributed to cumulative effect of turbulent eddies present in the atmosphere along the laser beam propagation path. Current analysis of atmospheric turbulence effects is based on theoretical and statistical models [1]–[3] derived from the Kolmogorov approach. Supporting experimental evidence [4]–[6] helped validate the classical theory of optical propagation. Inaccuracies develop when implementing this model for various conditions like long range propagation, deep atmospheric turbulence, or a combination of both showing experimental results that deviate from the theory. While Kolmogorov approach is still used for long path propagation, further investigation is needed to comprehend its limitations and determine parameters that can describe deep turbulence effects.

3 Research Progress

3.1 Methodology

Atmospheric turbulence is caused by the large-scale variations of temperature in the air. Temperature variations generate wind. Strong wind can transition laminar flow to turbulent flow. In turbulent flow, air at different temperatures begin to mix producing randomly distributed pockets of air that vary in size called eddies. Attempting to numerically solve turbulent flow can be difficult and computationally taxing, especially for the case of deep turbulence. To characterize atmospheric turbulence in this study, the Kolmogorov approach is implemented to statistically model the effects of eddies. Kolmogorov model proposes that the kinetic energy of a turbulent flow is transmitted from the larger eddies into smaller eddies.

The energy cascade theory begins with eddies the size equal to that of the outer scale length L_o . The input energy is then progressively divided and redistributed into smaller eddies until the eddies are equal, in size, to the inner scale length l_o (Figure 1). Eddies within this inertial range are assumed to be statistically homogeneous and isotropic.

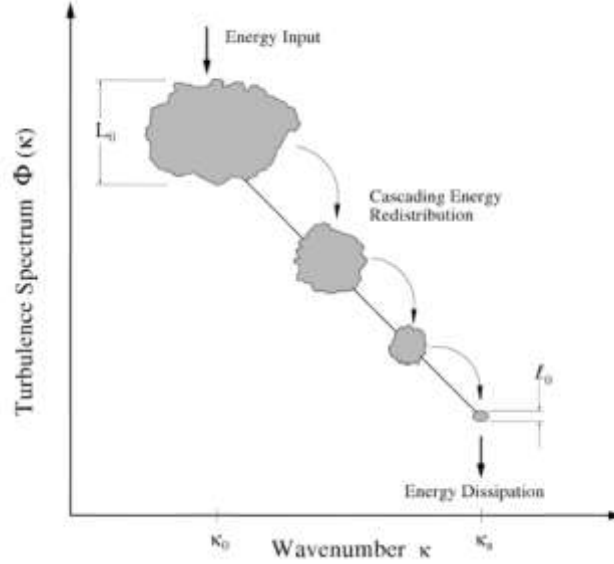


Figure 1: Description of Kolmogorov Energy Cascade Theory [6]

In this subrange of local homogeneity, Kolmogorov [1] considered the difference between two velocity points to be statistically dependent on the displacement vector separating them. The assumption led him to develop the statistical structure of turbulence that can be described by a single structure function. This theory only holds true for displacement vectors between the outer-inner scale lengths.

3.2 Optical Turbulence and Associated Parameters

Tatarskii [3], [7] proved, with the concept of conservative passive additives, that the refractive index fluctuations obey the same structure function law as velocity and temperature turbulence. In doing so we calculate the refractive index structure constant C_n^2

$$C_n^2 = \left(\frac{79 \times 10^{-8} P}{T^2} \right)^2 C_T^2$$

where P is atmospheric pressure, T is air temperature, and C_T^2 is the structure function parameter for temperature. The typical range of values for C_n^2 are between $(10^{-13} - 10^{-17} \text{ m}^{-2/3})$. After deriving the structure function for the refractive index, we can determine the spectral density function $\Phi_n(\kappa)$ in term of Kolmogorov's inertial subrange model (Tatarskii [3] and Strohbehn [8]).

$$\Phi_n(\kappa) \frac{1}{4\pi^2 \kappa^2} \int_0^\infty \frac{\sin(\kappa r)}{\kappa r} \frac{d}{dr} \left[r^2 \frac{d}{dr} D_n(r) \right] dr = 0.033 C_n^2 \kappa^{-11/3}$$

There are other models for the refractive power spectral density. For our study, we will be using the von Kármán to include the outer-scale effects.

$$\Phi_n^{vk}(\kappa) = \frac{0.033 C_n^2}{(\kappa^2 + \kappa_0^2)^{11/6}} \quad \text{for } 0 \leq \kappa \ll 1/l_0$$

where $\kappa = 2\pi(f_x \hat{i} + f_y \hat{j})$ is the angular spatial frequency in rad/m, C_n^2 is the refractive index structure constant and $\kappa_0 = 2\pi/L_0$.

In this study, the Rytov index will be used to represent the intensity of the atmospheric turbulence. Rytov is the log-amplitude fluctuations for the plane wave and is given by $\sigma_x^2 = 0.312C_n^2 k^{7/6} L^{11/6}$ for Kolmogorov spectra.

3.3 Prior research

In the summer 2016 USAF SFFP report [9], the turbulence intermittency was studied for the turbulence effects. It was modeled through non-Kolmogorov methods using fractal descriptions of eddies that incorporate space filling concepts present in the inertial turbulence ranges. The main objectives of the research were to examine the impact on intensity variance and total number of zero intensity points in the outputs with respect to the Rytov index and the power law exponent. The study showed that the variance of intensity increased linearly approximately four times the Rytov value for the case of weak turbulence. The theory broke down at high turbulence as the variance of intensity drastically increased at Rytov=0.5 before saturating at around Rytov=1. Further investigations however are needed to establish stronger correlations between eddies' characteristics in deep turbulence and high order statistics for the beam propagation.

3.4 Simulations

As an extension to the previous study[9], statistical analysis of the phase information at the output signal is examined through the Kolmogorov spectrum. For simplicity, we consider a plane wave propagating through a 10km path with a constant C_n^2 . The inner and outer scale lengths are 5mm and 10m respectively. All simulations are carried for a 2048x2048 number of samples N , with a 5mm grid spacing interval δ , using Fresnel Propagation [10].

$$U(x_2, y_2) = \frac{e^{ik\Delta z}}{i\lambda\Delta z} \iint_{-\infty}^{\infty} U(x_1, y_1) e^{i\frac{k}{2\Delta z}(x_1x_2 + y_1y_2)} dx_1 dy_1$$

To introduce atmospheric turbulence, the propagation distance is subdivided into slabs of distance Δz (Figure 2-right). Within z , the plane wave is propagated linearly using Fresnel propagation in a vacuum. Atmospheric turbulence is introduced at the end of the partial propagation distance (Figure 2, Left). The propagated wave phase is altered by introducing a turbulent-induced phase term to the phase of the propagated wave in the Fourier domain. Therefore, the relationship between the phase spectrum for each screen and refractive-index is $\Phi_\phi(\kappa) = 2\pi^2 k^2 \Delta z \Phi_n^{vk}(\kappa)$.

Phase screen simulations are based on the Fresnel propagation, refractive index power-spectrum and computer-generated random numbers to obtain a random field. These values are then implemented into two-dimensional arrays of phase values on a grid of sample points that have the same statistical values as turbulence-induced phase variations [10].

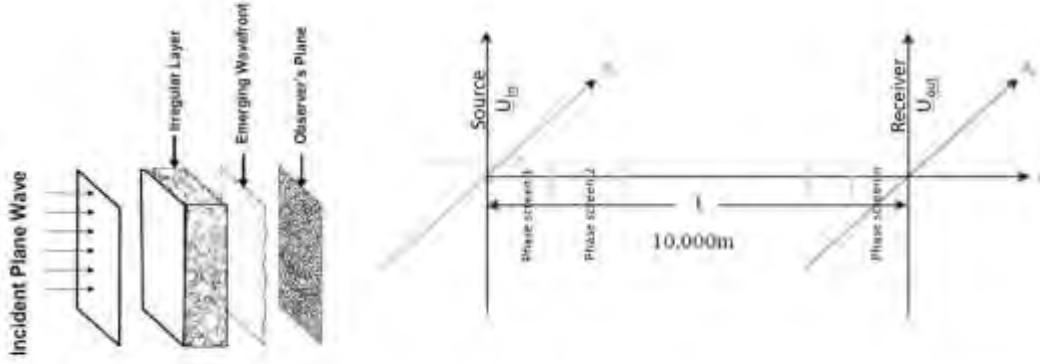


Figure 2: Plane beam propagation schematic

Propagation of the wave field from screen to screen can be written as

$$U = U_0 e^{-i\psi}$$

where U_0 is the solution to the Fresnel propagation in a vacuum of a distance Δz and ψ is the complex phase perturbation. The procedure is repeated for all screens until the plane wave reaches the receiver plane. The simulations are performed using 15 screens over a 10km propagation path, i.e., spacing between screens is 666.6m. Figure 3 presents results for the intensity, wrapped and unwrapped phase of the propagated wave at the receiver plane.

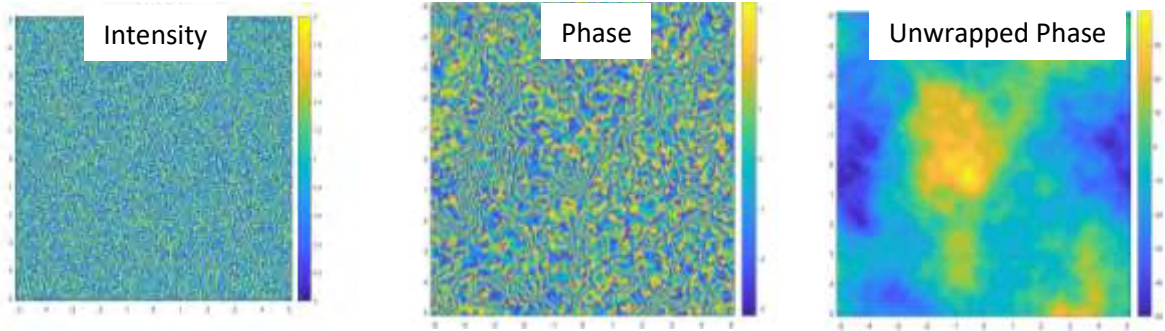


Figure 4: Phase screen simulated results

The phase images that are produced by the plane wave propagating through a turbulent medium have discontinuities (Figure 4-middle). The built-in Matlab version of 2D unwrap function embedded in MATLAB fails to smoothen the discontinuities for strong turbulent cases. A more robust unwrap scheme is developed to perform statistical calculations of the phase data. The scheme is based on the 2D Goldstein branch cut phase unwrapping algorithm [11], [12], compiled by *Bruce Spottiswoode* and edited by *Carey Smith* [13].

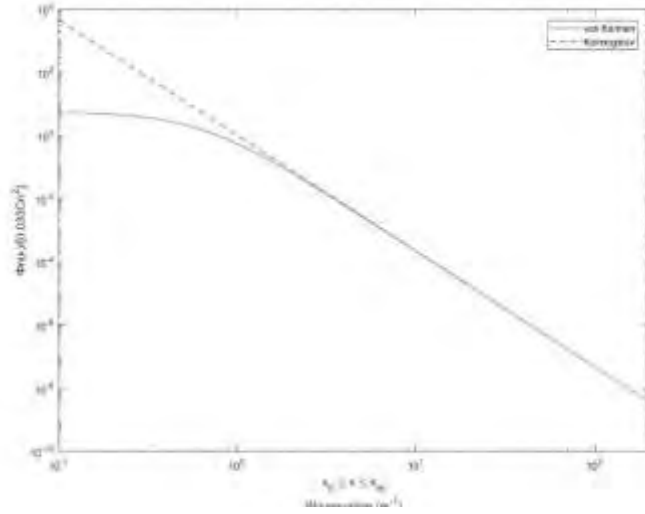


Figure 4: Effects of Large Eddies on the power spectrum

We take the unwrapped phase screens to compute the phase variance. Phase variance is sensitive to the small-wavenumber portion of the spectrum of irregularities and is also dependent on outer scale length, the region where energy is fed into the Kolmogorov energy cascade process. In order to demonstrate the outer scale effects on the phase variance, an approximation provided by *Wheelon* [6] as a function of the von Kármán power spectrum is used and is given as $\langle \varphi^2 \rangle = 0.782 L k^2 C_n^2 \kappa_0^{-5/3}$ where $k = 2\pi/\lambda$.

Like the power spectral density (Figure 4), phase variance expression is primarily determined by the largest eddies. This is important for optics applications because if measured values of L , k , C_n^2 and $\langle \varphi^2 \rangle$ are known, then we can establish a distribution of outer scale lengths. Determining phase variance can therefore help us to understanding outer scale effects of the atmospheric turbulence. If the simulated phase variance values are validated with the theoretical approximation by determining the phase structure function that obeys the same structure function law as the refractive index.

$$D_\varphi(\rho) = \langle (\varphi_1 - \varphi_2)^2 \rangle = 2[\langle \varphi^2 \rangle - \langle \varphi_1 \varphi_2 \rangle] = 2[\Gamma_{\varphi\varphi}(0) - \Gamma_{\varphi\varphi}(\rho)]$$

The phase structure function is another statistical measure to describe the random behavior of phase values and is closely related to the phase autocorrelation. Determining the autocorrelation is often too difficult to compute in the spatial domain but is equivalent to simple multiplication in the frequency domain.

$$\Gamma_{\varphi\varphi}(\rho) = \mathcal{F}^{-1}\{\mathcal{F}[\varphi(r)]\mathcal{F}[\varphi(r)]^*\}$$

The notation $*$ is the complex conjugate of the function and $\rho = |r_1 - r_2|$ is the scalar separation. This will allow for a straightforward approach in determining the phase structure function in the 2-D case.

3.5 Discussion

Calculations show that theoretical phase variance approximation linearly increases with respect to Rytov if C_n^2 , L , k , and L_o are known. Note that these results were obtained for 20 trials within the simulation at relatively smaller grid size (2048x2048) which may not be sufficient.

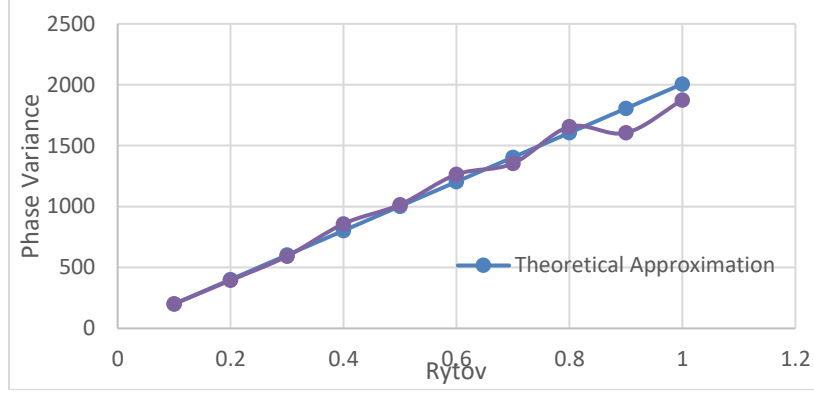


Figure 5: Phase variance at the receiving end in comparison to theoretical approximations

The phase variance at the receiver plane shows a linear trend with respect to the Rytov index with small variations from the theoretical values. Limitations appear at high turbulence ($Rytov > 0.9$) as the phase variance is less than the theoretical approximation. This can be attributed to the finite spatial dynamic range imposed by the maximum available grid size. This is because the von Kármán power spectrum has more influence in low spatial frequencies (Figure 3). The spatial phase screen size needs to be significantly larger so the sample frequency spacings can be low enough to accurately represent the outer scale effects. It would also be ideal to have large number of trials to improve on accuracy but is impractical due to the time constraint. Calculations in this study will continue within the limitations of the simulation ($Rytov < 0.9$).

Knowing that the phase variance approaches the value that is near the theoretical approach, we can proceed to determining the autocorrelation at different levels of turbulence defined by Rytov. The auto-correlation is calculated for phase screens at the receiving end that have a phase variance near the theoretical. The ideal case would be determining an ensemble average of all the autocorrelation values within each simulation. This would lead to storing 20, 2048x2048 matrices, which is not appropriate given the length of this study.

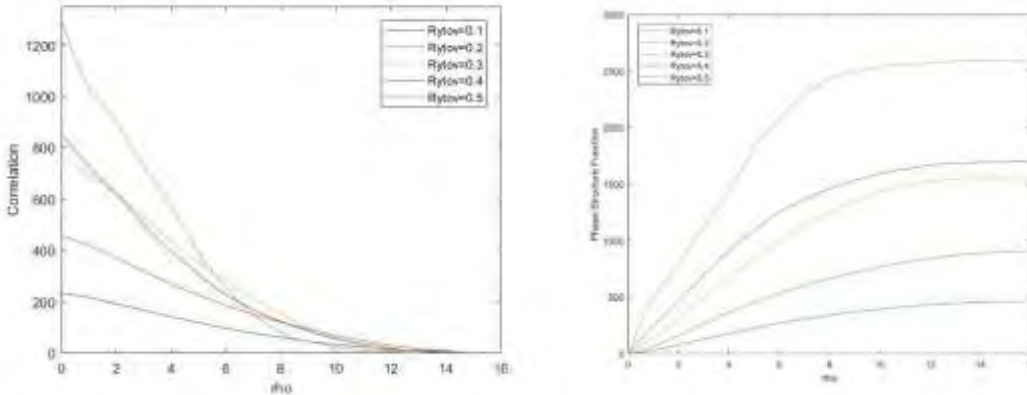


Figure 6: Phase Autocorrelation & Structure Function for varying values of Rytov

Autocorrelation values is calculated for a 2-D plane but only a vector from the center of the plane in the positive x-axis is used to plot (Fig 6.). This is done for simplicity and to compare to 1-D calculated values of the phase structure function. The slope of each autocorrelation decreases exponentially before approaching zero. The initial value for each curve is equivalent to

its respective phase variance. Theoretically at lower levels of turbulence, the autocorrelation should approach zero faster than that of higher levels. This is true for autocorrelation values between the range of $0.1 \leq Rytov \leq 0.3$. The autocorrelation values decline faster than predicted for the finite grid simulations at $Rytov \geq 0.4$. On the other hand, the structure function begins increasing quadratically but begins to bend over as the spacing increases before it saturates to twice the phase variance.

To evaluate the simulated structure function, we need to calculate the phase structure function established by Tatarskii [7].

$$D_{\varphi}(\rho) = 8\pi^2 L k^2 \int_0^{\infty} \kappa \Phi_n(\kappa) [1 - J_0(\kappa \rho)] d\kappa$$

In this equation $J_0(\kappa \rho)$ is the wavenumber weighting function and the structure function can be derived for various models of the power spectrum. The structure function using the Kolmogorov model approaches an asymptotic value. Meanwhile the von Kármán saturates more rapidly and approaches the asymptotic value smoothly. We use these results to interpret the simulated data.

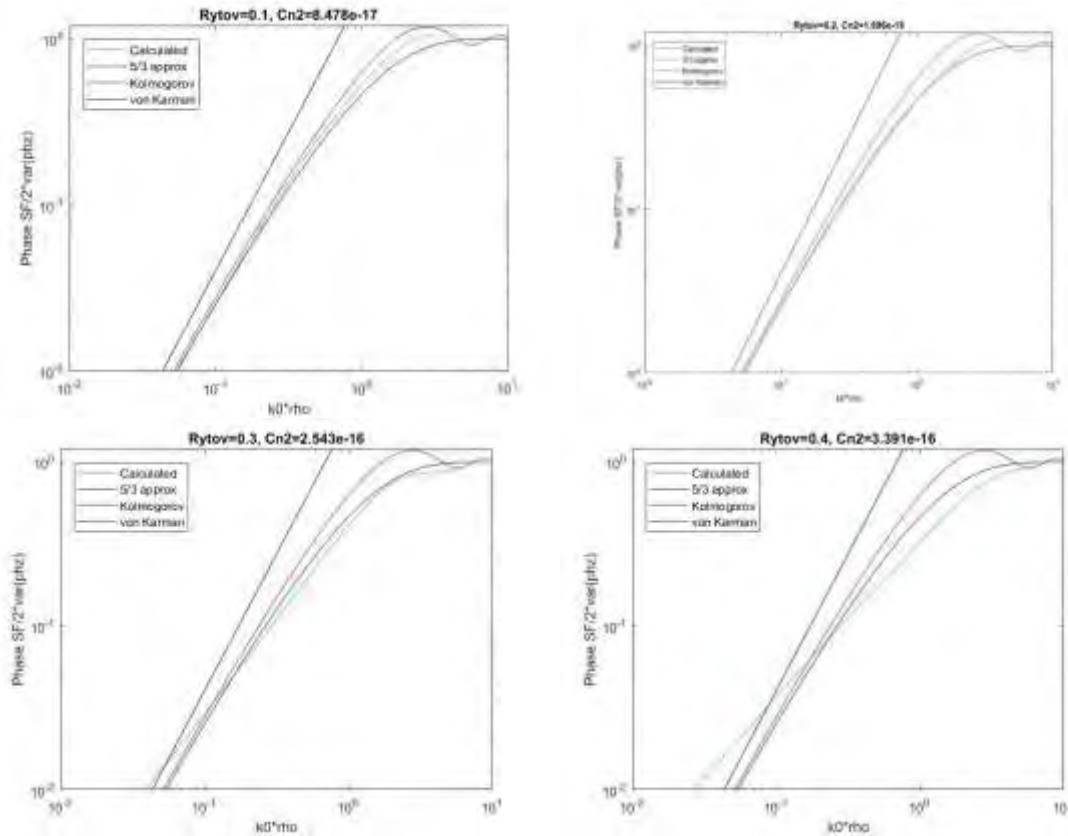


Figure 7: Phase Structure Function compared to calculated values

Progressing from phase variance to the phase structure function, limitations of using a finite spatial grid become more prominent at $Rytov \geq 0.4$. Rytov values above this range will still saturate at the asymptotic value but does not retain the shape of the theoretical von Kármán calculation. Within the range of $0.1 \leq Rytov \leq 0.3$, the phase structure function is approximately same as the calculated theoretical values. Structure Function in Rytov values 0.1 and 0.2 range are initially like the von Kármán before overshooting the asymptotic value. The

structure function at $Rytov=0.3$ is not as smooth as the previous two and doesn't match the von Kármán initially but remains a good approximation at $\kappa_o \rho > .1$. Currently, we can only characterize laser propagation at low turbulence. This does not mean the Kolmogorov model breaks apart at high turbulence, but further investigations are needed. With High Performance Computing (HPC), we can begin simulation with a screen size large enough that will allow for deep turbulence effects for a laser propagating over a long path.

3.6 Conclusion

Unlike the variance of intensity, the phase variance does not saturate with respect to Rytov which allowed for further calculations of phase autocorrelation and structure function. We have noticed the dependence of phase trends to larger eddies in the calculation. HPC can allow us to simulate a spatial phase screen size that is significantly larger than the outer scale length so the sample frequency spacings can accurately represent phase data at high turbulence. This will allow for future investigation of characterizing deep turbulence effects through the phase variance. Further investigations using non-Kolmogorov descriptions can also help us better understand the deep turbulence effects and develop better metrics for the laser propagation.

4 Accomplishments

4.1 Students

- Dr. Luis Rodriguez, PhD. (2020), Mechanical Engineering, UTEP
- Diego Lozano, MS (2018), Mechanical Engineering, UTEP (Employed by Northrop Grumman now)
- Arturo Schiaffino, MS Candidate (2018), Mechanical Engineering, UTEP
- Arturo Rodriguez, Senior, Mechanical Engineering, UTEP

4.2 Awards, Fellowship & Collaboration

- Mr. Diego Lozano, NNSA Fellowship, 2019
- Mr. T. Dorothy (Graduate Student) & Dr. V. Kumar (Faculty), AFOSR SFFP, Student and Faculty Team, Summer 2012
- Mr. T. Dorothy, DE Scholars, Summer 2013 (currently at Tinker Air Force Base)
- Dr. V. Kumar (Faculty), AFOSR SFFP at AFRL - Maui, Summer 2016
- Mr. Diego Lozano, DE Scholars at ARFL Maui, Summer 2018

AFRL Mentor: Dr. V.S. Rao Gudimetla, Senior Research Physicist, AFRL/RDSMA, MSSS

4.3 Scholarly contributions

- D. Lozano*, V. Kumar, V. Gudimetla, "Characterization of Laser Propagation Over a Long Path through Atmospheric Turbulence", SPIE Defense & Security, SI18D_10637-42 (2018)
- D. Lozano*, V. Kumar, V. Gudimetla, V Kottedda, "Artificial Intelligence to predict metrics for characterizing laser propagation in atmospheric turbulence", ASME Fluids Engineering Division Summer Meeting, FEDSM2018-83209 (2018)

- D. Lozano*, V. Kumar, V. Gudimetla, “Investigating metrics based on Phase variance for atmospheric turbulence effects on a 10km Laser beam propagation path”, Advanced Maui Optical and Space Surveillance Technologies Conference (2017)

4.4 AFRL Relevant Capabilities Development

Through collaborative research work with the AFRL scientist, we performed preliminary investigation on the impact of deep turbulence. In summer 2016, we investigated the intermittency behavior through non-Kolmogorov descriptions. This was achieved by choosing the value of power law exponent from a range of exponents and examining their impacts on intensity variance with respect to the Rytov index and number of zero-intensity in the output signals. We employed beta-model for the structure functions which incorporates space filling concepts for the turbulence eddies present at different scale in the inertial turbulence subranges and uses fractal descriptions for the eddies. Following tasks were achieved:

- Developed a Phase Screen codes for plane wave Fresnel propagation
- Incorporated turbulence fluctuations through computerized random number generator (in statistical sense) with prescribed values of Rytov index, C^2_n , or coherence diameter
- Implemented and investigated non-Kolmogorov description of power-spectrum process in deep turbulence using β -model
- Implemented and analyzed space-filling concepts using zero-intensity measure and have crude idea on fillingness of the spectrum (fractal dimension of eddies)
- Computed fractal dimension and analyzed zero-intensity to develop correlation functions

In Summer 2012 and 2013, we examined the role of turbulence in affecting the various important optical parameters such as Fried coherence length, isoplanatic angle, spectra of figure and tilt and several other optical parameters and use this information to develop analytical expressions [1, 2]. The objectives of summer research were to familiarize the adaptive optics and optical turbulences and study the importance of high fidelity computational fluid dynamics and advance the understandings for the experimental work done by Vorontsov, et.al. for the laser beam propagation over a long near-horizontal propagation path between Mauna Loa (Hawaii Island) and Haleakala (Island of Maui) mountains [3]. The optical parameters computed in this report were based on many assumptions made, e.g., computational domain size, physical constants and physics, and sensitivities of the solutions.

5 Acknowledgement

The study was supported by the AFOSR Award Number – FA9550-17-1-0253. I would like to acknowledge the Air Force, AFRL RDSM, MHPCC, and Computational Science and Mechanical Engineering programs at UTEP for partially supporting our research and providing necessary computational and other resources. I would also like to extend a special thanks to the AFRL mentor, Dr. V. Gudimetla, and the AFOSR PM, Dr. Stacie Williams, for valuable inputs at various stages of the project.

6 Bibliography

- [1] V. Kumar and T. Dorothy, "Analysis of Strong Turbulence Effects on Laser," DTIC, 2012.
- [2] D. P. Greenwood, "Bandwidth specification for adaptive optics systems," *Journal of the Optical Society of America*, p. 67 (3): 390–303, 1977.
- [3] M. A. Vorontsov, G. W. Carhart, V. S. Rao Gudimetla, T. Weyrauch, E. Stevenson and et.al., "Characterization of atmospheric turbulence effects over 149 km propagation path using multi-wavelength laser beacons," AIR FORCE RESEARCH LAB KIHAI MAUI HI DETACHMENT 15, 2010.
- [4] V. Tatarskii, "Wave Propagation in a Turbulent Medium," 1971.
- [5] R. Lawrence and J. Strohbehn, "A Survey of Clear-Air Propagation Effects Relevant to Optical Communications," Vols. VOL. 58, NO. 10, , no. OCTOBER, 1970.
- [6] R. Fante, "Electromagnetic Beam Propagation in Turbulent Media," Vols. VOL. 63, NO. 12, , no. DECEMBER, 1975.
- [7] Fante, "Electromagnetic Beam Propagation in Turbulent Media: An Update," Vols. VOL. 68, NO. 11, NOVEMBER , 1980.
- [8] J. Martin and S. Flatt, "Intensity images and statistics from numerical simulation of wave propagation in 3-D random media," Vols. Vol. 27, No. 11, 1988.
- [9] J. Vernin and C. Munoz-Tunon, "Optical seeing at La Palma Observatory. 2: Intensive site testing campaign at the Nordic Optical Telescope," *Astronomy and Astrophysics*, pp. 284, 311-318, 1994.
- [10] F. Roddier, J. Gilli and G. Lund, "On the origin of speckle boiling and its effects in stellar speckle interferometry," *Journal of Optics*, pp. 13, 263-271, 1982.
- [11] J. C. Vassilicos, Intermittency in turbulent flows, Cambridge University Press., 2001.
- [12] U. Frisch, Turbulence: the legacy of AN Kolmogorov, Cambridge university press, 1995.
- [13] J. McDonough, "Introductory lectures on turbulence," University of Kentucky, 2007.
- [14] C. Rao, W. Jiang and N. Ling, "Spatial and temporal characterization of phase fluctuations in non-Kolmogorov atmospheric turbulence," *Journal of Modern Optics*, pp. 47(6), 1111-1126, 2000.
- [15] M. Dowker, "Numerical Study on the Regularity of the Navier-Stokes Equations," University of Sheffield, 2012.
- [16] B. E. Stribling, B. M. Welsh and M. C. Roggemann, " Optical propagation in non-Kolmogorov atmospheric turbulence. In SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics," *International Society for Optics and Phot*, pp. pp. 181-196, 1995.
- [17] J. Schmidt, Numerical Simulation of Optical Wave Propagation with examples in MATLAB, Bellingham: SPIE, 2010.
- [18] V. R. Gudimetla, R. B. Holmes, T. C. Farrell and J. Lucas, "Phase screen simulations of laser propagation through non-Kolmogorov atmospheric turbulence," in *SPIE Defense, Security, and Sensing*, 2011.
- [19] J. P. Bos, M. C. Roggemann and V. R. Gudimetla, "Anisotropic non-Kolmogorov turbulence phase screens with variable orientation," *Applied optics*, pp. 54(8), 2039-2045, 2015.

- [20] V. R. Gudimetla, R. B. Holmes and J. F. Riker, "Analytical expressions for the log-amplitude correlation function for plane wave propagation in anisotropic non-Kolmogorov refractive turbulence," *JOSA A*, pp. 29(12) 2622-2627., 2012.
- [21] R. Rao, "Statistics of the fractal structure and phase singularity of a plane light wave propagation in atmospheric turbulence," *APPLIED OPTICS*, pp. Vol. 47, No. 2, 2008.
- [22] L. C. Andrews and R. L. Phillips, *Laser Beam Propagation Through Random Media*, 2nd Edition, Washington: SPIE Press, 2005.
- [23] V. Kumar, "Laser beam propagation through a long path: Developing new metrics for deep turbulence effects," SFFP report - DTIC, 2016.
- [24] A. Dipankar and S. P., "A new phase-screen method for electromagnetic wave propagation in turbulent flows using large-eddy simulation," *Journal of Computation Physics*, 2009.
- [25] V. Nosov, V. Grigoriev, P. Kovadlo, V. Lukin, E. Nosov and A. Torgaev, "Coherent structures in turbulent atmosphere," Vols. Vol. 7296, 729609, 2009.
- [26] S. Pope, *Turbulent Flows* - 11th edition, Cambridge University Press, 2011.
- [27] C. Higgs, H. Barday, D. Murphy and C. Primmerman, "Atmospheric Compensation and Tracking Using Active Illumination," *LINCOLN LABORATORY JOURNAL*, pp. 11-1, 1998.
- [28] A. Mahalov and M. Moustauoui, "Characterization of atmospheric optical turbulence for laser propagation," *Laser Photon*, Vols. Rev. 4, No. 1, p. 144–159, 2010.

7 **Appendices**

7.1 MS Thesis, Diego Lozano

IMPLEMENTING LARGE EDDY SIMULATION TO NUMERICAL
SIMULATION OF OPTICAL WAVE PROPAGATION

DIEGO ALBERTO LOZANO JIMENEZ
Master's Program in Mechanical Engineering

APPROVED:

Vinod Kumar, Ph.D., Chair

V.S. Rao Gudimetla, Ph.D., Co-Chair

Arturo Bronson, Ph.D.

Heidi Taboada-Jimenez, Ph.D.

V. M. Krushnarao Kottedda., Ph.D.

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

Copyright ©

by

Diego Alberto Lozano Jimenez

2018

IMPLEMENTING LARGE EDDY SIMULATION TO NUMERICAL
SIMULATION OF OPTICAL WAVE PROPAGATION

by

DIEGO ALBERTO LOZANO JIMENEZ, BSME

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

ProQuest Number:13424314

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 13424314

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Acknowledgements

This work supported by Air Force Office of Scientific Research (AFOSR) under Grant Number BAA-AFRL-AFOSR-2016-0007, and Department of Mechanical Engineering and Computational Science Program at University of Texas at El Paso. Simulations were running on XSEDE computational resources such as Bridges, Stampede, and Lonestar. I would also like to thank Dr. Stacie Williams, AFOSR Program Officer, for her hard work that helped provide the grant. Plane wave simulation and study was conducted at AFRL in Maui with Dr. Rao Gudimetla who was funded in part by an AFOSR Grant. I also want to give a special thanks to Dr. Rao Gudimetla for his patience and support throughout my study. None of this work could not have been possible without the help from Dr. Vinod Kumar and Dr. V. M. Krushnarao Kottedda.

Table of Contents

Acknowledgements	iv
Table of Contents	v
List of Tables	vii
List of Figures	viii
Chapter 1: Introduction	1
Chapter 2: The Scales of Turbulent Motion	3
2.1 Kolmogorov Turbulence	3
2.2 Non-Kolmogorov Turbulence	11
Chapter 3: Numerical Simulation of Laser Propagation in MATLAB	15
3.1 Statistical Metrics	19
Chapter 4: Accelerating Numerical Simulation with Neural Networks	27
4.1 Introduction to TensorFlow	28
4.2 Supervised Algorithms in TensorFlow	29
4.3 Simple Neural Network	30
4.4 Convolutional Neural Network	35
4.5 Reasoning Behind Implementing a Deep Neural Network	38
4.6 Conditional Generative Adversarial Network	41
4.7 Phase Unwrapping with Images	47
4.8 Phase Unwrapping with Matrices	49
Chapter 5: Governing Equations of Fluid Mechanics	53
5.1 Mass Conservation Equation	56
5.2 The Equation of Motion	57
Chapter 6: Computational Fluid Dynamics for Laser Propagation	60
6.1 Finite Volume Method	61
6.2 Large Eddy Simulation	66
6.3 Wall-Adapting Local Eddy-Viscosity (WALE) Model	68
6.4 Implementing LES on Current Numerical Simulation	69
6.5 Calculating for Structure Function Constant	73

Chapter 7: Putting it All Together	77
References	80
Vita	84

List of Tables

Table 6.1: Expanded coefficients for discretized continuity equation	65
Table 6.2 Geometry of computational domain and inlet velocity	70

List of Figures

Figure 2.1: Description of Kolmogorov energy cascade theory	4
Figure 3.1: Schematic of Laser Propagation between Δz	16
Figure 3.2: Wrapped and Unwrapped Phase Screens using the MATLAB embedded code.....	17
Figure 3.3: Wrapped and Unwrapped Phase Screens using the 2D unwrapping algorithm	17
Figure 3.4: Intensity variance in the Kolmogorov spectrum	19
Figure 3.5: Variance of intensity, and number of zeros with respect to Rytov at varying value of alpha.....	20
Figure 3.6: Phase variance in the Kolmogorov scale compared to Wheelon's approximation	21
Figure 3.7: Phase Variance with respect to $\sigma\chi^2$ at varying value of α	22
Figure 3.8: 2D Goldstein Unwrapping Algorithm breaking at high levels of atmospheric turbulence.....	23
Figure 3.9: Phase Structure Function with respect to ρ at varying level of $\sigma\chi^2$; Left to Right, Top to Bottom: $\alpha = 236, \alpha = 226, \alpha = 216, \alpha = 206, \alpha = 196$	25
Figure 4.1: Graphical representation of a simple function	28
Figure 4.2: TensorFlow supervised algorithm flow chart.....	30
Figure 4.3: Neural network node.	30
Figure 4.4: Sigmoid Function (Left), ReLu Function (Right)	31
Figure 4.5: Simple Neural Network with an input layer, hidden layer and an output layer	32
Figure 4.6: Convolution Layer with only 2 filters	36
Figure 4.7: Max Pooling of a 4x4 matrix to a 2x2 matrix	37
Figure 4.8: U-Net Model for Image Segmentation [32]	37
Figure 4.9: Goldstein Algorithm Compute Time relative to spatial grid size at varying levels of atmospheric turbulence intensity	40
Figure 4.10: Conditional Generative Adversarial Network for Phase Unwrapping.....	42
Figure 4.11: Generator Architecture.....	43
Figure 4.12: Encoder Structure.....	44
Figure 4.13: Decoder Structure.....	45
Figure 4.14: Discriminator Architecture.....	46
Figure 4.15: (Left) Training for the discriminator (Right) Training for the generator.....	47
Figure 4.16: Training Image Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image.....	48
Figure 4.17: Validation Image Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image.....	49
Figure 4.18: Validation Matrix Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix	49
Figure 4.19: Validation Matrix Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image	50
Figure 4.20: 4x4 Transpose convolution with a stride of 2	51
Figure 4.21: Modified cGAN Training Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix.....	51
Figure 4.22: Modified cGAN Validation Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix.....	52
Figure 5.1: Two-Dimensional shearing of a block: Top (Lagrangian) and Bottom (Eularian)....	54
Figure 5.2: Fluid particle moving across a controlled volume	56

Figure 6.1: Various Models of CFD	60
Figure 6.2: Schematic of computational domain	70
Figure 6.3: Temperature profile of our simplified 3D model	74
Figure 6.4: Calculations of the Structure Function Constant at, $t = 200$, $t = 400$, and $t = 600$	75

Chapter 1: Introduction

There is a growing interest in optical systems capable of propagating lasers over long distances in atmospheric conditions for a variety of applications. Improvement and understanding of laser propagation over a long path can have influences on the performance of long-range communication systems, active imaging, laser countermeasures, target tracking, and other related optical systems. The challenge is being able to predict atmospheric turbulence induced effects on scintillations, beam spreading, phase fluctuations, beam wander and jitter that can degrade and limit the performance of imaging and laser systems. The fluctuations are caused by temperature variations in the atmosphere associated with turbulent eddies. Atmospheric turbulence behavior is chaotic, and it is impossible to exactly determine the specific value of an optical parameter at all positions in time and space. That is why atmospheric turbulence-induced effects are modeled by numerical and theoretical statistical models to describe the random behavior.

In this study, we want to simulate long-range laser propagation in atmospheric turbulence. The numerical simulations are carried out to study the impact of strong atmospheric turbulence in spatial, temporal, and related spectral domains. The first section of this study will be concerned with modeling this numerical simulation in Kolmogorov and non-Kolmogorov spectrum. To validate our numerical simulation, we will compare the statistical parameter to theoretical approximation in both Kolmogorov and non-Kolmogorov spectrums. Once the code is validated, we want to integrate Large Eddy

Simulation (LES) turbulence modeling. LES simulations allow us to study strong fluid turbulence and can predict advection-dissipation at various energy spectrums. Therefore, this choice of modeling allows us to characterize turbulence at outer and inner regimes important for correcting the effects of optical turbulence over long-distance laser beam propagation. We conduct Computational Fluid Dynamics (CFD) simulations of atmospheric turbulence to realistic three-dimensional systems that better represent the geography of the summits between Maui and Hawaii, where long-range laser propagation measurements were conducted.

LES simulations are computationally expensive and need high wall time and computational power. To overcome that, we implement a conditional generative adversarial network (cGAN) that applies phase unwrapping to a wrapped phase screen. The cGAN framework establishes competition between two distinct players in the model, the discriminator and the generator. The network is trained on 512x512 wrapped and unwrapped images and learns the mapping to conduct phase unwrapping. This portion of the study focuses on employing neural networks to improve the accuracy of current simulation. The focus of this study is to accurately simulate laser propagation through the atmosphere via our knowledge of fluid mechanics and machine learning.

Chapter 2: The Scales of Turbulent Motion

When instabilities are introduced in ambient wind fields, eddies of varying sizes are created. The larger eddies will begin to subdivide onto smaller eddies until their energy is dissipated as heat. Energy dissipation is proportional to the kinematic viscosity, which is related to Reynolds number, Re , which describes the type of fluid flow [1]. To comprehend the complexity of atmospheric turbulence we analyze its behavior through incompressible Navier Stokes equations in nondimensional form.

$$\nabla u = 0 \quad (2.1)$$

$$\frac{\partial u}{\partial t} + u \cdot \nabla u_i = -\nabla p + \frac{1}{Re} \Delta u + f_j \quad (2.2)$$

Reynolds number, $Re = \frac{UL}{\nu}$, is the only non-dimensional parameter in the Navier Stokes equations if the dimensionless body force f_j is absent. In case of atmospheric turbulence, Reynolds number lies between 10^6 and 10^7 . This means that ν , the kinematic viscosity of the medium is relatively small and the flow is non-linear and advection dominated [2]. This results in producing approximately 10^{14} to 10^{16} nonlinear ordinary differential equations and they must be solved numerically. Implementing Computational Fluid Dynamics (CFD) would be ideal but the flow is very nonlinear, and the computational power needed to model the large geometry associated with long distance propagation is not feasible for our current computers. Therefore, atmospheric turbulence-induced effects in adaptive optics are modeled by numerical and theoretical

statistical models to describe its behavior. To characterize atmospheric turbulence, it is common to analyze the effects of turbulent eddies by modeling the random behaviors through the Kolmogorov statistical approach.

2.1 Kolmogorov Turbulence

The Kolmogorov model states that the kinetic energy of a turbulent flow is transmitted within an inertial sub range called the energy cascade theory [1]. The inertial subrange begins with eddies the size equal to that of the outer scale length L_o . The input energy is then progressively divided and redistributed into smaller eddies until the eddies are equal in size to the inner scale length l_o (Figure 1.1). Anything below this length is dissipated into heat. Eddies within this inertial range are assumed to be statistically homogeneous and isotropic.

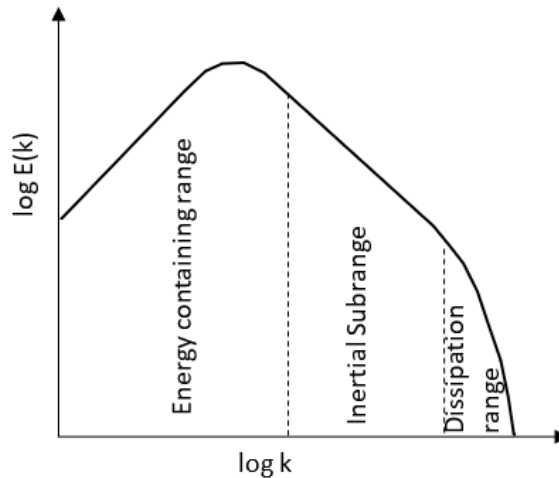


Figure 2.1: Description of Kolmogorov energy cascade theory

In this inertial subrange of local homogeneity, the Kolmogorov theory states that the time variation of two different velocities at two varying points statistically depend only on the displacement vector, r , between them [3]. This simplifies our analysis by implying that the magnitude of the displacement vector is most critical variable when considering the velocity

difference. The assumption led to the derivation of the structure-function which is the structure-function of wind velocity component parallel to the displacement.

$$D_v(r) = \langle [v_r(r_1 + r) - v_r(r_1)]^2 \rangle \quad (2.3)$$

If the separation displacement, r , is within the inertial subrange of turbulence, the velocity structure function follows the form

$$D_v(r) = C_v^2 r^{2/3} \quad l_o < r < L_o \quad (2.4)$$

C_v^2 is the velocity structure constant, which is a measure of the total amount of energy in the turbulent flow. Between the bounds of the inner and outer scale, the eddies may be considered isotropic and homogenous.

Tatarskii related the turbulence model to the index of refractivity through the concept of conservative passive additives. It can be shown [4], [5], that the refractive index has the same statistical properties as and obey the same structure-function laws as velocity and potential temperature.

$$D_n(r) = C_n^2 r^{2/3} = 8\pi \int_0^\infty \kappa^2 \Phi_n(\kappa) \left(1 - \frac{\sin(\kappa r)}{\kappa r} \right) d\kappa. \quad (2.5)$$

Now that turbulence structure function has been analytically derived we can use it to determine Kolmogorov's power spectral density function [6]. One can calculate the power spectral density given the structure-function in terms of the spectral density and inverting the equation to solve for the spectral density.

$$\Phi_n(\kappa) = \frac{1}{4\pi^2 \kappa^2} \int_0^\infty \frac{\sin(\kappa r)}{\kappa r} \frac{d}{dr} \left[r^2 \frac{d}{dr} D_n(r) \right] dr \quad (2.6)$$

κ is the angular spatial frequency in rad/m and is equal to $2\pi/\lambda$. Then by substituting equation (2.5) onto equation (2.6), we can obtain the spectral density function in terms of the Kolmogorov's inertial subrange.

$$\Phi_n(\kappa) = 0.033 C_n^2 \kappa^{-11/3} \quad \frac{2\pi}{L_o} < \kappa < \frac{2\pi}{l_o} \quad (2.7)$$

$\Phi_n(\kappa)$ is the Kolmogorov's power spectral density as a function of the spatial wavenumber κ , which is valid only in the inertial subrange. The probability density function is the probability measure associated with the atmospheric turbulence. Because there have been observations of atmospheric turbulence showing behavior that deviate from the Kolmogorov theory, it has become necessary to develop a statistical model that better represents the effects of atmospheric turbulence.

2.1.1 The Effects of Turbulence on Optical Propagation

Now that we have characterized the atmospheric induced effects on the refractive index, we can look at a propagation of a flat wavefront through turbulence. The phase shift produced by the refractive index fluctuation at an atmospheric layer thickness δh at a height h is

$$\varphi(x) = k \int_h^{h+\delta h} dz n(x, z). \quad (2.8)$$

Other independent variables can affect the phase shift other than the refractive index, for layers thicker than the individual turbulent cell. That is why the Central Limit Theorem is implied, meaning that the phase shifts have a Gaussian statistic. Given the statistical properties of the refractive index, we can proceed to derive the statistical behavior of a wavefront, $U = e^{i\varphi(x)}$. In order to do this, we need to express the coherence function $B_h(\rho)$ in terms of the phase structure function [7].

$$B_h(r) = \langle U(x)U^*(x+r) \rangle$$

$$B_h(r) = \langle \exp(i[\varphi(x) - \varphi(x+r)]) \rangle \quad (2.9)$$

Since we have stated previously that we have implemented the Central Limit Theorem, $[\varphi(x) - \varphi(x+r)]$, has Gaussian statistics with zero mean. We can then apply the relation $\langle \exp(\alpha X) \rangle = \exp\left(\frac{1}{2}\alpha^2 \langle X^2 \rangle\right)$.

$$B_h(\rho) = \exp\left(-\frac{1}{2}\langle |\varphi(x) - \varphi(x+r)|^2 \rangle\right) \quad (2.10)$$

$$B_h(\rho) = \exp\left(-\frac{1}{2}D_\varphi(r)\right)$$

Due to the chaotic behavior of atmospheric turbulence, random processes like velocity, or temperature cannot be approximated as being stationary. We can still statistically analyze these random processes if they have stationary increments. Instead of

focusing on the random process over time or a point in space $x(r)$, we focus on the function $x(r + r_1) - x(r_1)$, which behaves like a stationary process [8].

The random process that has stationary increments and that can be described by such function are better analyzed through the phase structure function. It is customary when studying atmospheric turbulence to express a random process as a sum.

$$x(r) = x(r) + x_1(r) \quad (2.11)$$

The mean phase of our signal is defined by $m(r)$, and $x_1(r)$ is the stationary process with moments that are invariant under translations in space $\langle x_1(r) \rangle = 0$. The structure function for a random process can be defined by

$$D_x(r_1, r_2) = [m(r_1) - m(r_2)]^2 + \langle [x_1(r_1) - x_1(r_2)]^2 \rangle. \quad (2.12)$$

Previously we established that the random process has a slow varying mean, meaning that the difference in means is nearly zero. We can then approximate the structure-function to be

$$D_x(r_1, r_2) \cong \langle [x_1(r_1) - x_1(r_2)]^2 \rangle. \quad (2.13)$$

Let's consider a random process such that its ensemble average of $x(r + \rho) - x(r)$ is independent of displacement, r . Then it would be true that the ensemble average of $[x(r + \rho) - x(r)]^2$ is independent of displacement. We can then call this random process to

have stationary increments and can be characterized by the structure function. However, a stationary process can be considered a process with stationary increments in some special cases. For instance, the phase fluctuations are a stationary process and its structure function is closely related to its covariance B_φ .

$$D_\varphi(\rho) = \langle (\varphi_1 - \varphi_2)^2 \rangle = 2[\langle \varphi^2 \rangle - \langle \varphi_1 \varphi_2 \rangle] = 2[B_\varphi(0) - B_\varphi(\rho)] \quad (2.14)$$

The covariance function is defined by the ensemble average.

$$B_\varphi(\rho) = \langle \varphi(r) \varphi(r + \rho) \rangle \quad (2.15)$$

For an atmospheric layer thickness, δh , larger than the correlation scale of the fluctuations we can determine the phase covariance at an extended bound of $-\infty$ and ∞ . This allows us to relate the refractive index variations $B_n(\rho, z)$ to the phase covariance.

$$B_\varphi(\rho) = k^2 \delta h \int_{-\infty}^{\infty} dz B_n(\rho, z). \quad (2.16)$$

Given both equations (2.14) and (2.16) we can derive an analytical approximation of the phase structure function in terms of the refractive index structure constant, C_n^2 .

$$D_\varphi(\rho) = 2[B_\varphi(0) - B_\varphi(\rho)] \quad (2.17)$$

$$\begin{aligned}
&= 2k^2\delta h \int_{-\infty}^{\infty} dz [B_n(0, z) - B_n(\rho, z)] \\
&= 2k^2\delta h \int_{-\infty}^{\infty} dz [(B_n(0, 0) - B_n(\rho, z)) - (B_n(0, 0) - B_n(0, z))] \\
&= k^2\delta h \int_{-\infty}^{\infty} dz [D_n(\rho, z) - D_n(0, z)]
\end{aligned}$$

Inserting equation (2.5) and integrative over the bounds gives

$$D_\varphi(\rho) = 2.914k^2\delta h C_n^2 \rho^{5/3}. \quad (2.18)$$

Now that we have the desired expression of the phase structure function in Kolmogorov turbulence, we can proceed to derive the coherence function.

$$B_h(\rho) = \exp\left(-\frac{1}{2}[2.914k^2\delta h C_n^2 \rho^{5/3}]\right) \quad (2.19)$$

We need to consider that in adaptive optics most the time we are not looking vertically into the sky and need to take into account the zenith angle $\mathbf{z}$. The expression can then be integrated over the whole atmosphere.

$$B_h(\rho) = \exp\left(-\frac{1}{2}\left[2.914k^2 \sec(z) \rho^{5/3} \int \delta h C_n^2(h) \right]\right) \quad (2.20)$$

In order to simplify the expression, we can introduce the Fried Parameter

$$r_o = \left[0.423 k^2 \sec(z) \int \delta h C_n^2(h) \right]^{-3/5} \quad (2.21)$$

The coherence function and structure function can be simplified by now implementing Fried Parameter.

$$B_h(\rho) = \exp \left[-3.44 \left(\frac{\rho}{r_o} \right)^{5/3} \right] \quad , \quad D_\phi(\rho) = 6.88 \left(\frac{\rho}{r_o} \right)^{5/3} \quad (2.22)$$

Both the coherence function and the phase structure function both depend on the Fried Parameter which is a function of turbulence strength, zenith angle, and wavelength [7]. This is the importance of determining the phase structure function in our simulation. It will be used to determine a complex parameter that will characterize the effects of atmospheric turbulence on laser propagation such as the Fried parameter. As an example, observations with telescopes smaller than r_o , are essentially diffraction-limited. Telescopes much larger than r_o , are seeing-limited. More complex parameters are far off the score of this study as this is only an example to show the importance of the phase structure function.

2.2 NON-KOLMOGOROV TURBULENCE

Current analysis of atmospheric turbulence effects is based on theoretical and statistical models [3], [5], [9] derived from the Kolmogorov approach. Supporting experimental evidence [1], [10], [11] helped validate the classical theory of optical propagation. This is the reason Kolmogorov's power spectrum of refractive index fluctuation is widely employed in the study of optical wave propagation in the atmosphere. However, inaccuracies develop in the Kolmogorov model at various

conditions in atmospheric turbulence, which shows behavior that deviates from the theory [12]–[15]. Further investigation is needed to comprehend laser propagation in atmospheric turbulence that does not obey the Kolmogorov statistics.

The beta-model incorporates space-filling concepts for the turbulent eddies in the inertial ranges using fractal descriptions for the eddies. That is why we need to understand non-Kolmogorov descriptions of energy cascade theory and comprehend if it is better suited to model laser propagation over long distances. The spectral behavior of the refractive index fluctuations changes with altitude, which makes the Kolmogorov model inadequate for predicting atmospheric behaviors. Instead, we will implement a 3D power spectrum of refractive index fluctuations in terms of the arbitrary power law [16], which can be defined as

$$\phi_n(\kappa, \alpha) = a(\alpha)\beta\kappa^{-\alpha} \quad (2.23)$$

where κ is the spatial wavenumber, β is the general refractive index structure constant is like C_n^2 and has units of $m^{3-\alpha}$. $a(\alpha)$ is the function that maintains consistency between the index structure function and its power spectrum. To be able to use the 3D power spectrum, we need to determine the solution of $a(\alpha)$. Let's consider a structure function that describes a random media with an index of refraction fluctuations with the form of a power law [17].

$$D_n(r) = \beta r^\gamma \quad (2.24)$$

By using the relation derived from Tatarski [5], which relates the index power spectrum to the derivative of the structure-function, we can determine the power spectrum in terms of the arbitrary power law [18]. The structure function of the form of a power law is implemented onto equation (2.6) to derive the power spectrum of the turbulence in terms of the arbitrary power law

$$\phi_n(\kappa, \alpha) = 2^{\alpha-6} \beta (\alpha^2 - 5\alpha + 6) \pi^{-2/3} \frac{\Gamma\left[\frac{\alpha-2}{2}\right]}{\Gamma\left[\frac{5-\alpha}{2}\right]} \kappa^{-\alpha} \quad (2.25)$$

Γ is Euler's gamma function and $a(\alpha)$ can be determine by equating equation (2.23) and equation (2.25) and algebraically solving for $a(\alpha)$.

$$a(\alpha) = 2^{\alpha-6} (\alpha^2 - 5\alpha + 6) \pi^{-2/3} \frac{\Gamma\left[\frac{\alpha-2}{2}\right]}{\Gamma\left[\frac{5-\alpha}{2}\right]} \quad (2.26)$$

Stribling [18] showed that values of α range from $3 < \alpha < 4$ and, when $\alpha = 11/3$, equation 9 reduces to the Kolmogorov power spectrum. When $\alpha \rightarrow 3$, values for the consistency function approach zero and the turbulence power spectrum vanishes. For large outer scale turbulence behavior, adjustment to power spectrum as proposed by von Karman can be used and is given by

$$\phi_n^{vK}(\kappa, \alpha) = a(\alpha) \beta (\kappa^2 + \kappa_0^2)^{-\alpha/2}. \quad (2.27)$$

where $\kappa_0 = 2\pi/L_0$.

Chapter 3: Numerical Simulation of Laser Propagation in MATLAB

We will begin our simulation by first implementing Kolmogorov and when validated move towards a non-Kolmogorov model to simulate optical wave propagation. For simplicity, we will be propagating a plane wave over 10km to derive statistical parameters that will characterize atmospheric turbulence effects. The simulated laser propagation will have a wavelength of $0.532\mu\text{m}$. Levels of turbulence intensity will be represented by a constant Rytov index and will vary through each simulation from a range of $0.1 \leq \sigma_x^2 \leq 1.0$. The Rytov index is the log-amplitude variance to describe the strength of scintillations and is related to C_n^2 by

$$\sigma_x^2 = 0.312C_n^2 k^{7/6} L^{11/6} \quad (3.1)$$

The inner scale and outer scale length are 5mm and 10m, respectively. Due to computational constraints, the grid size will be equivalent to 10.24m with a 2048x2048 number of samples, N , and a 5mm grid spacing interval, δ . Within each simulation, 50 trials will run, and statistical parameters will be calculated from each trials output phase screen. These statistical parameters will then be averaged by the number of trials to consider the variation within each trial. Propagation will be simulated using Fresnel propagation in a vacuum [19] and introduce atmospheric turbulence effects through the phase screen approach.

$$U(x_2, y_2) = \frac{e^{ik\Delta z}}{i\lambda\Delta z} \iint_{-\infty}^{\infty} U(x_1, y_1) e^{i\frac{k}{2\Delta z}(x_1^2 + y_1^2)} dx_1 dy_1 \quad (3.2)$$

The phase screen approach introduces atmospheric turbulence by subdividing the propagation length into uniform slabs of distance Δz . Initially within the propagation range $0 < L < \Delta z$, the plane wave will propagate linearly using Fresnel propagation in a vacuum. Atmospheric turbulence effects will be implemented when the plane wave reaches $L = \Delta z$. The plane wave is altered by a turbulent-induced phase screen in the Fourier domain

$$U = U_0 e^{-i\Phi_\phi(\kappa)} \quad (3.3)$$

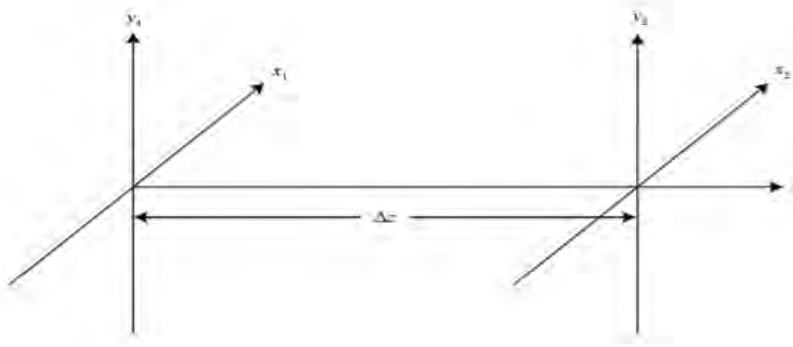


Figure 3.1: Schematic of Laser Propagation between Δz

The process will repeat itself between each slab subsection until the plane wave is transmitted throughout the whole propagation length. Each phase spatial power spectral density (PSD), $\Phi_\phi(\kappa, \alpha)$, can be expressed in terms of its refractive index spectrum, Φ_n^{vk} .

$$\Phi_\phi(\kappa, \alpha) = 2\pi^2 k^2 \Delta z \Phi_n^{vk}(\kappa, \alpha) \quad (3.4)$$

Phase screen simulations are generated through random numbers that are taken from a Gaussian distribution. These values are then implemented into two-dimensional arrays of phase values on a grid of sample points that have the same statistical values as turbulence-induced phase variations. The simulations are conducted using 15 screens over a 10km propagation path.

To derive statistical parameters of the phase fluctuations, the phase needs to be made continuous. Discontinuities are present in the phase at the output signal that wrap around $-\pi < \varphi < \pi$. The built-in MATLAB function can unwrap the phase at laminar flow but shows discontinuities at increased turbulence.

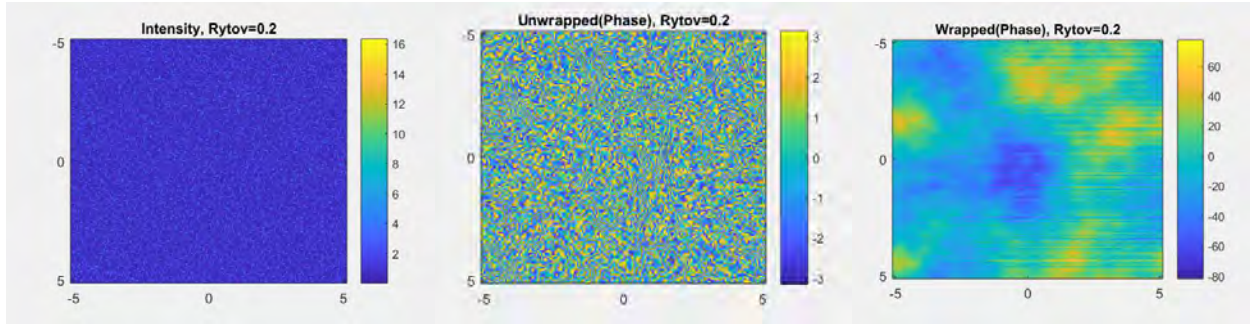


Figure 3.2: Wrapped and Unwrapped Phase Screens using the MATLAB embedded code

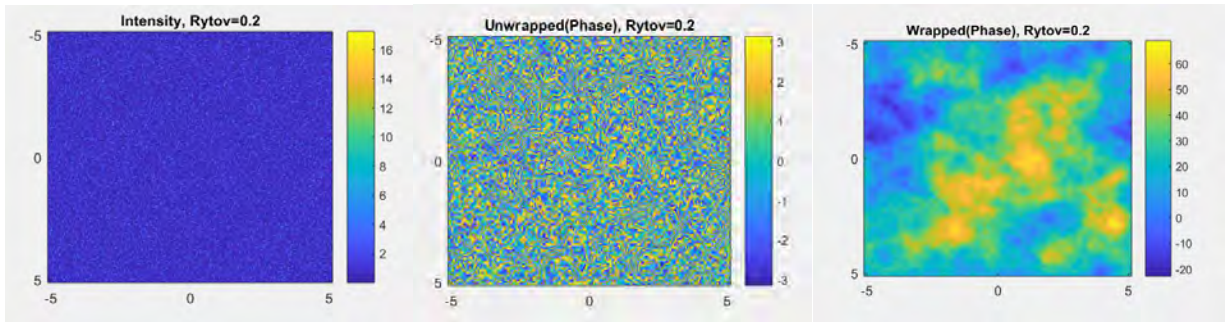


Figure 3.3: Wrapped and Unwrapped Phase Screens using the 2D unwrapping algorithm

A 2D unwrapping algorithm compiled by *Bruce Spottiswoode* [20] and edited by *Carey Smith* is used to make the phase fluctuations continuous. Once the phase is continuous, we are able to calculate statistical parameters like the phase variance. If the simulated phase variance values are validated with the theoretical approximation, we can then proceed to determine the phase structure function that obeys the same structure-function law as the refractive index. For a random process, the covariance is closely related to the autocorrelation by

$$B_{\varphi}(\rho) = R_{\varphi}(\rho) - m^2 \quad (3.5)$$

In this case for a random process like the phase shift, can be considered a stationary process the covariance and autocorrelation are identical. Determining the autocorrelation is often too difficult to compute in the spatial domain but is equivalent to simple multiplication in the frequency domain.

$$R_{\varphi}(\rho) = \mathcal{F}^{-1}\{\mathcal{F}[\varphi(r)]\mathcal{F}[\varphi(r)]^*\} \quad (3.6)$$

The notation $*$ is the complex conjugate of the function and $\rho = |r_1 - r_2|$ is the scalar separation. This will allow for a straight forward approach in determining the phase structure function in the 2-D case. The phase structure function is another statistical measure to describe the random behavior of phase values and is closely related to the phase autocorrelation.

$$D_{\varphi}(\rho) = 2[R_{\varphi}(0) - R_{\varphi}(\rho)] \quad (3.7)$$

3.1 STATISTICAL METRICS

To be able to validate our simulation theoretical calculations states that the intensity linearly increases with increase in Rytov index [21]. The theory states that the increase in variance is approximately four times the Rytov value for weak turbulence. range of $0.1 \leq \sigma_X^2 \leq 0.5$, the simulated values do agree with theoretical calculations. However, once $\sigma_X^2 > 0.5$, simulated values do not agree with the theoretical results and the intensity variance saturates at increased turbulence.

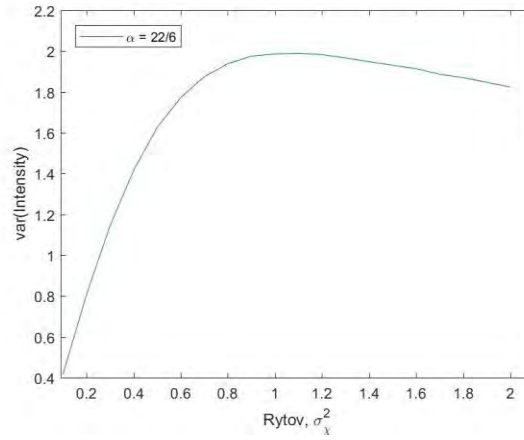


Figure 3.4: Intensity variance in the Kolmogorov spectrum

We wanted to continue our examination by determining the impact of α on the variance of intensity and a new parameter the zero-intensity value. This new parameter is any intensity value less than one percent of the maximum value of intensity, defined as the zero-intensity value.

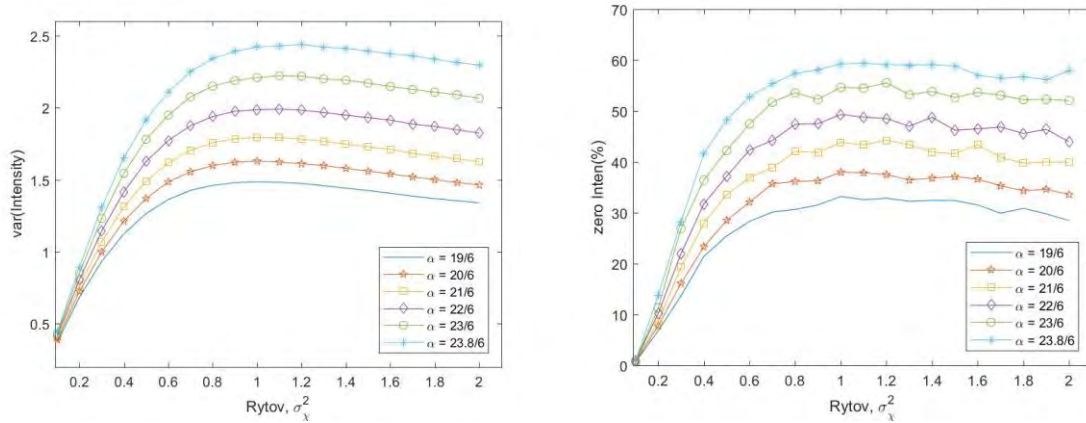


Figure 3.5. Variance of intensity, and number of zeros with respect to Rytov at varying value of alpha

To determine the impact of α on the statistical parameters, phase-screen simulations were performed from a range values of $3 < \alpha < 4$, at varying turbulence levels defined by the Rytov. We notice that the variance of intensity saturates for all values of α , but the point of saturation increases for larger values of α as shown in Figure 6 (Left). Like the variance of intensity, the zero-intensity value saturates for all values of α , but different points for larger values of α as shown in Figure 6 (Right). A noticeable difference between the variance of intensity and the zero-intensity values is that the zero-intensity values appear to fluctuate at different values of α and Rytov. This portion of the study was only to verify if our current numerical simulation is correct. This study mainly focuses on the phase fluctuations at the receiver plane.

Now that we verified our code, we can determine the phase variance at the receiver plane. The phase variance depends on the first moment of the spectrum, meaning that it is determined mainly by the largest eddies. This makes the phase variance sensitive to the small-wavenumber portion of the spectrum of irregularities. Therefore, we modeled our simulation by the von Karman PSD to attempt to represent large eddy irregularities [1]. As an example, during terrestrial experiments, values for L , k and C_n^2 are known. We can then establish a distribution of

outer scale length by determining the average phase variance at the output signal. Given a theoretical approximation, we can algebraically calculate κ_0 and determine L_0 , the outer length. An approximation provided by Dr.V.S Gudimetla [22] helps us determine an approximate phase variance value that is expressed in terms of the non-Kolmogorov power spectrum, $\Phi_n^{vK}(\kappa, \alpha)$.

$$\langle \varphi^2 \rangle = \frac{1}{2(\alpha/2 - 1)} \Gamma(\alpha - 1) \sin\left((\alpha - 3)\frac{\pi}{2}\right) k^2 L C_n^2 \kappa_0^{-\alpha+2} \quad (3.8)$$

Given that at $\alpha = 22/6$ the power spectrum in terms of an arbitrary power law reduces to the Kolmogorov power spectrum. The phase variance in terms of the von Karman power spectrum in the Kolmogorov regime can be reduced to an approximated averaged phase variance [1] $\langle \varphi^2 \rangle = 0.782 L k^2 C_n^2 \kappa_0^{-5/3}$.

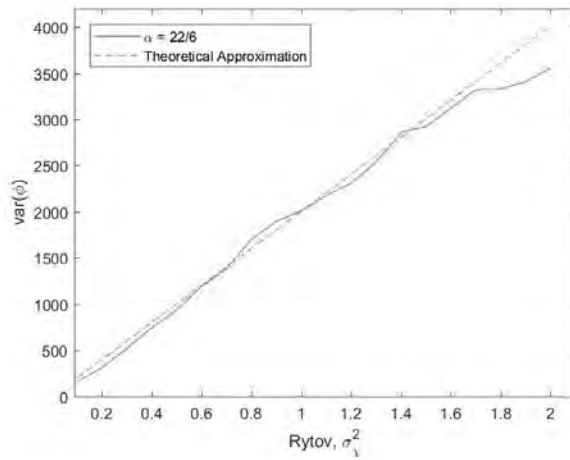


Figure 3.6: Phase variance in the Kolmogorov scale compared to Wheelon's approximation

When only looking at the Kolmogorov spectrum at an outer scale length of 10m, limitations do appear once it reaches $\sigma_\chi^2 \leq 1.6$. This can be attributed to the finite spatial dynamic range imposed by the maximum available grid size. This is because the von Kármán power spectrum has more influence in low spatial frequencies that are attributed to the outer scale length effects. The spatial grid needs to be larger so the sample frequency spacings can be low enough to accurately represent the outer scale effects. Another factor that can improve on the discrepancies is the number of trials within each simulation. To be able to reduce the variations of the mean phase variance, we would need to have a significantly larger number of trials. This would reduce the effects caused by any outliers that vary from the mean.

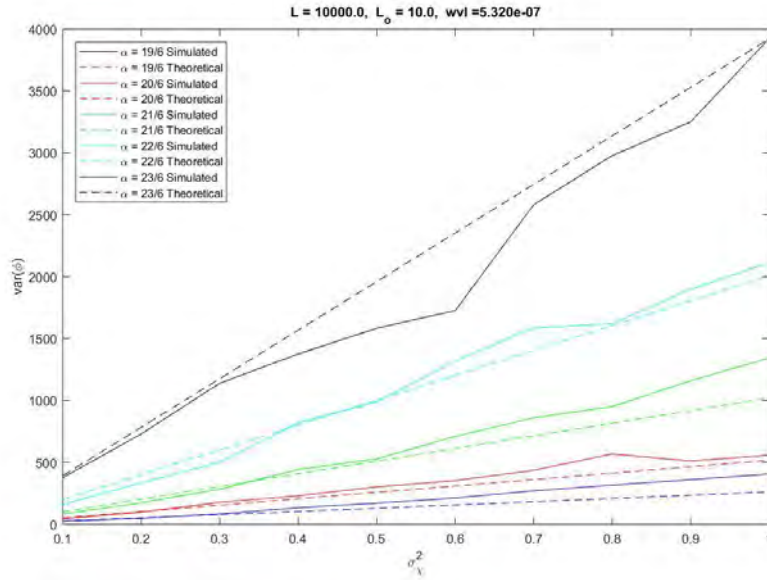


Figure 3.7: Phase Variance with respect to σ_χ^2 at varying value of α

When looking at varying levels of α the phase variance does not saturate and increases linearly with respect to Rytov. However, fluctuations do become more apparent at $\sigma_\chi^2 \geq 0.5$ and for greater values of α . This does show promising results as these values do not saturate. Aside

from the number of simulations within each trial, the fluctuations can also be attributed to the 2D Goldstein unwrapping function. At a high level of turbulence, the algorithm breaks down and show discontinuities in the image.

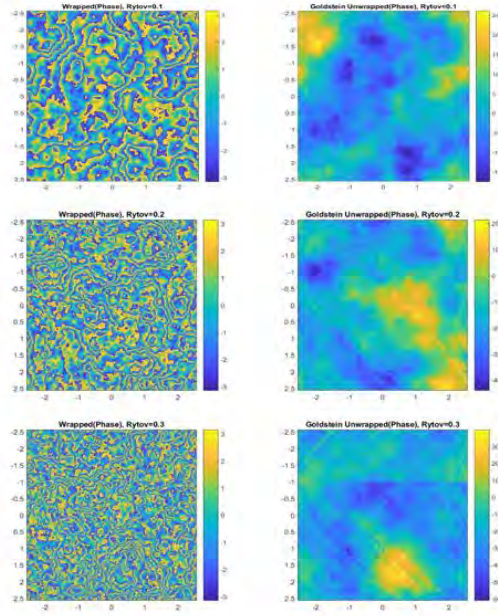


Figure 3.8: 2D Goldstein Unwrapping Algorithm breaking at high levels of atmospheric turbulence

We can continue our analysis by deriving parameters that better describe the strength of the phase fluctuations in turbulence bellow $\sigma_x^2 \leq 0.5$. Note that these simulations were obtained with 50 trials at relatively smaller spatial grid size (2048x2048), which may not be sufficient for higher levels of turbulence. The spatial grid needs to be larger to accurately represent the outer scale effects and increase the number of trials within each simulation to reduce any variations from the mean variance.

In order to calculate the phase structure function, we need to determine the phase covariance. But since the phase fluctuations is a stationary random process, we can determine the phase covariance by calculating the phase autocorrelation. The

autocorrelation values are calculated in the Fourier domain using equation (3.5). This results in a 2-D plane but only a vector from the center of the plane in the positive x-axis. This vector is now used in equation (3.6) to determine the phase structure function. This is done for simplicity and to compare to 1-D theoretical values of the phase structure function [22].

$$D_{\varphi}(\rho) = 2C_{n\kappa}^2 k^2 L \Gamma(\alpha - 1) \sin\left((\alpha - 3)\frac{\pi}{2}\right) \left[\frac{\kappa_0^{2-\alpha}}{\alpha - 2} - \frac{\alpha 2^{-\alpha/2}}{\Gamma(1 + \alpha/2)} \left(\frac{\rho}{\kappa_0}\right)^{\alpha/2-1} K_{1-\alpha/2}(\kappa_0 \rho) \right] \quad (3.9)$$

In this equation $K_{1-\alpha/2}$ is the modified Bessel function of the second kind and Γ is the gamma function. This function allows us to determine the phase structure function various levels of α .

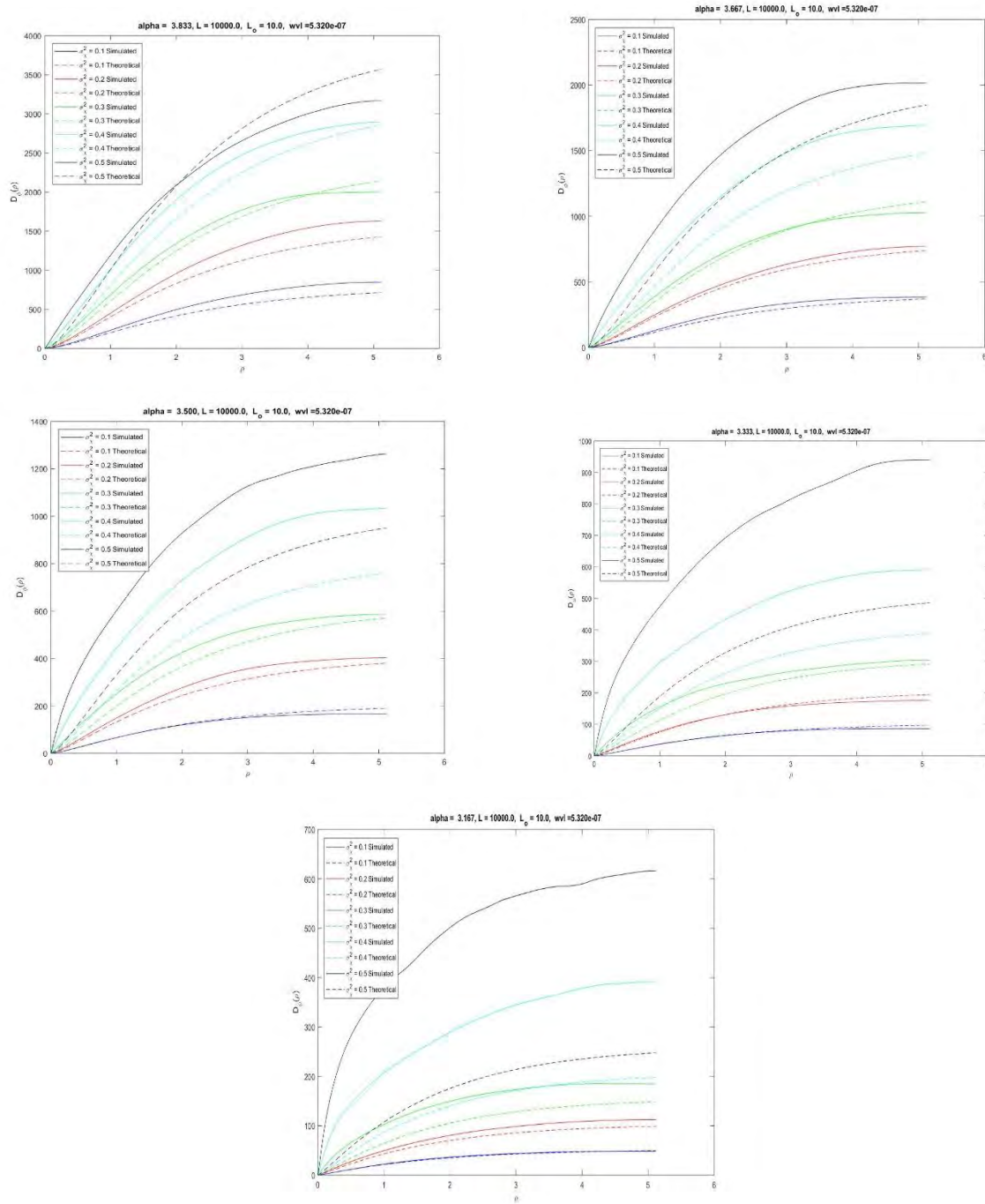


Figure 3.9: Phase Structure Function with respect to ρ at varying level of σ_X^2 ; Left to Right, Top to Bottom: $\alpha = 23/6, \alpha = 22/6, \alpha = 21/6, \alpha = 20/6, \alpha = 19/6$

Progressing from phase variance to the phase structure function, due to the 2D Goldstein algorithm become more prominent at $\sigma_X^2 \geq 0.4$. Rytov values above this range

will still have simulated phase structure function that are larger in value than that of the theoretical, for all values of α . Within the range of $0.1 \leq \sigma_x^2 \leq 0.3$, the phase structure function is approximately the same as the calculated theoretical values for all values of α , except $\alpha = 19/6$. However, at $\alpha = 23/6$ we seem to have a better approximation but worsen as we begin to decrease the value of α and reach $\alpha = 19/6$. Since the phase variance, autocorrelation and structure function are all closely related I believe these discrepancies are due to the unwrapping algorithm. Since there are more discontinuities it is a possibility that it is increasing the simulated value of the phase variance and so on. One method to overcome this problem is by increasing the grid size which has proven to produce better quality unwrapped images.

Currently, we can only characterize laser propagation at low turbulence. This does not mean the Kolmogorov and non-Kolmogorov model breaks apart at high turbulence, but further investigations are needed. We can increase the grid size from 2048x2048 to be 4096x4096. However, this increase in grid size is computationally expensive and would require significant amount of time to run these simulations. The ideal goal would be to have grid size of 8192x8192 to keep the grid spacing relatively small and still incorporate the outer scale effects.

With High Performance Computing (HPC), we can begin simulation with a screen size large enough that will allow for deep turbulence effects for a laser propagating over a long path. We can also train a deep neural network that can learn phase unwrapping. This will accelerate our algorithm if we are successful in training the model and integrating it with our current model. We can also avoid the discrepancies caused by the 2D Goldstein Algorithm at high atmospheric turbulence. Our attention will shift into training a deep neural network with the wrapped and unwrapped phase screens, to learn phase unwrapping.

Chapter 4: Accelerating Numerical Simulation with Neural Networks

Machine learning enables statistics and computer science to learn how to conduct a specific task without being programmed to do so. As for example we can train machine learning algorithm to recognize the difference between a dog and a cat. It does this by taking in statistical patterns within the data to allow it to make decisions whether that data given is either a dog or a cat. Other machine learning application are face recognition, or spam filters in emails. Machine learning has the characteristic in laying in several different fields, such as artificial intelligence, statistics, computer science and some others [23]. For these simple reasons, TensorFlow library is one of the most interesting tools used during these days, even for solving engineering problems. Many machine learning libraries are available for making the predictions. TensorFlow is based on graphs-based computation to model and be implemented in Python.

TensorFlow is a second-generation system created by Google Brain that can run neural network algorithms at small portable processing systems, as well as, large scalable processing units, such as compositions of CPUs, GPUs, and hybrid systems [24][25]. Its applications are many, such as image processing, sound detection, speech recognition, natural language processing and engineering problems [26]. 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[27][28]. 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[29]. Such as flexible programming, better performance and supports the training of various neural network models[26][30].

4.1 INTRODUCTION TO TENSORFLOW

TensorFlow is based on graph-based computation and it better understood with a given example. When trying to solve a simple calculation such as $a = (1 + c)/(c - b)$, we break down the function into the following concepts.

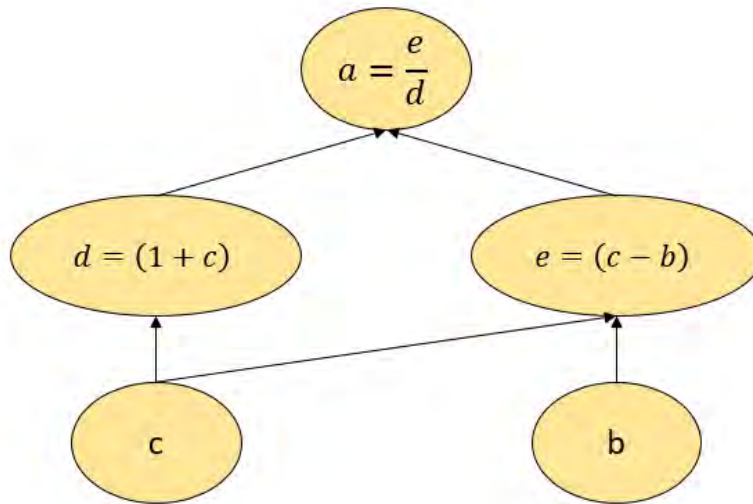


Figure 4.1: Graphical representation of a simple function

This may be over complicating a simple function, but the computations can now be done in parallel. By splitting up the computation across CPUs and GPUs can give a significant speed up time when conducting calculations. This is something that is necessary when implementing machine learning especially in the case of complicated networks. Whether it is a simple function or a complex network the idea remains similar. As a TensorFlow developer, we create the full set

of graph operations, initialize the variables, start a session, feed data into the graph and compute the output of the graph.

4.2 SUPERVISED ALGORITHMS IN TENSORFLOW

Neural networks can be trained unsupervised but, in this study, we will be mainly focusing on the supervised method. 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 depend on datasets. For machine learning applications the data must be normalized. The other requirement is that data must have partitions, which are training, 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 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 to 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.

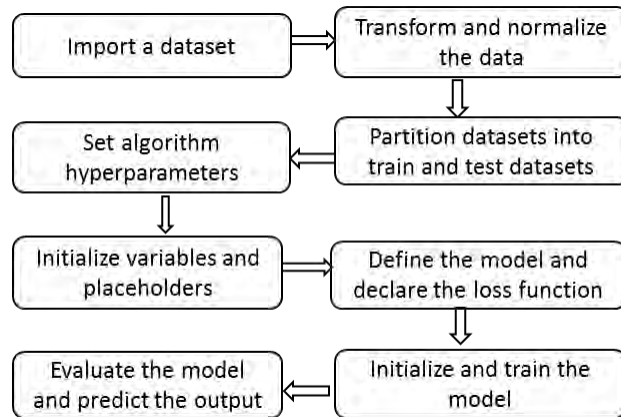


Figure 4.2: TensorFlow supervised algorithm flow chart.

4.3 SIMPLE NEURAL NETWORK

A simple neural network is composed of a variety of nodes that are connected together and communicate with one another. Each node takes in multiple inputs and conducts the dot product with each input weight. It then applies an activation function to the product of the inputs and generates an output.

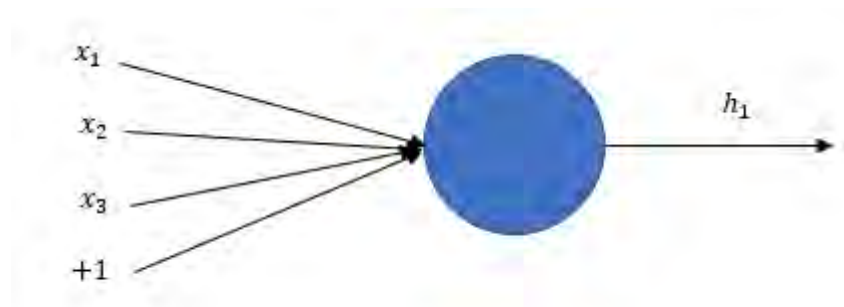


Figure 4.3: Neural network node.

The circle represents the nodes with three inputs and a bias. It begins by taking the dot product of the inputs and weights added by a bias.

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + [b] = x_1w_1 + x_2w_2 + x_3w_3 + b \quad (4.1)$$

This product then used as an input to an activation function. An activation function acts as a switch and depending on what the function is, it can range a value from 0 to 1 or from -1 to 1. It is common practice that most activation function is either sigmoid or ReLu (Rectified Linear Unit). However, we will later see that there is a more complex activation function.

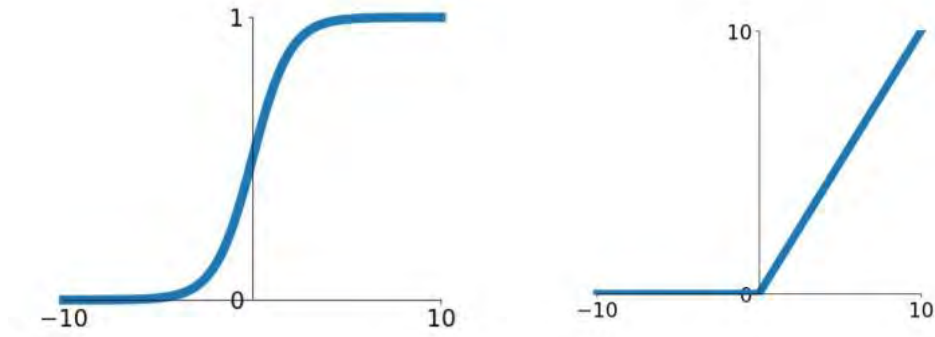


Figure 4.4: Sigmoid Function (Left), ReLu Function (Right)

After the activation function is calculated the output is then determined, denoted by ***h***.

$$h_1 = f(x_1w_1 + x_2w_2 + x_3w_3 + b) \quad (4.2)$$

In this case, the activation function is denoted by ***f*(*)**. When it comes to neural network it easier to remember that there are two components. The is the linear components which is the linear algebra and the non-linear component which is the activation function.

Now to better explain machine learning, it would be better to go over a simple neural network. A common neural network consists of an input layer, a hidden layer, and an output layer.

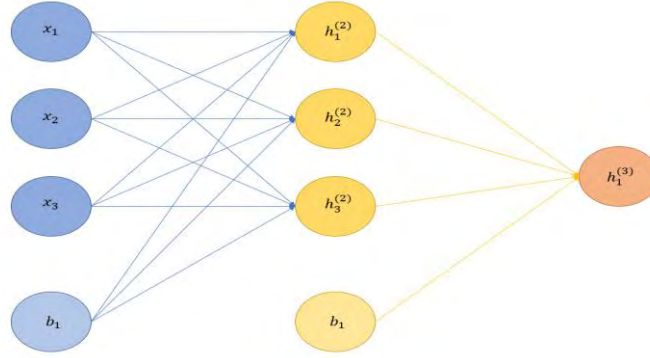


Figure 4.5: Simple Neural Network with an input layer, hidden layer and an output layer

In order to identify the weights, we will use the following notation, $w_{ij}^{(l)}$. The notation l refers to the connection layer and the i notation refers to the node number of the connection in the layer $l + 1$. While the j notation refers to the node number of the connection layer l . So, for the connection between node 1 in layer 1 and node 3 in layer 2, the weight notation would be $w_{31}^{(1)}$. Since the bias is not really a node with an activation function, it has no inputs. The notation for the bias weight is $b_i^{(l)}$, where i is the node number of the connection in the layer $l + 1$. So, the bias notation for the connection between node 1 in layer 1 and node 3 in layer 2, is $b_3^{(1)}$. Finally, the node output notation is $h_j^{(l)}$, where j refers to the node number in layer l . The output of node 2 in layer 2 would be $h_2^{(2)}$.

The feed-forward pass is straightforward. To calculate any of the hidden nodes including the output layer would be like calculating one node but now we have multiple nodes.

$$h^{(l+1)} = f(W^{(l)}h^{(l)} + b^{(l)}) \quad (4.3)$$

The value $\mathbf{W}^{(l)}$ denotes the matrix form of the weights, and the input layer could be also written as $\mathbf{h}^{(1)}$. To make things easier we can expand the equation and solve for the hidden layer in our simple neural network.

$$h^{(2)} = f \left(\begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} \end{bmatrix} \right) \quad (4.4)$$

In this case, it is simply matrix multiplication followed by an activation function to solve the hidden layer and we would repeat this process to solve the output layer. When building a neural network, the key goal is to determine the correct weights for a given problem. At the beginning of constructing a neural network, the weights are initialized through a Gaussian random number generator. The goal is to get the neural network to adjust its weights as to get better predictions at the output node. This is done by varying the weights to minimize the error between the true data set and the predicted value. To do so we need to implement backpropagation and gradient descent.

$$w = w_o - \alpha * \nabla error \quad (4.5)$$

The gradient descent method uses the gradient to make a change in the new value of the weight $\mathbf{w}$. It takes the current position $\mathbf{w}_o$, and adjust it by the gradient error weighted by a step size α . The most important component in machine learning would be the loss function.

To optimize a neural network revolves around minimizing the loss function. As mentioned before when we are conduction supervised training, we want the network to learn how to make predictions off the given data set. For a given training pair, we will denote the input data

as $\mathbf{x}^z$ and the desired output to be $\mathbf{y}^z$. So, an L^2 the loss for a training pair in a neural network can be given as

$$J(w, b, x, y) = \frac{1}{2} \|\mathbf{y}^z - \mathbf{y}_{pred}(\mathbf{x}^z)\|^2 \quad (4.6)$$

Let it be noted that $\mathbf{y}_{pred}(\mathbf{x}^z)$ is the output layer of our neural network and in our case can be written as $\mathbf{h}_1^{(3)}(\mathbf{x}^z)$. To generalize the cost function to include more than one training pair we want to minimize the loss function over all m , training pairs.

$$J(\mathbf{w}, \mathbf{b}) = \frac{1}{m} \sum_{z=0}^m \frac{1}{2} \|\mathbf{y}^z - \mathbf{y}_{pred}(\mathbf{x}^z)\|^2 \quad (4.7)$$

To determine the new weights, the gradient decent for every weight and bias in a neural network becomes

$$w_{ij}^{(l)*} = w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} J(\mathbf{w}, \mathbf{b}) \quad (4.8)$$

$$b_i^{(l)*} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(\mathbf{w}, \mathbf{b}) \quad (4.9)$$

The way neural networks work is that we initialize the weights. Then we feedforward and calculate the predicted value. We then take the predicted value and the true value and determine the loss function. Through backpropagation and gradient decent we adjust the weights and bias. We can see the importance of the activation function from equation (4.8) and equation (4.9). Whenever the output of an activation function is near zero there would be no information to adjust the weights when it is conducting gradient decent. This is how a neural network learns and

begging to exclude unnecessary values. When the weights are not being adjusted we know that our neural network has converged. This may be a simple neural network, but the overall concept remains the same for complex networks.

4.4 CONVOLUTIONAL NEURAL NETWORK

To be able to comprehend complex neural network we need to introduce the idea of a convolutional neural network. Like our previous example of a simple neural network, we are now working with images or matrices. The weights are now introduced as a filter that convolves with the image. The weight is dependent on the kernel size and stride. It is common practice to keep the weights or filters to be square matrices. However, in order to work well each convolutional stage usually has multiple filters.

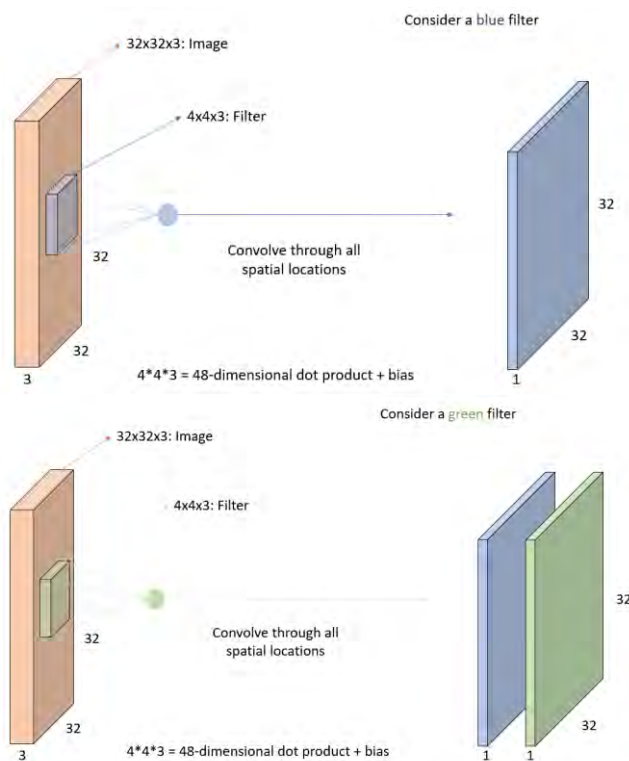


Figure 4.6: Convolution Layer with only 2 filters

Each filter will convolve fully with the image and produce their own 2D output. Each filter has its own set of weights that will be adjusted to train the network. In this example, the filter will convolve fully with the colored image of multiple channels producing an output. This will be repeated depending on the number of filters producing a new 3D matrix. Each convolution output for each node will also be passed through an activation function.

The kernel size is related to the size of the filter/weight and in our example, it is a [4,4] kernel. We can also control how much the filter moves or the stride, whether it be a single space or more. In cases where we might want to downsize an image to reduce the number of parameters and to do so, we can increase the stride. However, if we want to maintain the same dimension of an image but want to increase the stride as to not overlap filter areas, padding may be involved. Dummy nodes have zero value with a rectified linear unit activation of the previous layer it won't have much value when introduced. This is implemented only when we want to maintain the same dimensions as the previous image.

Another important component of the convolutional neural network is called pooling. Some convolutional neural network will implement pooling in order to downsample an image. The reason to downsample is to reduce the number of parameters in a neural network. We also want to make feature detection more complex by making it tolerant to scale and orientation changes. Like convolutional filters, pool filter moves along an image taking only the largest value within the image. However, pool filters do not attain any weights and are better explained through figure (4.7).

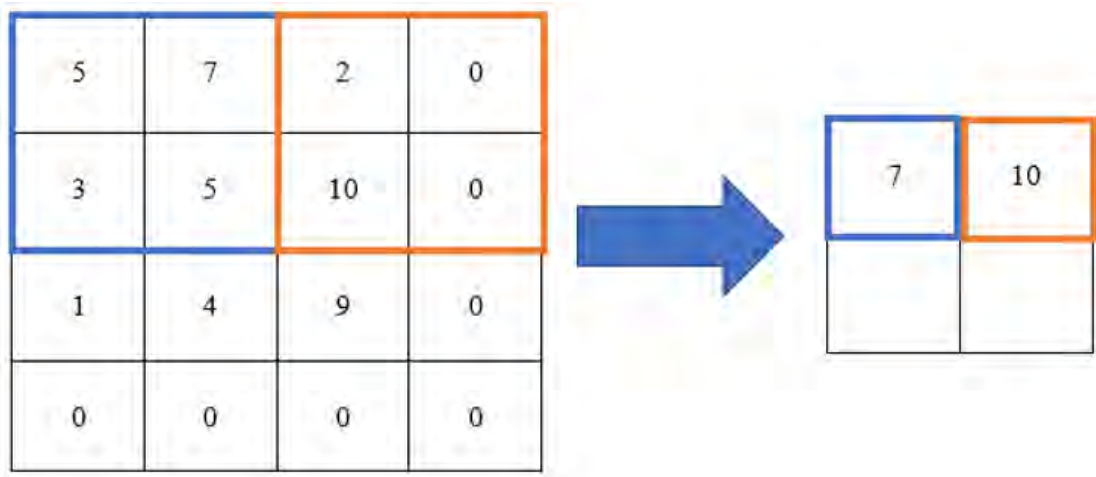


Figure 4.7: Max Pooling of a 4x4 matrix to a 2x2 matrix

Also note that this max pooling filter has a stride of two, causing the matrix to be downsized. When introducing padding the output height and width is equivalent to $H = H_o/\text{stride}$, $W = W_o/\text{stride}$. Let's consider a more complex network called a U-Net model for tumor detection and segmentation [31], [32]. This will allow us to see the overall picture of fully convolutional neural networks.

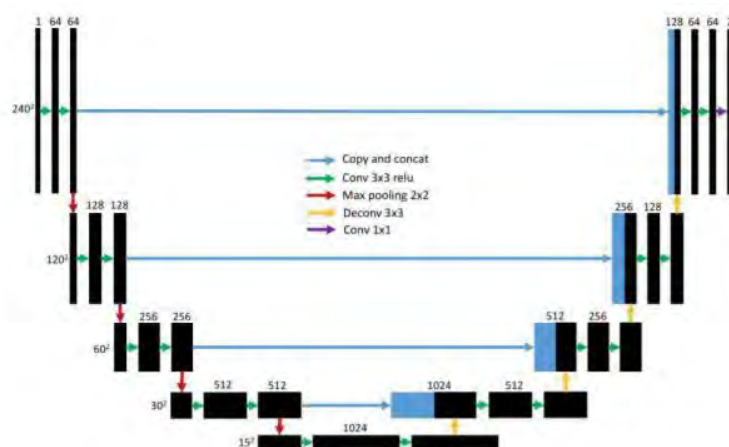


Figure 4.8: U-Net Model for Image Segmentation [32]

This fully convolutional neural network consists of down sampling and upsampling. With the input and output image being of the same size. The down-sampling path has 5 convolutional portions. Every block has two convolutional layers with a filter size of 3×3 , and a stride of 1. The activation function is a rectifier activation. For each block of the down-sampling, the number of convolutional layers increases by a factor of 2. After each convolutional block, a down-sampling of max pooling is used with a stride 2×2 is applied. The decrease in size is divisible by 2 until reach the last block of 15×15 . In the up-sampling path, to increase each block, a deconvolutional layer with a filter size of 3×3 and stride of 2×2 is applied. It doubles the size of feature maps. In every up-sampling block, two convolutional layers are also implemented. The number of filter maps is reduced by a factor of 2. However, feature maps of the down-sample path are concatenated with the up-sample path before being convoluted. The loss function is a ‘Soft’ Dice loss function introduce in [32]. Like our previous example of a simple neural network, the U-Net model will adjust itself through backpropagation and gradient descent. With a general concept of machine learning we can begin to dive into the type of architecture we implemented on our current numerical simulation and why.

4.5 REASONING BEHIND IMPLEMENTING A DEEP NEURAL NETWORK

To analyze turbulence effects, laser propagation is simulated through the phase screen approach and modeled by Kolmogorov [19] and non-Kolmogorov statistical theories [18]. Our main objective is to characterize atmospheric turbulence effects through statistical analysis of the phase fluctuations at the observer’s plane. To do so we need to calculate the phase screen with the arctangent function, and the values are restricted to an interval of $(-\pi, \pi)$. Overcoming this problem requires 2D phase unwrapping to restore the true phase values, assuming that the unwrapped phase map is a continuous surface. To unwrap the phase screens a 2D Goldstein

unwrapping algorithm is implemented that is compiled by *Bruce Spottiswoode* and edited by *Carey Smith* [20]. However, there are some complication when using the current unwrapping algorithm.

As we mentioned before when calculating the phase variance and phase structure function, we saw discrepancies at Rytov values greater than 0.5. This may be due to the grid size not being sufficiently large, such that sample frequency spacings can be low enough to accurately represent the outer scale effects. It could also be due to the Goldstein algorithm to the current numerical simulation breaking down when we begin to implement deep atmospheric turbulence. When we plot the unwrapped phase screens as shown in figure(14c) we see fractures and discontinuities in the image. To accurately develop a numerical simulation of laser propagation we need to overcome this problem.

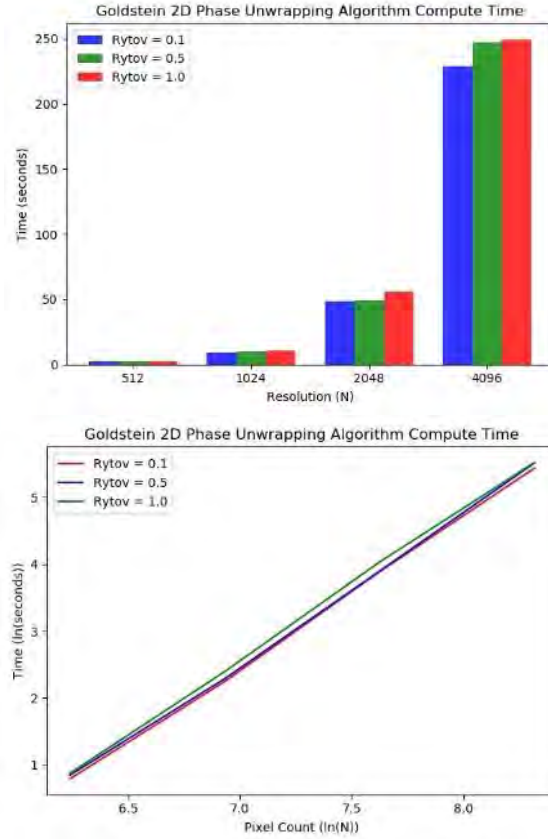


Figure 4.9: Goldstein Algorithm Compute Time relative to spatial grid size

The algorithm does not scale well for large spatial grids (Figure 1). Another reason to implement neural networks is to accelerate the current simulation. It would be beneficial to reduce the computational time between each simulation. This will allow us to compute statistical parameters at larger grid sizes at a reasonable time. These statistical parameters need to be as accurate as possible to better represent the effects of atmospheric turbulence. When plotted on a log-log scale we can see that the algorithm compute time scales exponentially with the grid size. To accelerate our numerical simulations of wave propagation, we train a deep neural network to approximate the Goldstein 2D phase unwrapping.

4.6 CONDITIONAL GENERATIVE ADVERSARIAL NETWORK

Convolutional neural networks have been the standard to solve a variety of classification and regression problems [32]–[35]. However, convolutional neural networks can't be applied to a problem domain naively, because the specified loss may produce undeniable results. For example, it is common that implementing an L2 norm loss to a convolutional model can produce a blurred image [36], as a local average is often a good approximation of an image. Implement a loss function that produces desired images is complicated and problem-specific. Generative adversarial networks (GAN) [37] are a deep neural network architecture that effectively learns a loss function simultaneously, alongside a useful mapping function. A standard (or vanilla) GAN comprises two components, the generator, and the discriminator, and pits them against each other.

For this study, we implement cGAN [38] to learn phase unwrapping, which is similar to a vanilla GAN but applies a conditioning variable. This conditioning variable is seen by both generator and discriminator which establish some arbitrary condition for a generation. We can state the objective of a using a min-max value function.

$$\mathcal{L}_{cGAN}(G, D) = \min_G \max_D \left(\mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,z} \left[\log \left(1 - D(x, G(x, z)) \right) \right] \right) \quad (4.10)$$

By minimizing this loss function, we train the discriminator, $D(x, y)$, to maximize the expectation of target data distribution, y , and minimize expectation of the generated model data distribution, $G(x, z)$. At the same time, we need to train the generator to maximize the probability that the discriminator is fooled into choosing the generated data or fake image. In this

instance, the conditioning variable is the input image, x , which will be seen by both the generator and the discriminator. In our problem, the wrapped phase screen will be the input image, and the unwrapped phase screens will be the target image (Figure 4.10).

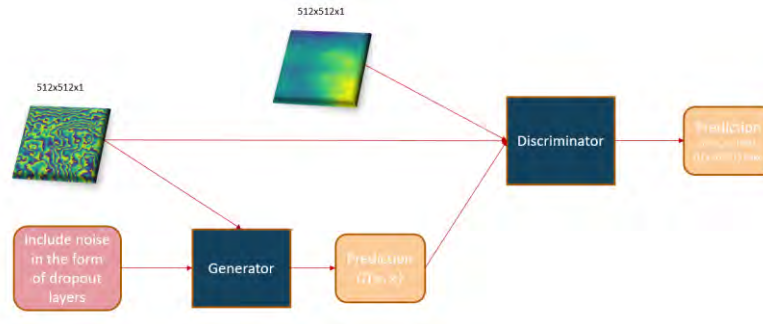


Figure 4.10: Conditional Generative Adversarial Network for Phase Unwrapping

In a cGAN, it is common to input a noise variable, z , to the generator to impose stochasticity on the outputs. Past cGan models try to solve this problem by implementing a Gaussian noise, z , to the generator in addition to, x . The strategy is ineffective, as the generator learns to ignore the random noise. The cGan model that implemented in this work for phase unwrapping applies noise in the form of dropout layers in several layers of the generator [38].

Before going in depth into each architecture involved in a cGAN, it is important to understand why we need to preprocess the dataset before training. Before training a model, we need to normalize the data set before feeding it into the neural network. Depending on the activation function we need to normalize the data set to be between $[0,1]$ or in our case $[-1,1]$. This is because the activation functions only work within a small range. As an example, our generator final activation function is a $\tanh(x)$. Any information that is not within the range of $[-1,1]$ will be lost and not included in our network. That is why normalizing and batch

normalization are important. This makes sure that values do not stray far from the range of the activation function. Thus, the gradient descent can reduce the oscillations when approaching the minimum point and converge faster. It also acts as a form of regulation and prevents overfitting.

The task of the generator is to be able to take the input image and perform the requires transforms to produce the target image. In our case, we would want the generator to be able to perform phase unwrapping. It would input the unwrap phase and be able to transform the 2D matrix and be able to produce an unwrapped phase. Most generators consist of the encoder (down-sampling) and decoder (up-sampling) layers. However, the generator we are implementing resembles a U-Net model and implements skip-connections.

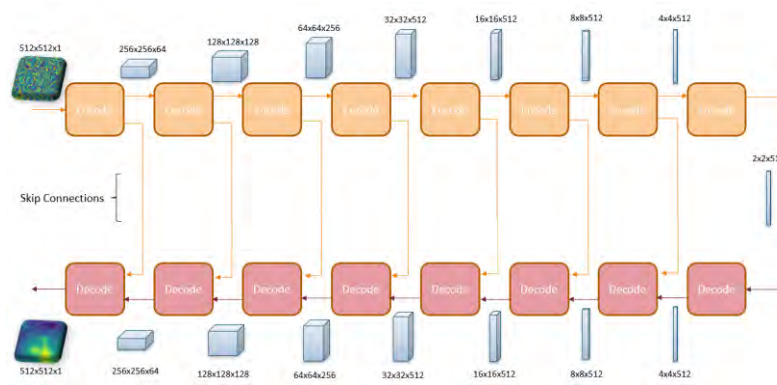


Figure 4.11: Generator Architecture

The generator takes the input image and reduces it with a series of encoders into a much smaller representation. The idea is to not only reduce the number of parameters in the networks but have a higher representation of data in the final encoder layer. Then the

decoder will do the exact opposite and propagate this information to higher resolution layers. The skip connections have shown to help recover spatial resolution that may have been lost within the network [39].

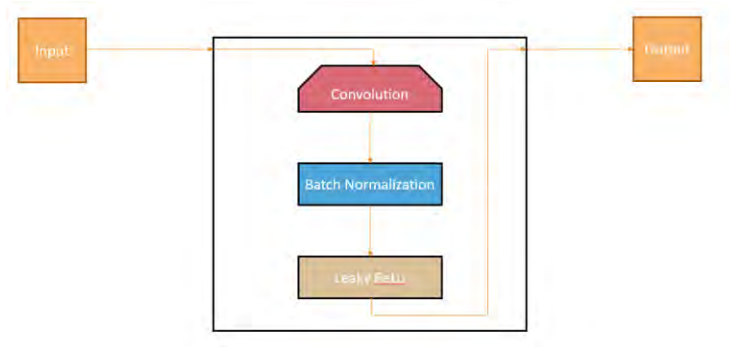


Figure 4.12: Encoder Structure

The encoders consist of convolution, batch normalization, and an activation function. The convolution consists of a kernel size of 4x4 and a stride of 2. This decreases the input image by a factor of 2. We then implement batch normalization before the activation function. The encoder activation function is a Leaky Rectified Linear Unit (Leaky ReLU). Like a ReLU, a Leaky ReLU does not saturate for values less than zero. This allows for more information to be included in the network for gradient descent. The output channel after each encoder should increase by a factor of 2 before reaching 512. After reaching a channel of 512 it will remain at that value for the remainder of the encoders.

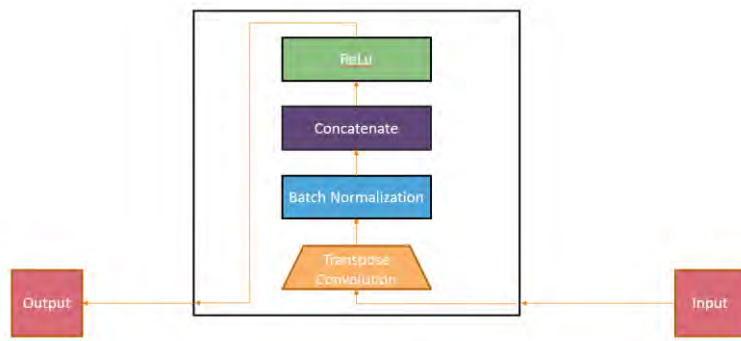


Figure 4.13: Decoder Structure

The decoder consists of transpose convolution with a kernel size of 4x4 and a stride of 2. It increases the input image by a factor of 2. Then the image will be normalized before being concatenated with an encoder image of similar height and width. The activation function for a decoder is a ReLu. The first four decoder blocks will have a filter channel of 512 before reducing by a factor of 2. The images are a simplification of the generator model. Other things that are not mentioned in the image is that the last activation function of our generator will be a $\tanh(x)$. We need to also note that the first three decoder blocks include the drop out layers. This is the architecture of the generator which will try to fool the discriminator and make predictions of phase unwrapping.

The discriminator will take two images as an input. It will take the input image which is out conditioning variable and an unknown image. This unknown image is either the target image or the predicted image that is produced by the generator. The generator will take this two pairs of images and concatenate them together and determine if it was produced by the generator or not.

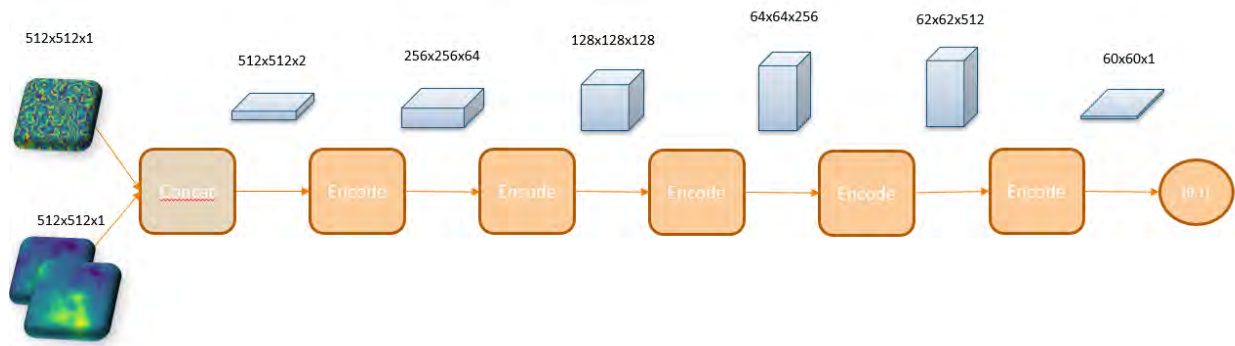


Figure 4.14: Discriminator Architecture

The architecture of the discriminator built up of only encoders. The encoders are similar to those of the generator which have convolution, batch normalization, and a Leaky ReLU activation function. However, the last two encoder portions have a stride of 1. The output should be a 60x60x1 image where each pixel value ranges from 0 to 1. The 0 denotes that it fake and 1 represents that is the target image. Each pixel value corresponds to how believable the image is to the target image. This type of architecture is called a PatchGAN [38].

To train this network, there are two steps, training the discriminator and training the generator. To train the discriminator, the generator will generate a fake image. The discriminator looks at the input/target pair and we will specify to the model that this is a real image which will be denoted by 1. Then the discriminator will look at the input/fake pair and we will specify to the model that this is the fake pair which is denoted by 0. The weights of the discriminator are then adjusted based on the classification error of the input/output pair and the input/fake pair (Figure 4.15: Left).

output an HDF5 file containing the unwrap and wrapped matrices. These values contain negative and positive float values. We would then call on executable file through python where we converted the matrices into images. This was only done as a proof of concept and make sure that this will be a feasible problem.

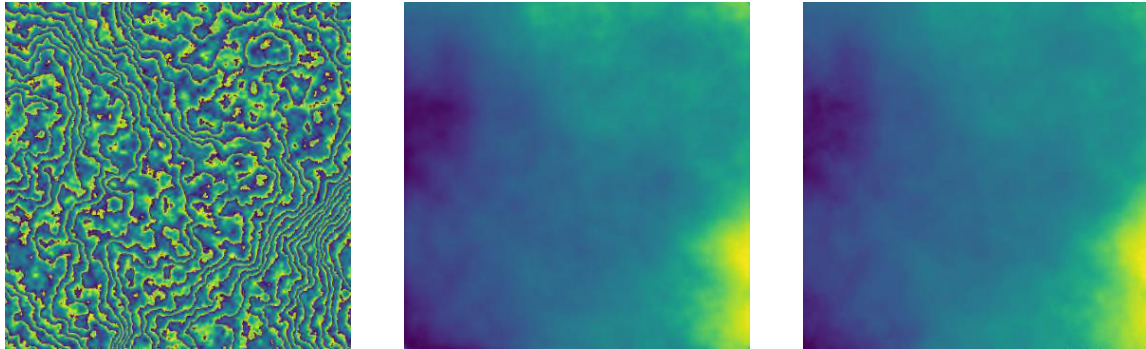


Figure 4.16: Training Image Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image

The dataset generated for this study comprised of 4000 training images and 1000 validation images. The grid size of these images is 512x512. The cGAN model was trained for 3.5 hours, which yielded 45 epochs of training. A qualitative comparison of the resultant model output is shown in the figures below. These images are taken at random from the validation partition of the dataset and were therefore never used to train the model. Thus, these images constitute a qualitative estimate of the model's generalization performance.

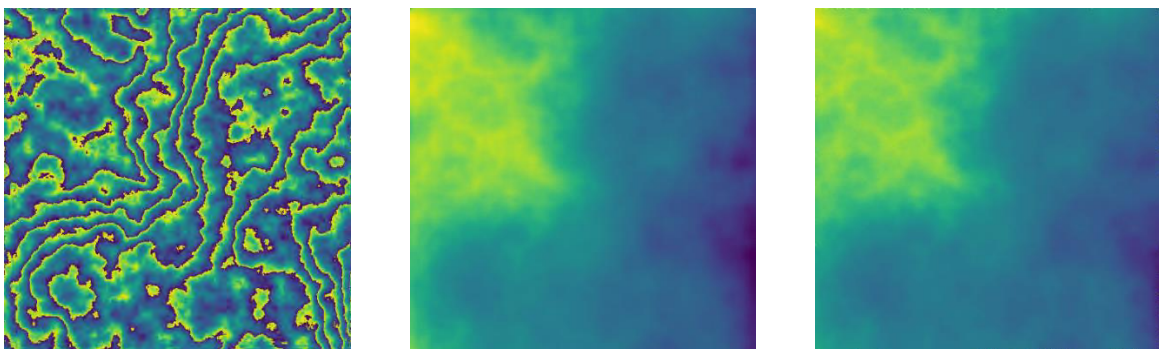


Figure 4.17: Validation Image Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image

The obtained results suggest that conditional generative adversarial networks are a promising approach to learn phase unwrapping. It may be possible to make predictions for larger image sizes but would need further investigation to verify. We would need to test this current neural network onto the actual float matrices of the wrap and unwrapped phase screens. The objective is to integrate our model to our current numerical simulation of wave propagation and calculate statistical parameters on the unwrapped phase screens. So instead of working with images, we would need to test our model on the actual data set.

4.8 PHASE UNWRAPPING WITH MATRICES

Instead of converting the matrices into images we want to train out a model with the float matrices. But before feeding the data into the neural network we need to determine the sample data set mean and standard deviation. We need to normalize the data set before feeding it into our model.

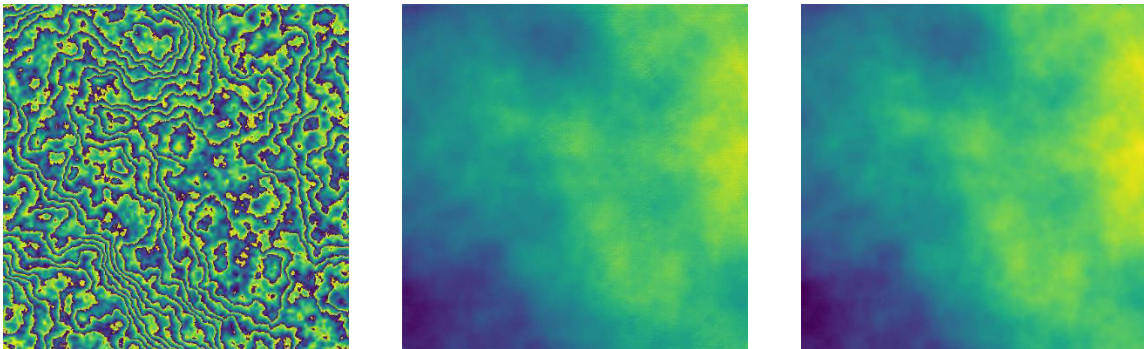


Figure 4.18: Validation Matrix Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix

The dataset generated for this study comprised 4800 training matrices and 1200 validation matrices. The grid size of these images is 512x512. The cGAN model was trained for approximately 40 hours, which yielded 250 epochs of training. A comparison of the resultant model prediction and the target image are shown in Figure (4.18). These images are taken at random from the training partition of the dataset like before. It seems that the predicted image is approximately like the target image, being able to show the target image pattern. But looking at the predicted matrix closely, we see that there is a checkerboard pattern in the image.

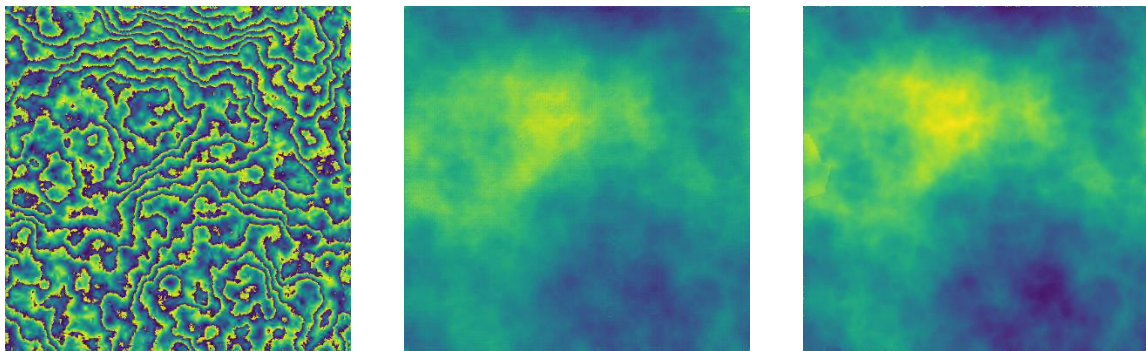


Figure 4.19: Validation Matrix Set: (Left) Input Image, (Middle) Predicted Image, (Right) Target Image

Even though the predicted image almost resembles the target image for the validation set, the checkerboard artifacts become more prominent than in the training set. This is due to the transpose convolution operation that we use to go from a low resolution to a higher one.

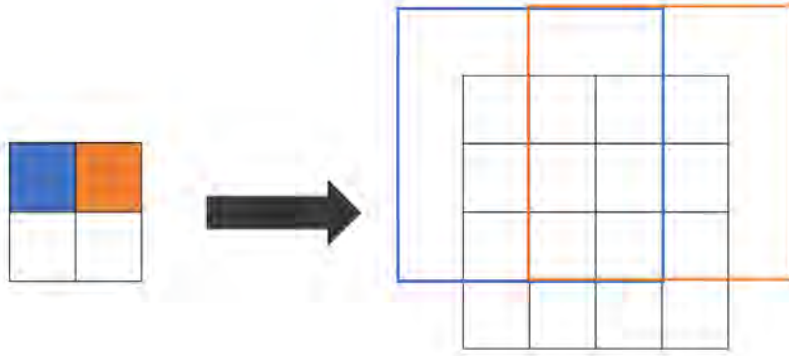


Figure 4.20: 4x4 Transpose convolution with a stride of 2

Transpose convolution work by swapping the forward and backward passes of a convolution. This can take a small 2x2 image and increase the resolution to a 4x4 image. The problem with transpose convolution is it can have uneven overlap between pixels. This uneven overlap occurs more when the kernel size is not divisible by the stride. The uneven overlap can be compounded when transposed convolutional layers are stacked. In practice to avoid these artifacts a convolution of stride 1 is used on the last layers, but in other models, it isn't enough [40]. To improve on the image resolution [41], instead of up-sampling with transpose convolution we resize the image followed by a convolutional layer. We will be replacing our transpose convolutional layers with nearest neighbor interpolation. Then follow it with a convolutional layer with a kernel size of [2,2] and a stride of 1.

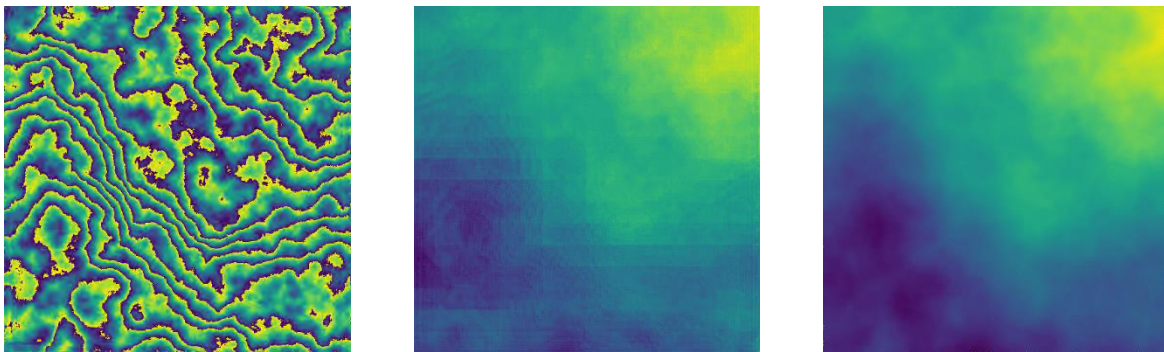


Figure 4.21: Modified cGAN Training Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix

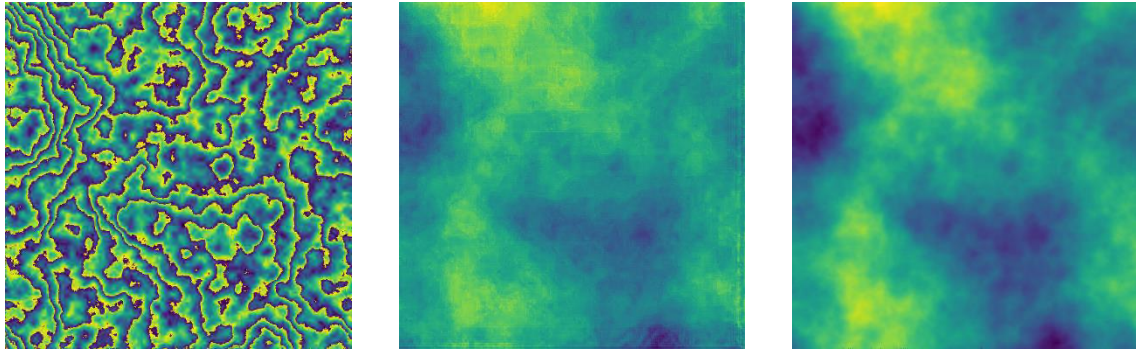


Figure 4.22: Modified cGAN Validation Set: (Left) Input Matrix, (Middle) Predicted Matrix, (Right) Target Matrix

However, when we modify our cGAN to include the resize method using nearest neighbor interpolation and a convolutional layer we see more checkerboard artifacts. This may be due to the kernel size and stride chosen for the convolutional layer. There may be pixel overlap and aren't really negating the effects caused by transpose convolution in the backwards pass. More work needs to be conducted to remove any artifacts in the matrix, so we can make actual prediction with float matrices. Being able to conduct phase unwrapping using neural networks will not only accelerate our current simulation but allow us to conduct larger grid sizes that isn't currently feasible. We can still conduct phase unwrapping with the current algorithm with some restrictions. This still allows us to implement CFD to model laser propagation in atmospheric turbulence.

Chapter 5: Governing Equations of Fluid Mechanics

To begin implantation of CFD on our current simulation this section will go over the basics of fluid mechanics, to plant the base of our overall research. We will consider a fluid, whether it be liquid or gas, to be moving in the domain Ω included in three-dimensional space, $\mathbb{R}^3$. We will describe the fluid flow with the Navier Stokes equation which is derived by the mass conservation and fluid motion equation. In this study, we will only be looking at air and the effects of atmospheric turbulence on laser propagation. We will consider air to be an incompressible Newtonian fluid.

$$\frac{\partial u}{\partial t} + \bar{v} \cdot \nabla \bar{v} = -\nabla p + \nabla \cdot (2\nu D\bar{v}) + f \quad (5.1)$$

$$\nabla \cdot \bar{v} = 0 \quad (5.2)$$

In this scenario $\bar{v}$ is the velocity of the fluid, p is pressure and $D\bar{v} = (1/2)(\nabla \bar{v} + \nabla \bar{v}^T)$ is the deformation tensor. Eq.1 is the momentum equation, derived from Newton's law while, equation (5.2), is determined from the mass conservation equation of incompressible fluids. We will go over the derivation of both equation and how coupled yield the Navier Stokes equation.

Before deriving the governing equations, we need to define the Lagrange and Euler coordinate systems. However, in fluid mechanics, we will be dealing mainly with the Euler description. Euler coordinates are denoted by $\mathbf{x}$, which represents the spatial

coordinate. A spatial coordinate specifies the location of a point in space. Lagrangian coordinates also known as material coordinates, are denoted by $\mathbf{X}$. The material coordinate specifies the materials point or location. In Eulerian coordinates, the nodes of the mesh are fixed in $\mathbf{x}$, and the material moves through the mesh. In a Lagrangian coordinate the mesh is fixed in $\mathbf{X}$, and the mesh moves with the material.

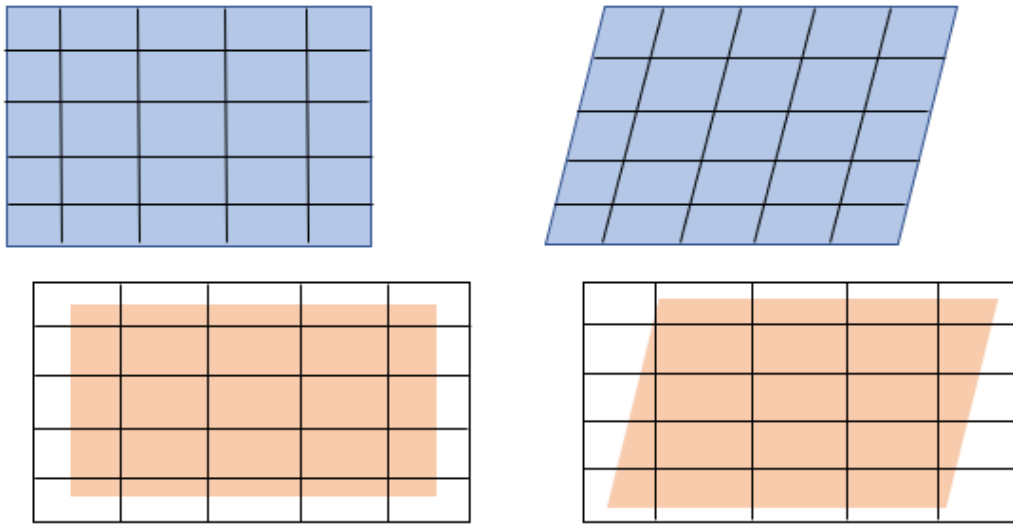


Figure 5.1: Two-Dimensional shearing of a block: Top (Lagrangian) and Bottom (Eularian)

The deformation and/or motion of a body is described by the function $\phi(\mathbf{X}, t)$. This function gives the spatial position of the material coordinate point as a function of time.

$$\mathbf{x} = \phi(\mathbf{X}, t) \quad (5.3)$$

This is also the map between the initial and current configuration. The displacement u of a material point is the difference between the current position and the original position.

$$u(\mathbf{X}, t) = \phi(\mathbf{X}, t) - \mathbf{X} \quad (5.4)$$

The velocity of the material point can be easily determined by taking the time derivative of the motion or deformation of the body.

$$v(\mathbf{X}, t) = \frac{\partial \phi(\mathbf{X}, t)}{\partial t} \quad (5.5)$$

All of these derivations were done with the material coordinate being fixed meaning that this is in the Lagrangian frame of reference. We can also express the material coordinates in terms of the spatial coordinates by inverting the mapping function.

$$\mathbf{X} = \phi^{-1}(\mathbf{x}, t) \quad (5.6)$$

This will allow us to describe the velocity as a function of the Eulerian coordinates.

$$\bar{v}(\mathbf{x}, t) = v(\phi^{-1}(\mathbf{x}, t), t) \quad (5.7)$$

We have indicated Eulerian frame of reference by placing a bar over the velocity symbol. The difference in the Eulerian and Lagrangian meshes is seen in the behavior of the nodes. If the mesh is Eulerian, the Eulerian coordinates of nodes, $\mathbf{x}$, are fixed. If the mesh is Lagrangian, the material points, $\mathbf{X}$, are time invariant.

5.1 MASS CONSERVATION EQUATION

Let us consider a fluid particle that is located at $\mathbf{x} = \mathbf{x}(t)$ and moves to $\mathbf{x} = \mathbf{x}(t + \delta t)$. This particle is moving along a trajectory with a given velocity $\bar{\mathbf{v}}(\mathbf{x}, t)$.

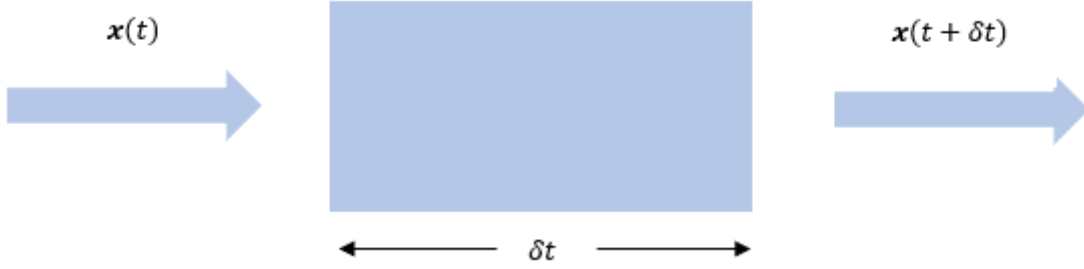


Figure 5.2: Fluid particle moving across a controlled volume

The mass conservation equation states that the mass of a particle remains constant along a given trajectory.

$$\frac{D(dm)}{Dt} = 0 \quad (5.8)$$

Recall that mass $dm(\mathbf{x}, t)$ is equivalent to $\rho(\mathbf{x}, t)d\mathbf{v}(\mathbf{x}, t)$, where $d\mathbf{v}$ is the local volume.

All we need to do is compute the total derivative of mass.

$$\frac{D(dm)}{Dt} = \left(\frac{\partial \rho}{\partial t} + \bar{\mathbf{v}} \cdot \nabla \bar{\mathbf{v}} \right) d\mathbf{v} + \rho(\nabla \bar{\mathbf{v}}) d\mathbf{v} \quad (5.9)$$

We can then combine equation (5.8) and (5.9) to obtain

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\bar{v} \rho) = 0. \quad (5.10)$$

However, for incompressible flow, the mass density remains constant during in motion. This reduced our conservation of mass to be

$$\nabla \cdot \bar{v} = 0. \quad (5.11)$$

5.2 THE EQUATION OF MOTION

To solve conservation of momentum by applying Newton law $\sum F = ma$, to a given fluid particle. We assume that the external forces exerted on a point in the particle $\mathbf{x}$, at time t by the fluid can be described by a density function $f_{ext}(\mathbf{x}, t)$. The sum of applied forces is equal to

$$\rho a(\mathbf{x}, t) d\mathbf{v} = (\nabla \cdot \sigma + f_{ext}) d\mathbf{v} \quad (5.12)$$

We aim to determine the acceleration of the particle denoted by a , which is equal to

$$a(\mathbf{x}, t) = \frac{D\bar{v}}{Dt}(\mathbf{x}, t). \quad (5.13)$$

The by computing the total derivative of velocity we obtain

$$\frac{D\bar{v}}{Dt}(\mathbf{x}, t) = \frac{\partial \bar{v}}{\partial t} + (\bar{v} \cdot \nabla) \bar{v}. \quad (5.14)$$

Applying Newtown's law, equation (5.12), to our particle yields

$$\rho \left(\frac{\partial \bar{v}}{\partial t} + (\bar{v} \cdot \nabla) \bar{v} \right) dv = (\nabla \cdot \sigma + f_{ext}) dv \quad (5.15)$$

Since the local volume dv , is not zero we can divide each side and determine the momentum equation

$$\rho \left(\frac{\partial \bar{v}}{\partial t} + (\bar{v} \cdot \nabla) \bar{v} \right) = \nabla \cdot \sigma + f_{ext}. \quad (5.16)$$

The stress tensor σ , is often determined by experiments and is often convenient to split the stress tensor as

$$\sigma = \mathbb{D} - p\mathbb{I}. \quad (5.17)$$

The expression is composed of the dynamic pressure p and $\mathbb{D}$ the deviatoric part of σ . Furthermore, since the stress tensor is symmetric, and through experimentation [42], the deviatoric component is $\mathbb{D} = 2\mu Du$. Consequently, this will alternate the expression of the momentum equation to be

$$\rho \left(\frac{\partial \bar{v}}{\partial t} + (\bar{v} \cdot \nabla) \bar{v} \right) = \nabla \cdot (2\mu Du - p\mathbb{I}) + f_{ext}. \quad (5.18)$$

For incompressible flow, we assume that density is constant, $\rho = \rho_o$. When we divide the momentum equation by density, we will denote the ratio p/ρ by p . This ratio is better known as the density of pressure per unit of mass and volume. We will also denote the ratio of f_{ext}/ρ as f , which is still called the external force. We can then take the momentum and mass conservation equation while noting that $\nabla \cdot (p\mathbb{I}) = \nabla p$, to derive the Navier Stokes equation.

$$\frac{\partial u}{\partial t} + \bar{v} \cdot \nabla \bar{v} = -\nabla p + \nabla \cdot (2\nu D \bar{v}) + f \quad (5.19)$$

This is the governing equation of fluid motion which we will use to conduct our CFD simulations.

Chapter 6: Computational Fluid Dynamics for Laser Propagation

Computational fluid dynamic (CFD) is the numerical simulation of fluid motion. The approach is geared toward modeling turbulence at a degree of accuracy to determine the coupled fluid dynamic behaviors. There are various numerical models that have been developed over the years to describe a wide variety of fluid dynamic phenomenon. They range from least to best in accuracy when simulating turbulent flow, but computational requirements can restraint researchers from using the most sophisticated models.

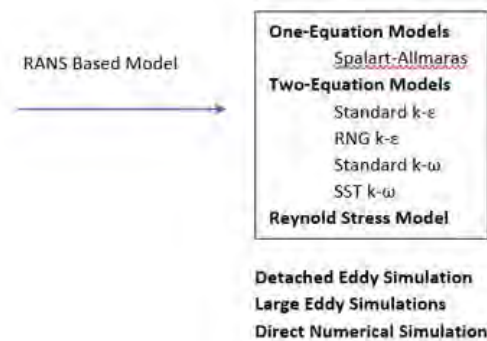


Figure 6.1: Various Models of CFD

The most accurate approach for simulating turbulent flows is the Direct Numerical Simulation (DNS) model in which the full Navier-Stokes equation is numerically solved. There is no modeling involved as it is solved directly by a finite mesh to capture all the scales that are present in a turbulent flow. Therefore, DNS is computationally prohibitive and is not the most commonly used method to simulate turbulent flow. In the other hand for modest computing requirement, the Reynolds-averaged Navier-Stokes (RANS) approach solves the ensemble-average of the Navier-Stokes equations. This approach requires extensive modeling to model the effect of all the scales of instantaneous turbulent motion. It is also the least accurate approach when trying to model turbulent flow.

An alternative approach would be Large-Eddy Simulations (LES) which directly solves the large-scale motions in turbulent flow while the small-scale motions are modeled. This results in a significant reduction of computational cost and does not adopt the conventional ensemble averaging that is used in the RANS approach. LES is more accurate than the RANS approach by directly solving the large eddies which are responsible for the momentum transfer and contain the most energy in a turbulent flow. The smaller eddies, or Sub-Grid Scale (SGS), are assumed to be more isotropic and therefore easier to model. Therefore, LES is a promising method of accurately modeling turbulent flow with reasonable computing power.

6.1 FINITE VOLUME METHOD

To evaluate the partial differential equations in fluid motion we need to implement the finite volume method. Let's consider the governing equations for a two-dimensional, steady flow

$$\frac{\partial}{\partial x}(\rho uu) + \frac{\partial}{\partial y}(\rho vu) = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) + S_u \quad (6.1)$$

$$\frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho vv) = \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) + S_v \quad (6.2)$$

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \quad (6.3)$$

where $S_u = -\partial p / \partial x$ and $S_v = -\partial p / \partial y$ and they are the pressure gradient terms which is what we initially try to solve. There are complications with solving the governing equations because the convective terms contain non-linear quantities and all three equations are coupled with each other. We would initially solve for the pressure gradient by implementing a checker-board pressure field grid, but this results in nonrealistic representation of the flow. To solve this problem we need to implement a staggered grid for the velocity components [43].

The idea is to solve for scalar variables at ordinary nodal points like before but to calculate velocity components on staggered grids centered around the cell faces. H. K. Versteeg and W. Malalasekera can determine the coefficients for pressure-velocity coupling to be [44]

$$a_{i,j}u_{i,j} = \sum a_{nb}u_{nb} + (p_{I-1,J}p_{I,J})A_{i,j} + b_{i,j} \quad (6.4)$$

$$a_{I,j}v_{I,j} = \sum a_{nb}v_{nb} + (p_{I,J-1}p_{I,J})A_{I,j} + b_{I,j} \quad (6.5)$$

where $b_{i,j} = \bar{S}\Delta V_u$ the momentum source term, and $A_{i,j}$ is the cell face area. Coefficients $a_{i,j}$ and a_{nb} are calculated through any differencing methods suitable for convection-diffusion problem. The unbroken grid lines are numbered by capital letters and the dashed lines are noted by lowercase letters. Scalar nodes are located at the intersection of two grid lines and identified by two capital letters. Velocity components are located at the intersection of a line defining a cell boundary and grid line which is why they noted by both lowercase and capital notation. Then we can continue to determine coefficients for the convective flux per unit mass F and diffusive conductance D at u and v control volume faces [44].

u-control volume

$$F_w = (\rho u)_w = \frac{F_{i,J} + F_{i-1,J}}{2}$$

$$F_e = (\rho u)_e = \frac{F_{i+1,J} + F_{i,J}}{2}$$

$$F_s = (\rho v)_s = \frac{F_{l,j} + F_{l-1,j}}{2}$$

$$F_n = (\rho v)_n = \frac{F_{l,j+1} + F_{l-1,j+1}}{2}$$

$$D_w = \frac{\Gamma_{l-1,J}}{x_i - x_{i-1}}$$

$$D_e = \frac{\Gamma_{l,J}}{x_{i+1} - x_i}$$

$$D_s = \frac{\Gamma_{l-1,J} + \Gamma_{l,J} + \Gamma_{l-1,J-1} + \Gamma_{l,J-1}}{4(y_j - y_{j-1})}$$

$$D_n = \frac{\Gamma_{l-1,J+1} + \Gamma_{l,J+1} + \Gamma_{l-1,J} + \Gamma_{l,J}}{4(y_{j+1} - y_j)}$$

v-control volume

$$F_w = (\rho u)_w = \frac{F_{i,J} + F_{i,J-1}}{2} \quad (6.6)$$

$$F_e = (\rho u)_e = \frac{F_{i+1,J} + F_{i+1,J-1}}{2} \quad (6.7)$$

$$F_s = (\rho v)_s = \frac{F_{l,j-1} + F_{l,j}}{2} \quad (6.8)$$

$$F_n = (\rho v)_n = \frac{F_{l,j} + F_{l,j+1}}{2} \quad (6.9)$$

$$D_w = \frac{\Gamma_{l-1,J-1} + \Gamma_{l,J-1} + \Gamma_{l-1,J} + \Gamma_{l,J}}{4(x_l - x_{l-1})} \quad (6.10)$$

$$D_e = \frac{\Gamma_{l,J-1} + \Gamma_{l+1,J-1} + \Gamma_{l,J} + \Gamma_{l+1,J}}{4(x_{l+1} - x_l)} \quad (6.11)$$

$$D_s = \frac{\Gamma_{l,J-1}}{y_{jk} - y_{j-1}} \quad (6.12)$$

$$D_n = \frac{\Gamma_{l,J}}{y_{j+1} - y_j} \quad (6.13)$$

To solve for the velocity and pressure for a given control volume we need to implement an algorithm presented by Patankar and Spalding [45]. Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) begins by guessing the pressure field. The initial guess will be implemented onto the discretized momentum equation (6.4) and equation (6.5) so solve for the velocity components noted by (*).

$$a_{i,j}u_{i,j}^* = \sum a_{nb}u_{nb}^* + (p_{I-1,J}^* - p_{I,J}^*)A_{i,j} + b_{i,j} \quad (6.14)$$

$$a_{I,j}v_{I,j}^* = \sum a_{nb}v_{nb}^* + (p_{I,J-1}^* - p_{I,J}^*)A_{I,j} + b_{I,j} \quad (6.15)$$

Now we have to define a correction parameter noted by (') which is the difference between the correct parameter and the guessed parameter.

$$p' = p - p^* \quad u' = u - u^* \quad v' = v - v^* \quad (6.16)$$

Subtracting the momentum equation with the correct pressure field by the discretized equation with the guessed pressure field yields the correction values. Then the main approximation for the SIMPLE algorithm is implemented by dropping the summation value in the discretized equation for the correction velocity fields,

$$u'_{i,j} = (p'_{I-1,J} - p'_{I,J})d_{i,j} \quad (6.17)$$

$$v'_{I,j} = (p'_{I,J-1} - p'_{I,J})d_{I,j} \quad (6.18)$$

where $d_{i,j} = \frac{A_{i,j}}{a_{i,j}}$ and $d_{I,j} = \frac{A_{I,j}}{a_{I,j}}$. Now we can use the correction values to solve for equation.

(6.17) and 5.(18).

$$u_{i,j} = u_{i,j}^* + d_{i,j}(p'_{I-1,J} - p'_{I,J}) \quad (6.19)$$

$$v_{I,j} = v_{I,j}^* + d_{I,j}(p'_{I,J-1} - p'_{I,J}) \quad (6.20)$$

There is a similar expression exist for $u_{i+1,j}$ and $v_{i,j+1}$ that contains the similar form as equation (6.14) and equation (6.15). Now we need to consider the constraints in the velocity field that satisfy the continuity equation from equation (6.3).

$$[(\rho u A)_{i+1,j} - (\rho u A)_{i,j}] + [(\rho v A)_{i,j+1} - (\rho v A)_{i,j}] = 0 \quad (6.21)$$

Substitution of the corrected velocities onto the discretized continuity equation gives

$$a_{i,j} p'_{i,j} = a_{i+1,j} p'_{i+1,j} + a_{i-1,j} p'_{i-1,j} + a_{i,j+1} p'_{i,j+1} + a_{i,j-1} p'_{i,j-1} + b'_{i,j} \quad (6.22)$$

where the coefficients are given below.

Table 6.1: Expanded coefficients for discretized continuity equation

$a_{i,j}$	$a_{i+1,j}$	$a_{i-1,j}$	$a_{i,j+1}$		$a_{i,j-1}$	$b'_{i,j}$
$a_{i+1,j}$ + $a_{i-1,j}$ + $a_{i,j+1}$ + $a_{i,j-1}$	$(\rho dA)_{i+1,j}$	$(\rho dA)_{i,j}$	$(\rho dA)_{i,j+1}$		$(\rho u A)_{i,j}$	$(\rho u^* A)_{i,j}$ - $(\rho u^* A)_{i+1,j}$ + $(\rho v^* A)_{i,j}$ - $(\rho v^* A)_{i,j+1}$

By solving the equation, we can determine the correction field p' at all points and use it to determine the correct pressure field p . The pressure correction equation diverges unless some under-relaxation term is used in the iterative process [44].

$$p^{new} = p^* + \alpha_p p' \quad (6.23)$$

The pressure under-relaxation factor is α_p and ranges between zero and one. Taking $0 < \alpha_p < 1$ allows us to add to the guessed field p^* by a fraction of the correction field p' that will allow us to move the iterative process forward. The velocity components are also under-relaxed.

$$u^{new} = \alpha_u u + (1 - \alpha_u)u^{n-1} \quad (6.24)$$

$$v^{new} = \alpha_v v + (1 - \alpha_v)v^{n-1} \quad (6.25)$$

The velocity under-relaxation factor is α_u and α_v for u- and v-velocity. The corrected velocity components without relaxation are u and v . The values obtained in the previous iteration are represented by u^{n-1} and v^{n-1} . Optimum values for the under-relaxation parameters are dependent on the flow and must be determined for each individual problem. The SIMPLE algorithm provides a method for calculating pressure and velocities. The method is iterative, and when other scalars are coupled to the momentum equations the calculation needs to be done sequentially [44].

6.2 LARGE EDDY SIMULATION

In LES simulations the large-scale motions are represented directly, and the smaller scales are modeled through sub-grid scale modeling. The velocity component u is split between the filtered component $\bar{u}$ and a residual subgrid-scale component $\acute{u} = u - \bar{u}$ [46]. Then we derive the governing equations employed in LES from a filtered Navier Stokes equation to compute the evolution of the filtered velocity field. This filtered Navier Stokes equation has the same form as the unfiltered equation except for a residual stress tensor arising from the residual motions. This

residual stress tensor must be modeled by the various sub-grid scale models. Then the filtered equation can then be solved numerically for the filtered velocity.

The filtering operator is defined as

$$\bar{u}(x, t) = \int_D G_\Delta(r, x) u(x - r, t) dr \quad (6.26)$$

where D is the fluid domain and G_Δ is the filter function that determines the scale of the resolved eddies. The filter is called uniform if it does not depend on x , and isotropic if it depends only on the displacement vector r . There are many filter functions with varying properties and is dependent on what effects are needed from the filtered continuity equation. In Ansys Fluent, the filter function filters out the eddies whose scales are smaller than the filter width or grid spacing [47]. This simplifies the integration over the flow domain to be

$$\bar{u} = \frac{1}{V} \int_V u(x - r, t) dr, \quad r \in V \quad (6.27)$$

where V is the volume of the computational cell. This will help us resolve the filtered Navier Stokes equation to be

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}}_i) = 0 \quad (6.28)$$

$$\frac{\partial (\rho \bar{\mathbf{u}}_i)}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}}_i \bar{\mathbf{u}}_j) = -\nabla \bar{p} + \nabla \sigma_{ij} + \nabla \tau_{ij} \quad (6.29)$$

where $\sigma_{ij} = [\mu(\nabla \bar{u}_i + \nabla \bar{u}_j)] - \frac{2}{3}\mu \nabla \bar{u}_l \delta_{ij}$ is the stress tensor due to molecular viscosity and $\tau_{ij} = \rho \overline{u_i u_j} - \rho \bar{u}_i \bar{u}_j$ is the subgrid-scale stress. The next step will be deciding which sub-grid scale model will be used to derive the residual stress.

6.3 WALL-ADAPTING LOCAL EDDY-VISCOSITY (WALE) MODEL

In LES scales smaller than the smaller than the grid size are not directly resolved but are represented by the subgrid-scale stress. Most sub-grid scale modeling are based on the Boussinesq's eddy viscosity assumption [48] to model the subgrid scale tensor.

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_t \bar{S}_{ij} \quad (6.30)$$

μ_t is the subgrid-scale turbulent viscosity and $\bar{S}_{ij} = \frac{1}{2}(\nabla \bar{u}_i + \nabla \bar{u}_j)$ is the rate-of-strain tensor for the resolved scale. In various sub-grid scales these components are modeled differently for their respected operation or goal. In the WALE model [47] the eddy viscosity is modeled by

$$\mu_t = \rho L_s^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\bar{S}_{ij} \bar{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}} \quad (6.31)$$

where $L_s = \min(\kappa d, C_w V^{\frac{1}{3}})$ where C_w is 0.325 which has been found to yield good results for wide range of flow [47]. This method of modeling the eddy viscosity allows for proper behavior near the boundary walls since we are assuming that the cutoff length plays no important role in its behavior [49]. In this model as the sub-grid scale stress approaches the wall it should show behavior of order $O(y^3)$ if y is the direction normal to the wall. However, the behavior of $\bar{S}_{ij}$ is of order $O(1)$ in the same limit, which results in defining a new operator S_{ij}^d .

$$S_{ij}^d = \frac{1}{2}(\bar{g}_{ij}^2 + \bar{g}_{ji}^2) - \frac{1}{3}\delta_{ij}\bar{g}_{kk}^2 \quad (6.32)$$

In this case S_{ij}^d is based on the traceless symmetric part of the square gradient velocity tensor $\bar{g}_{ij}^2 = \frac{\partial u_i}{\partial x_j}$ [49]. This method of modeling the sub-grid scale should provide a local eddy-viscosity, able to switch off during the early stages of transition and offer a proper wall scaling to get a good prediction of friction coefficient.

6.4 IMPLEMENTING LES ON CURRENT NUMERICAL SIMULATION

To generate atmospheric turbulence over long distance laser propagation, a simplified 3D geometry was generated. This 3D model mimics the elevation profile along the propagation path from the Mauna Loa NOAA observatory to the AFRL AEOS telescope on Haleakala. These summits are the U.S. Air Force sites for conducting the optical turbulence studies [50]. The Maui Space Surveillance System (MSSS) is an electro-optical facility located at the summit of Haleakala an elevation of 3.055 km. The NOAA observatory located on top of the Mauna Loa summit reaches an elevation of approximately 3.397 km. The distance between both observatories reaches approximately 150 km length. Due to computational restriction, we were limited by the simplified geometry dimensions and made it as large as possible with a very coarse mesh to obtain a numerical simulation of fluid motion.

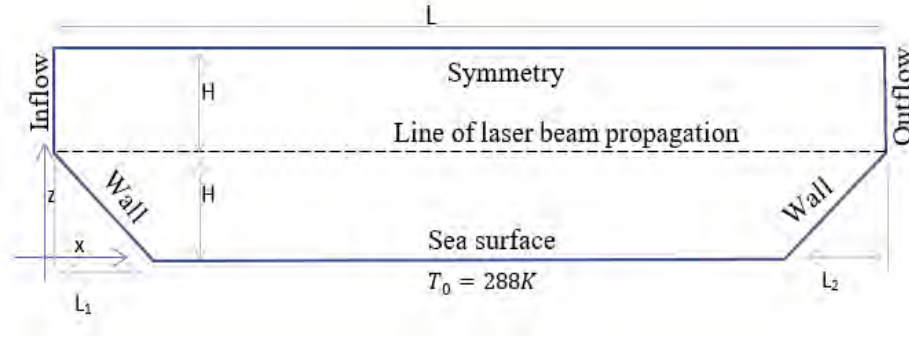


Figure 6.2: Schematic of computational domain

Table 6.2 Geometry of computational domain and inlet velocity

Geometry	Velocity
H=3m, L=150m, L1=10m, L2=20m, W=0.1m	4 m/s

Both observatories are assumed to be on the same horizontal axis 3 km above sea level. The dimensions L_1 and L_2 is an approximate distance of mountain peak to the beach. The inflow, outflow and top face surface are at an ambient air temperature of 298 K while the bottom face (sea level) is 288 K. The desired geometry was reduced by a factor of 10^3 m due to computational restraints. The remaining of the study will analyze the LES of the geometry model to see if it is possible to determine C_n^2 values to character atmospheric turbulence effects on laser propagation.

To be able to determine C_n^2 , we need to keep in mind that the variation in the refractive index is proportional to the variation in potential temperature and velocity. This allows us to relate the refractive-index constant, C_n^2 , to the temperature structure constant C_T^2 .

$$C_n^2 = \left(\frac{79 \times 10^{-8} P}{T^2} \right)^2 C_T^2 \quad (6.33)$$

where P is atmospheric pressure, T is air temperature. Values for temperature and pressure depend on the mean atmospheric temperature and pressure along the propagation path. C_T^2 is the structure constant parameter for temperature and can be calculated as

$$C_T^2 = a^2 \left(\frac{K_H}{K_M} \right) L_o^{\frac{4}{3}} \left(\frac{\partial \theta}{\partial x} \right)^2. \quad (6.34)$$

In this relation a^2 is an empirical constant, K_H and K_m are the exchange coefficients for heat and momentum, L_o is the outer length scale of turbulence, and $\frac{\partial \theta}{\partial x}$ is the gradient of potential temperature. D. L. Walters and D. K. Miller [51], [52] determined that the empirical constant to be 2.8. Masciadri, and E. Jabouille deduced that for a thermal and dynamic stability of the atmosphere, $\frac{K_H}{K_M}$ value can be assumed to be 0.7 [53]. The potential temperature, θ , is the temperature at an air parcel would have if it were expanded or compressed adiabatically to a pressure of 100 kPa.

$$\theta = T \left(\frac{10^5}{P} \right)^{0.286} \quad (6.35)$$

T is the static temperature and P is the absolute pressure in Pa over the propagation length.

Deardorff [54] calculated the outer scale length in thermally stable conditions to be

$$L_o = 0.76 \frac{\sqrt{k}}{N} \quad (6.36)$$

where N is the Brunt-Vaisala frequency and k is the turbulent kinetic energy. The Brunt-Vaisala frequency is also known as the buoyancy frequency. This frequency is associated with the oscillation caused by small density differences in a statistically stable environment.

$$N = \sqrt{-\frac{g}{\rho} \frac{\partial \rho}{\partial z}} \quad (6.37)$$

In atmospheric dynamics, the density difference is related to the potential temperature and hence the Brunt-Vaisala frequency can be given as

$$N = \sqrt{\frac{g}{\theta} \left| \frac{\partial \theta}{\partial z} \right|} \quad (6.38)$$

C_n^2 can ultimately be simplified as

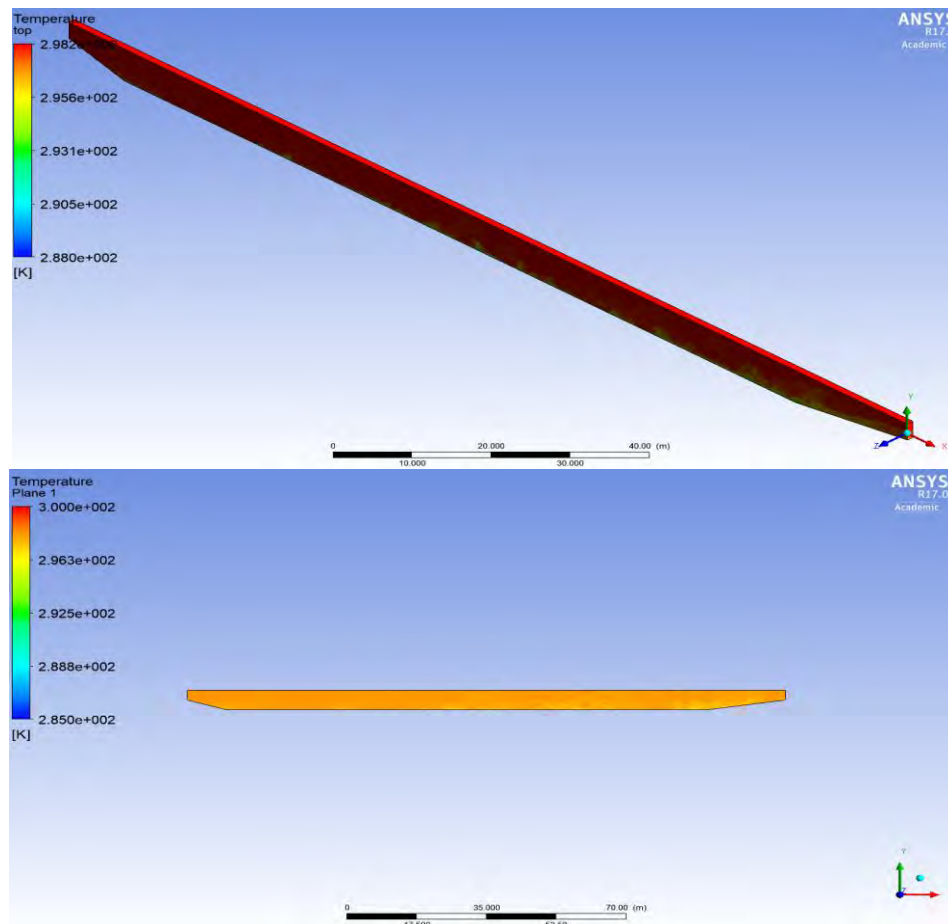
$$C_n^2 = 3.35 \times 10^{-6} P^{0.857} \theta^{-\frac{10}{3}} \left(\frac{\partial \theta}{\partial x} \right)^{\frac{4}{3}} (k)^{\frac{2}{3}} \quad (6.39)$$

In this case k angular spatial frequency. The derivation states that the optical refractive index fluctuations are caused by variations in the temperature in the atmosphere. It should be noted that C_n^2 is a variable that describes the strength of the refractive index turbulence. If we can model the

geometry that a laser will propagate and apply the appropriate boundary conditions, we can determine the structure function for laser propagation. This will then in return allow us to determine other optical parameters that help characterize atmospheric turbulence such as the Rytov Index.

6.5 CALCULATING FOR STRUCTURE FUNCTION CONSTANT

Once we get our results of our 3D model, we only need to attain the temperature and pressure profile at the line of sight.



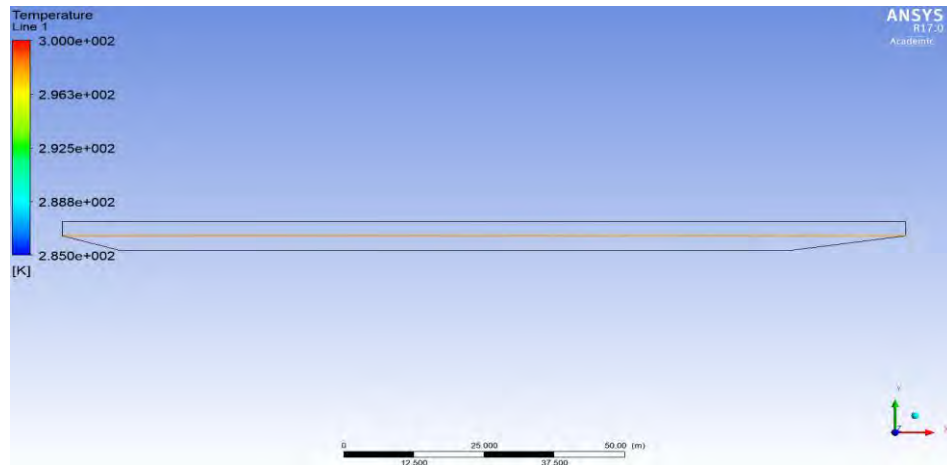


Figure 6.3: Temperature profile of our simplified 3D model

We then use equation (6.39) so solve the structure function constant as a function of atmospheric pressure and air temperature at time, $t = 200$, $t = 400$, and $t = 600$.

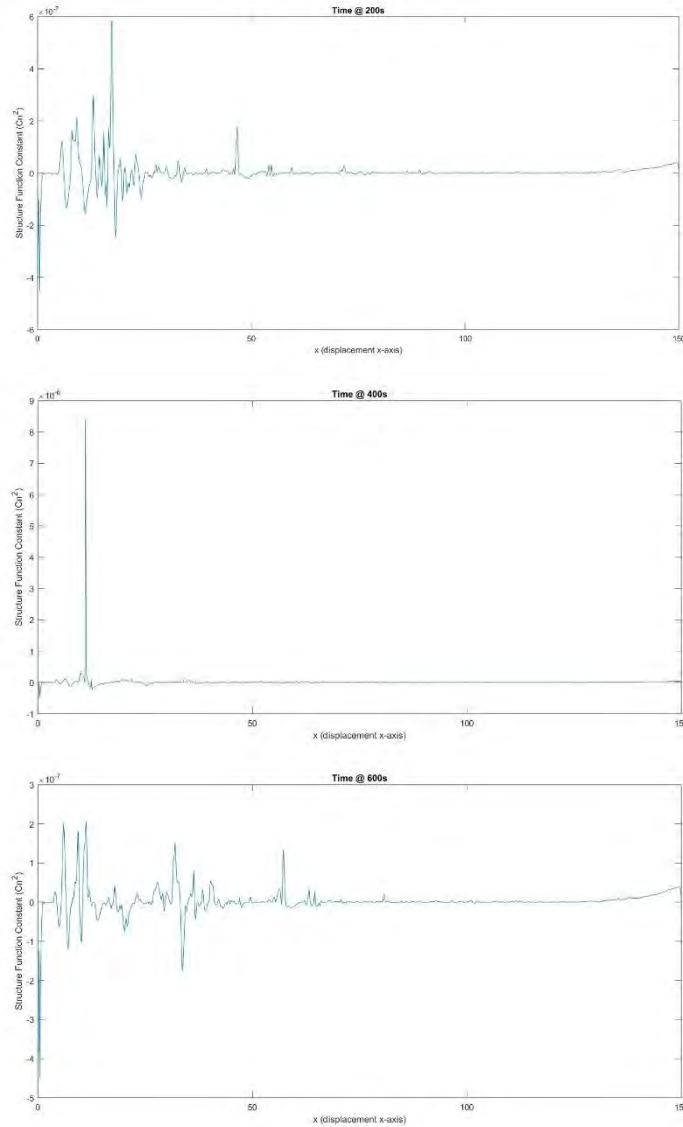


Figure 6.4: Calculations of the Structure Function Constant at, $t = 200$, $t = 400$, and $t = 600$

The average C_n^2 at the line of sight at $t = 200$ is $3.3335\text{e-}09$, at $t = 400$ is $1.1023\text{e-}08$, and at $t = 600$ is $2.6862\text{e-}09$. These values are extremely high for these conditions and there must be some error in our current modeling. Since we mainly use the Rytov index to represent atmospheric turbulence we can calculate C_n^2 , from a range of $0.1 \leq \sigma_x^2 \leq 1.0$. With the initial conditions, we can use equation (3.1) and determine that

within this range the structure constant should be within $1.8248 * 10^{-13} \leq C_n^2 \leq 1.8248 * 10^{-12}$. We seem to be off significantly, and further revision needs to be made to the model and the boundary conditions given.

If we are then able to model the actual propagation path between Mauna Loa NOAA observatory to the AFRL AEOS telescope on Haleakala we can then compare with experimental results. There are various publications that we can use to compare our simulated results with [50], [55]. However, due to the decrease in geometry, the Large Eddy Simulation is being compared with the current numerical simulation and theoretical approximation. Future work would include monitoring our model alongside previous and current experimental results of the propagational path.

Chapter 7: Putting it All Together

In this study, we analyzed atmospheric effects for a laser propagating a long distance through our developed phase screen simulator. We found that metrics based on the variance of intensity and number of zero intensity value saturate at high turbulence, but phase variance does not. Since the phase variance is dependent mainly on the largest eddies, we can make estimates of the outer scale length by determining the phase variance. However, we see that our values of phase variance fluctuate for $\sigma_x^2 \geq 0.5$. This may be due to the 2D Goldstein algorithm breaking down at high levels of atmospheric turbulence. To be more accurate, further investigation is needed at higher grid size and numbers of trials.

We also attempted to accelerate the numerical simulation by implementing a conditional generative adversarial network (cGAN) that applies phase unwrapping to a wrapped phase screen. The cGan framework establishes competition between two distinct players in the model, the discriminator and the generator. The network is trained on 512x512 wrapped and unwrapped images and learns the mapping to conduct phase unwrapping. We achieved reasonable results of generated images that differ slightly from the target images. However, we do run into complications when we tried to conduct phase unwrapping on the float matrices themselves. The images do not seem to be far off, but slight checkerboard artifacts are apparent in the images. We even attempted to improve our current architecture by using a different up-sampling method but did not generate better results. More in-depth understanding of neural networks and adjusting certain parameters could result in a working network. This will allow us to not only

conduct faster simulations but use larger grid size without much computational cost. This can in return allow us to study phase fluctuations at $\sigma_X^2 \geq 0.5$.

We would need to implement high-performance computing to run our phase screen simulator. This will allow us to simulate a spatial grid size that is significantly larger than the outer scale length so the sample frequency spacings can accurately represent outer scale effects. It can also help shorten the time of each simulation and include a greater number of trials to reduce any discrepancies. We have also generated a simple 3D model of the propagation path from the Mauna Loa NOAA observatory to the AFRL AEOS telescope on Haleakala. We do have licensing limitations and had to factor our model by 10^3m . With the given boundary conditions and our derived formulation of the structure-function constant our simulation values over exceed the theoretical. More work is required for improving the simplified model and changing some of the boundary conditions. If improved, we can begin to integrate our CFD simulation of the structure-function constant in our phase screen simulator.

The whole point of this study is to be able to code a numerical simulation of wave propagation for long-range optical systems. We will be able to analytically understand the effects without having to conduct the actual test themselves. This is beneficial since actual testing is not only economically expensive but requires a lot of time and manpower. We want to include our CFD simulation to our current numerical simulation to get an actual representation of C_n^2 across the propagation length instead of a constant C_n^2 . More importantly, we want to overcome the fluctuations cause by the 2D Goldstein Algorithm. We can improve on our current neural networks, but we can also increase the grid size by using high-performance computing. Once the

numerical simulation model works as needed, we can begin to characterize the effects of deep atmospheric turbulence on laser propagation over a long path.

References

- [1] A. Wheelon, *Electromagnetic Scintillation I. Geometrical Optics*, 1st ed. New York: Cambridge University Press, 2001.
- [2] J. M. McDonough, "LECTURES IN ELEMENTARY FLUID DYNAMICS : Physics , Mathematics and Applications," *Analysis*, p. 163, 2009.
- [3] A. Kolmogorov, *In Turbulence, Classic Papers on Statistical Theory*. New York: Interscience, 1961.
- [4] G. Weiss, "Wave Propagation in a Turbulent Medium. V. I. Tatarski. Translated by R. A. Silverman. McGraw-Hill, New York, 1961. 285 pp. Illus. \$9.75," *Science* (80-), vol. 134, no. 3475, pp. 324–325, Aug. 1961.
- [5] V. I. Tatarskii, *The Effects of the Turbulent Atmosphere on Wave Propagation*. Springfield: National Technical Information Service U.S. Dept. of Commerce, 1971.
- [6] J. W. Strohbehn, "Line-of-sight wave propagation through the turbulent atmosphere," *Proc. IEEE*, vol. 56, no. 8, pp. 1301–1318, 1968.
- [7] A. Quirrenbach, "The Effects of Atmospheric Turbulence on Astronomical Observations," *Exp. Astron.*, 2009.
- [8] L. C. Andrews and R. L. Phillips, *Laser Beam Propagation through Random Media*. 1000 20th Street, Bellingham, WA 98227-0010 USA: SPIE, 2005.
- [9] L. A. Chernov, *Wave propagation in a random medium. Translated from the Russian by R. A. Silverman*. New York: McGraw-Hill, 1960.
- [10] R. L. Phillips and L. C. Andrews, "Measured statistics of laser-light scattering in atmospheric turbulence," *J. Opt. Soc. Am.*, vol. 71, no. 12, p. 1440, 1981.
- [11] A. Consortini, F. Cochetti, J. H. Churnside, and R. J. Hill, "Inner-scale effect on irradiance variance measured for weak-to-strong atmospheric scintillation," *J. Opt. Soc. Am. A*, vol. 10, no. 11, p. 2354, 1993.
- [12] D. Dayton, B. Pierson, B. Spielbusch, and J. Gonglewski, "Atmospheric structure function measurements with a Shack-Hartmann wave-front sensor," *Opt. Lett.*, vol. 17, no. 24, p. 1737, 1992.
- [13] R. Kerr, "Experiments on Turbulence Characteristics and Multiwavelength Scintillation Phenomena," *J. Opt. Soc. Am.*, vol. 62, no. 9, pp. 1040–1049, 1972.
- [14] R. S. Lawrence, G. R. Ochs, and S. F. Clifford, "Measurements of atmospheric turbulence relevant to optical propagation," *J. Opt. Soc. Am.*, vol. 60, no. 6, pp. 826–830, 1970.
- [15] E. Velar, J. Haddon, and J. Haddon, "Measurement and Modeling of Scintillation Intensity

- to Estimate Turbulence Parameters in an Earth-Space Path,” *IEEE Trans. Antennas Propag.*, vol. 32, no. 4, pp. 340–346, 1984.
- [16] A. Zilberman, E. Golbraikh, and N. S. Kopeika, “Propagation of electromagnetic waves in Kolmogorov and non-Kolmogorov atmospheric turbulence: three-layer altitude model,” *Appl. Opt.*, vol. 47, no. 34, pp. 6385–91, 2008.
 - [17] J. P. Bos, M. C. Roggemann, and V. S. Rao Gudimetla, “Anisotropic non-Kolmogorov turbulence phase screens with variable orientation,” *Appl. Opt.*, vol. 54, no. 8, p. 2039, 2015.
 - [18] B. E. Stribling, “Optical propagation in non-Kolmogorov atmospheric turbulence,” in *Proceedings of SPIE*, 1995, vol. 2471, pp. 181–196.
 - [19] J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation with Examples in MATLAB*. 2010.
 - [20] C. Smith, “GoldsteinUnwrap2D_r1.” MATLAB 7.11 (R2010b), 2010.
 - [21] V. S. R. Gudimetla, R. B. Holmes, and J. F. Riker, “Analytical expressions for the log-amplitude correlation function for plane wave propagation in anisotropic non-Kolmogorov refractive turbulence,” *J. Opt. Soc. Am. A*, vol. 29, no. 12, p. 2622, 2012.
 - [22] V. S. Gudimetla, “Estimating Cn2 (Refractive index structure constant) in Deep Turbulence using Unwrapped Phase,” in *Imaging and Applied Optics*, 2013, p. PTu2F.1.
 - [23] R. S. Michalski, I. Bratko, and A. Bratko, Eds., *Machine Learning and Data Mining: Methods and Applications*. New York, NY, USA: John Wiley & Sons, Inc., 1998.
 - [24] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, “GeePS : Scalable deep learning on distributed GPUs with a GPU-specialized parameter server.”
 - [25] T. Chen *et al.*, “MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems,” pp. 1–6.
 - [26] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.”
 - [27] M. Isard, A. Birrell, and D. Fetterly, “Dryad : Distributed Data-Parallel Programs from Sequential Building Blocks,” 2007.
 - [28] J. Dean and S. Ghemawat, “MapReduce : Simplified Data Processing on Large Clusters,” pp. 1–13.
 - [29] J. Dean *et al.*, “Large Scale Distributed Deep Networks,” pp. 1–11.
 - [30] C. Bishop and N. Nasrabadi, “Pattern recognition and machine learning,” *Pattern Recognit.*, vol. 4, no. 4, p. 738, 2006.

- [31] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Miccai*, 2015.
- [32] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, “Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks,” in *Medical Image Understanding and Analysis: 21st Annual Conference, MIUA 2017, Edinburgh, UK, July 11–13, 2017, Proceedings*, 2017.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Alexnet,” *Adv. Neural Inf. Process. Syst.*, 2012.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.
- [35] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [36] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context Encoders: Feature Learning by Inpainting,” 2016.
- [37] I. Goodfellow *et al.*, “Generative Adversarial Nets,” *Adv. Neural Inf. Process. Syst.* 27, 2014.
- [38] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
- [39] M. Drozdal, E. Vorontsov, G. Chartrand, and C. V Sep, “The Importance of Skip Connections in Biomedical Image Segmentation.”
- [40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” pp. 1–10, 2016.
- [41] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and Checkerboard Artifacts,” *Distill*, vol. 1, no. 10, Oct. 2016.
- [42] T. Chacón Rebollo and R. Lewandowski, *Mathematical and Numerical Foundations of Turbulence Models and Applications*. New York, NY: Springer New York, 2014.
- [43] F. H. Harlow and J. E. Welch, “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface,” *Phys. Fluids*, vol. 8, no. 12, p. 2182, 1965.
- [44] H. K. Versteeg and W. Malaskechera, *An Introduction to Computational Fluid Dynamics*, vol. M. 2007.

- [45] S. V. Patankar and D. B. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *Int. J. Heat Mass Transf.*, vol. 15, no. 10, pp. 1787–1806, 1972.
- [46] Y. Zhiyin, "Large-eddy simulation: Past, present and the future," *Chinese J. Aeronaut.*, vol. 28, no. 1, pp. 11–24, 2015.
- [47] Ansys, "Ansys fluent 12.0 Theory Guide," *ANSYS Inc.*, no. April, pp. 49–53, 2009.
- [48] F. G. Schmitt, "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," *Comptes Rendus - Mecanique*, vol. 335, no. 9–10. pp. 617–627, 2007.
- [49] F. Ducros, F. Nicoud, and T. Poinso, "Wall-adapting local eddy-viscosity models for simulations in complex geometries," *Conf. Numer. Methods Fluid Dyn.*, pp. 1–7, 1998.
- [50] M. Vorontsov *et al.*, "Characterization of Atmospheric Turbulence Effects Over 149 km Propagation Path Using Multi-Wavelength Laser Beacons," Proc. AMOS Conference, 2010.
- [51] D. L. Walters and D. K. Miller, "Evolution of an upper-tropospheric turbulence event—Comparison of observations to numerical simulations," in *Preprints, 13th Symposium on Boundary Layer Turbulence*, 1999, pp. 157–160.
- [52] S. Berman and C. K. So, "Determination of the kondo scattering amplitude and coherence distance from tunneling in doped-electrode junctions," *Phys. Rev. Lett.*, vol. 40, no. 1, pp. 53–56, 1978.
- [53] E. Masciadri and P. Jabouille, "Improvements in the optical turbulence parameterization for 3D simulations in a region around a telescope," *Astron. Astrophys.*, vol. 376, no. 2, pp. 727–734, 2001.
- [54] J. W. Deardorff, "Stratocumulus-capped mixed layers derived from a three-dimensional model," *Boundary-Layer Meteorol.*, vol. 18, no. 4, pp. 495–527, 1980.
- [55] M. Vorontsov *et al.*, "Comparison of turbulence-induced scintillations for multi-wavelength laser beacons over tactical (7 km) and long (149 km) atmospheric propagation paths." Proc. AMOS Conference, 2011

Vita

I am Diego Alberto Lozano Jimenez and I am pursuing a master's in mechanical engineering at the University of Texas at El Paso. Under my faculty advisor Dr. Vinod Kumar, I interned as a graduate research assistant for the AFRL Summer Faculty Fellowship Program. During this time, I had to understand the effect of atmospheric turbulence in the field of Directed Energy and laser systems. With the limited time I was given I was able to simulate long distance laser propagation under atmospheric turbulence through the Kolmogorov and non-Kolmogorov models. With this current numerical simulation our goal is to characterize the effects of atmospheric turbulence on long range laser propagation. My task was to determine the variance, covariance, auto-correlation and structure function of the phase fluctuations.

The following year I continued my research by implementing Machine Learning to our current laser simulation. To accelerate simulation, we implement a conditional generative adversarial network (cGAN) that applies phase unwrapping to a wrapped phase screen. Additional work is required to conduct phase unwrapping on 4096x4096 wrapped images with our currently trained model.

I am finishing my research by studying the impact of strong atmospheric turbulence in spatial, temporal, and related spectral domains will be examined using Large Eddy Simulation (LES) turbulence modeling. LES turbulence modeling allows us to study strong fluid turbulence and can predict advection-dissipation at various energy spectrums. I have had the opportunity to also submit two publication under AMOS and SPIE conferences. Hopefully I will continue working in research and development for the Department of Defense or the Department of Energy.

This thesis/dissertation was typed by Diego Alberto Lozano Jimenez.

MACHINE LEARNING ANALYSIS TO CHARACTERIZE PHASE VARIATIONS IN
LASER PROPAGATION THROUGH DEEP TURBULENCE.

LUIS FERNANDO RODRIGUEZ SANCHEZ

Doctoral Program in Mechanical Engineering

APPROVED:

Vinod Kumar, Ph.D., Chair

V.S. Rao Gudimetla, Ph.D., Co-Chair

Arturo Bronson, Ph.D.

Heidi Taboada, Ph.D.

Jorge Munoz, Ph.D.

V M Krushnarao Kottedda, Ph.D.

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

Copyright ©

by

Luis Fernando Rodriguez Sanchez

2020

Dedication

I dedicate this dissertation to my wife Marta and my children Luis Fernando and Marta for all their permanent support in the realization of this dream. To my mom Lupita, for her wisdom and continuous support, my sister Angelica and her husband Konrad for their unconditional support at a time when my family really needed their help. To my brother David who has had a key role in the theory components of this dissertation and always has been my motivation to be his guide. To all my brothers and sisters. To all my in-laws for their unconditional support.

I also dedicate it to my dad Ricardo from whom I learned and inherited a taste for the Mechanical Engineering and the discipline for the study of the physical sciences.

Finally, I dedicate this dissertation to all my professors from whom I received a taste for the scientific knowledge.

MACHINE LEARNING ANALYSIS TO CHARACTERIZE PHASER VARIATIONS IN
LASER PROPAGATION THROUGH DEEP TURBULENCE.

by

LUIS FERNANDO RODRIGUEZ SANCHEZ

DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering

Title of Academic Dept. or Program as listed in the academic catalog

THE UNIVERSITY OF TEXAS AT EL PASO

May 2020

Acknowledgements

This work is supported by Air Force Office of Scientific Research (AFOSR) under Grant Number BAA-AFRL-AFOSR-2016-0007, and Department of Mechanical Engineering at University of Texas at El Paso. Simulations were running on XSEDE computational resources such as Bridges, Stampede, and Lonestar. I would also like to thank Dr. Stacie Williams, AFOSR Program Officer, for her hard work that helped to provide the grant.

I would like to pay my special regards to Dr. Vinod Kumar, my advisor, for his continued guide and support in all this walk towards the end of my doctoral studies. To Dr. Heidi Taboada, who was the main contact and from whom I received a lot of support for my doctoral studies at UTEP. To her husband, Dr. José Espiritu from whom I received unconditional advice.

To Dr. Rao Gudimetla, my coadvisor, for his unconditional support and advice while carrying out this research. To Dr. Arturo Bronson, for his patience and support in making the necessary corrections to this work. To Dr. V.M. Kottedda, for his continued help on supercomputational issues. To Dr. Jorge Munoz, which guidance in the topic Neural Networks was crucial for giving shape to this dissertation.

To all my evaluation committee who has been the unconditional guide to shape this dissertation, as well as to give the proper focus to my research. To all the members of the Dr. Kumar research team whose unconditional support gave the right structure to this dissertation, particularly Arturo Rodriguez and Arturo Schiaffino.

To UTEP, which is without a doubt a top university, for allowing me to fulfill this dream.
To Tec de Monterrey campus Monterrey because it opened the doors to me for starting my doctoral studies into that excellent institution.

To Dr. Nicolás Hendrichs and Dr. Alex Elías from Tec de Monterrey for their friendship and support on difficult moments.

To Dr. Isacc Jimenez and Dr. Hector Siller, who represented an enormous initial impulse for the realization of my doctoral studies.

To Professors Nardo Hernández and Lucio Saldívar who gave me the solid foundations of Physics and Chemistry that I currently use.

To Professors Víctor Medellín and Esthela Salado who gave me a strong training in Mathematics.

To Professor Silvia Licón, who taught me the art of order and cleanliness in writing, accounting records, and financial planning.

To all my professors at UTEP and Tec de Monterrey who shaped my doctoral studies.

Abstract

The present dissertation is focused on the analysis of the atmospheric conditions of a turbulent environmental system and its effects on the diffraction of a laser beam that moves through it. The study is based on the distance and height between the cusps of two mountains located in Hawaii and whose communication between two laboratories, placed precisely over these two top positions, is carried out by optical means.

The emitter system is located in the Mauna Loa mountain and the receiver in the Haleakala. The distance between both mountains is 150 km. The emitter system is at a height of 3.1 km and the receiver at 3.4 km.

The maritime environment in which both systems are found experiences continuous wind currents whose velocity takes the atmospheric conditions to the state of turbulence. The state of turbulence generates a series of rotating systems called Eddies, which act as a kind of lens that diffuses light because they change the refractive index of the air, affecting the phase of the laser beam used in the communication between the laboratories.

Therefore, the objective of this study has been to find a mathematical relation between the wind speed and the refractive index. Due to the importance of the Reynolds number and its implicit relation with the wind speed and the study of the fluids, the focus of the study has been to relate the Reynolds number and the refractive index.

The idea of generating an equation that relates the Reynolds number and the refractive index pursues the objective of serving as a theoretical basis for the creating of automatic focus adjustments to the optical systems used on the laser communication. Based on this equation, the optical systems can make adjustments only with the measurement of the wind velocity experienced at the location.

Table of Contents

Acknowledgements	v
Abstract	vii
Table of Contents	ix
List of Tables	xi
List of Figures	xiii
Chapter 1: Introduction	1
Chapter 2: Classical Theory of Optical Propagation	3
2.1 The Wave Equation of Optical Propagation	3
2.1.1 Solution by the Method of Small Perturbations	5
2.2 Effects of Turbulence on Optical Propagation	11
Chapter 3: The Kolmogorov Analysis	13
3.1 Structure Tensor	15
3.2 Turbulence Model of Refractivity Fluctuations	17
3.3 Effects of Turbulence on Optical Propagation	19
3.4 Behavior of the Refractive Index C_n^2 in Open Atmosphere	22
Chapter 4: Numerical Simulation of Laser Propagation	24
4.1 Fourier Transforms: from Analytical to Numerical	27
Chapter 5: Governing Equations of Fluid Mechanics for Turbulent Conditions	29
Chapter 6: Fundamentals of Computational Fluid Dynamics	32
6.1 Conservation of Mass	32
6.2 Conservation of Momentum	33
6.3 Conservation of Energy	35
6.3.1 Work	35
6.3.2 Heat Conduction	36
6.3.3 General Energy Equation	37
6.3.4 General Equilibrium Equation	37

Chapter 7: CFD Simulation	38
7.1 Model Design	38
7.1.1 Scaling the Model.	38
7.1.2 Justification of Modeling Scaling Down.....	40
7.1.2.1 Velocity Conditions	40
7.1.2.2 Temperature Conditions.	41
7.1.2.3 Pressure Conditions.	41
7.1.2.4 Testing.	41
7.1.3 3D Modeling.....	44
7.2 Meshing.....	46
7.3 Defining Boundary Conditions and Setting Up the Model.....	48
Chapter 8: Deep Learning Analysis	51
8.1 Neural Network Analysis.....	56
8.1 Mathematical Background	56
8.2 Neural Network Process	57
8.2.1 Importing Data.....	57
8.2.2 Training the Algorithm.....	58
8.2.2.1 Selecting the Data.....	58
8.2.2.2 Training.....	59
8.2.3 Prediction Process	59
8.2.4 Optimization	60
8.3 Mesh Comparison.....	61
Chapter 9: Problem Statement	63
9.1 Objective	65
9.2 General Conditions	65
Chapter 10: Analysis	68
Chapter 11. Results	73
Chapter 12. Conclusions.....	76
Chapter 13: Future Work.....	78

References	79
Vita.....	86

List of Tables

Table 10.1: <i>Reynolds</i> vs C_n^2	72
--	----

List of Figures

Figure 3.1: The Kolmogorov model of the energy cascade..	14
Figure 3.2: Rotational systems in turbulent media called eddies..	15
Figure 3.3: Changes in laser direction due to the presence of Eddies.....	19
Figure 3.4: Distribution of the C_n value during a typical summer day in Hawaii..	22
Figure 4.1: Definition of the source plane and observation plan inside a cartesian system.....	25
Figure 4.2: Distribution of the laser phase screen in a 2D map.....	26
Figure 6.1: Mass flow vectors entering and leaving an element volume.....	32
Figure 6.2: Components of the surface forces as shear stresses.	34
Figure 6.3: Mass flow vectors entering and leaving an element volume.....	36
Figure 7.1: General geometry under analysis.....	38
Figure 7.2: Scaling down the geometry.....	39
Figure 7.3: Final Geometry dimensions.....	39
Figure 7.4: Velocity simulated with scaled conditions... ..	42
Figure 7.5: Velocity simulated with original conditions.....	42
Figure 7.6: Temperature variations simulated with scaled conditions.....	43
Figure 7.7: Temperature conditions simulated with original conditions..	43
Figure 7.8: ANSYS Fluent Sections for fluid analysis..	44
Figure 7.9: 3D ANSYS solid model of the case of analysis.	45
Figure 7.10: Design tree for the ANSYS Meshing section.	46
Figure 7.11: Designation of named sections according to the original model.	46
Figure 7.12: Edge dimensioning for defining the element size of the mesh..	47
Figure 7.13: 3D mesh with 20 mm side elements.....	48

Figure 7.14: Components of the <i>Set up</i> section in ANSYS Fluent.....	49
Figure 7.15: Results of the velocity distribution of a 10 m/s inlet analysis.	50
Figure 7.16: Results of the temperature distribution of a 10 m/s inlet analysis.	50
Figure 8.1: Graphic definition of the neural network algorithm.....	51
Figure 8.2: Sigmoid Function (Left), ReLu Function (Right).....	52
Figure 8.3: Close view of the non-equidistant mesh generated in ANSYS Fluent.	57
Figure 8.4: Equidistant points created to simulate the original model.....	58
Figure 8.5: Distribution of the errors for Hidden Layers vs Neurons.....	61
Figure 8.6: Mesh comparison of meshes before and after Neural Network algorithm.....	62
Figure 9.1: Laboratory at the top of the mountain Mauna Loa.	63
Figure 9.2: Systems that use optical communication.	64
Figure 9.3:General Geometry.	65
Figure 9.4: Temperature and pressure data in the specified sections.	66
Figure 9.5: Codes to generate the UDF in C language for temperature and pressure.	67
Figure 10.1: Range of velocities for the 45 cases of analysis.....	68
Figure 10.2: Velocity behavior in the turbulent environment.	69
Figure 10.3: Non equidistant mesh from the ANSYS File.....	69
Figure 10.4: Transition from Non-Equidistant to Equidistant Mesh.	70
Figure 10.5: Temperature information in equidistant positions.	70
Figure 10.6: Range of Reynolds numbers.	72
Figure 11.1: Re vs C_n^2 trend.	74

Chapter 1: Introduction.

Laser propagation through atmospheric turbulence has been of interest by many scientists and engineers for centuries. Before the invention of lasers, astronomers observed the light coming from the stars distorted by a turbulent atmosphere. After the invention of lasers, scientists and engineers observed similar phenomenon specially on long distances, then many focused their efforts on understand it, particularly after the appearance of optical communication systems [1].

The first intentional approach is the use of analytical equations to understand the turbulent behavior of the atmosphere. The Navier-Stokes equation is excellent due to the inclusion of all the variables involved in the fluid phenomena, the truly weakness of this approach is the enormous quantity of calculations that takes part in every case of analysis. Even though talking about high performance computer capacity, the volume of fluids that can be fully calculated using Navier-Stokes equations is very small [2].

Then, a different approach is used for analyzing the behavior of the turbulent atmosphere, which is the theory of the energy cascade of Kolmogorov. This approach defines that the energy from the turbulent atmosphere is transmitted inside a spam called inertial subrange [3].

On the other hand, counting on the CFD fluid simulation resource by ANSYS Fluent, we decided to use it to make important contributions to the understanding of the turbulent conditions in the analyzed volume of turbulent air between two mountains.

Due to the random approach of ANSYS for mesh construction and the importance for our study to have a proportional mesh with specific values of pressure, temperature and wind velocity, we decided to use another source for exporting that data but work the results to obtain a

proportional matrix with the corresponding speed, pressure and temperature data within a proportional mesh.

This other resource is called neural networks, which are computational algorithms that estimate conditions based on a group of data used as training group. In the present case, the data corresponding to an atmospheric volume generated on a non-proportional mesh were used to train the algorithm and thereby estimate the temperature, pressure and speed conditions of another structured atmospheric volume based on a proportionate spatial mesh.

The need to have a proportional mesh is to calculate the optical parameter called diffraction constant C_n^2 . The objective of the thesis is to find the relation between the Reynolds number and the values of C_n^2 , and the values of C_n^2 are calculated on proportional deltas of distance, which accuracy would be enormously low basing the calculations on a non-proportional mesh [4].

At the end, the calculations of C_n^2 related to Reynolds number took us to the conclusion that there is a correlation of the Reynolds number over C_n^2 but only after velocities of 13 m/s that are present in the Hawaiian environment during the months of October through late January and May through July [5].

A future work is focused on relating data obtained from atmospheric devices to compare with the current analytical approach. The Dr. Rao, who belongs to the AFOSR laboratory at Maui, has validated these correlations as very close to the real turbulent behavior [1].

Chapter 2: Classical Theory of Optical Propagation.

The general description of our project starts with the theoretical bases of light propagation or more commonly known as optical propagation. It is worth remembering that a general accepted way to understand the propagation of light is to understand the light phenomena it as a wave of an electromagnetic nature. Then, for this reason, Maxwell's equations have a clear application [6].

During this chapter, theoretical results will be obtained from Maxwell's equations, using a perturbation technique described by Tatarskii [7]. For the sake of simplicity, a plane wave is assumed to be incident upon the random medium and just the resultant fluctuations of amplitude and phase are considered.

2.1. The Wave Equation for Optical Propagation.

The atmosphere is assumed to have zero conductivity and unitary magnetic permeability, while the electromagnetic field is assumed to have a sinusoidal time dependence, given by $e^{-i\omega t}$. Amplitude and phase statistics for short pulses can be derived by Fourier decomposition of their waveform into a spectrum of plane waves [8]. Given such conditions, Maxwell's equations are written as follows:

$$\nabla \cdot H = 0 \quad (2.1)$$

$$\nabla \times E = ikH \quad (2.2)$$

$$\nabla \times H = -ikn^2E \quad (2.3)$$

$$\nabla \cdot (n^2E) = 0 \quad (2.4)$$

Where $i = \sqrt{-1}$, $k = \frac{\omega}{c}$ stands for the wave number of the radiation, ω for its radian frequency, and c is the speed of light in free space. The operator ∇ is the vector derivative $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. Both E and H correspond to the vector amplitudes of the electric and magnetic field, respectively, and are position dependent functions. The assumed sinusoidal time dependence is contained in k . Finally, $n(r)$ is the atmospheric refractive index in a position dependent random function [9].

The most commonly known wave equation for the electric field is obtained by substituting (2.3) in (2.2) as seen below:

$$-\nabla^2 E + \nabla(\nabla \cdot E) = k^2 n^2 E \quad (2.5)$$

Then (2.4) is expanded and solved for $\nabla \cdot E$ and the result inserted into (2.5) to obtain:

$$\nabla^2 E + k^2 n^2 E + 2\nabla(E \cdot \nabla \log(n)) = 0 \quad (2.6)$$

In (2.5), $2\nabla(E \cdot \nabla \log(n))$ refers to the change in polarization as the wave propagates, which is considered negligible for $\lambda \ll l_0$ [2]. Depolarization effects of the atmosphere are even negligible for $\lambda > l_0$. Thus, (2.6) can be rewritten as:

$$\nabla^2 E + k^2 n^2 E = 0 \quad (2.7)$$

Finally, the desired results are obtained by solving equation (2.7).

2.1.1. Solution by the Method of Small Perturbations.

The method consists of a series of decreasing E terms. As assumption, each E_m is of the order of smallness $(n_1)^m$. If $|E_{m+1}| \ll |E_m|$, then the series converge, and propagation is accurately described. Otherwise, the series would diverge. Physically, the order of the terms represents the scattering of the wave [10]. The zero-order for unscattered, first order for single scattering, and so on. For the current analysis, just the single scatter case is solved.

First, $E = E_0 + E_1$ is inserted into (2.7) and terms of the same order in n_1 are gathered and equated to zero, as follows:

$$\nabla^2 E_0 + k^2 E_0 = 0 \quad (2.8)$$

$$\nabla^2 E_1 + k^2 E_1 + 2k^2 n_1 E_0 = 0 \quad (2.9)$$

The product $E_1 n_1$ is of order n_1^2 , therefore, it is ignored. As *Tataraskii* [13], the incident field is assumed to be a unit amplitude plane wave propagating in the z direction, $E_0 = e^{ikz}$, so (2.9) is rewritten as:

$$\nabla^2 E_1 + k^2 E_1 = -2k^2 n_1 e^{ikz} \quad (2.10)$$

Now, its solution is the convolution of the term on the right side with the Green's function of the equation:

$$E_1(r) = (4\pi)^{-1} \int_V d^3 r' \frac{e^{ik|r-r'|}}{|r-r'|} [2k^2 n_1(r') e^{ikz'}] \quad (2.11)$$

It is stated that the scattered field observed in r is a spherical wave emitted at r' , with an amplitude proportional to the product of the local refractive index $n_1(r')$ and the strength of the incident radiation.

The equation in (2.11) can be simplified by the characteristics of laser propagation: Monochromatic, scattered by weak large-scale refractivity fluctuations and contained in a narrow cone in the forward scatter direction (+z). From laws of diffraction, the maximum scattering angle is in the smallest eddies traversed. In this case, the scattering angle $\frac{\lambda}{l_0}$ is 3×10^{-4} radians, therefore, the transverse displacement from the forward propagation from scatterer to receiver $|q - q'|$ is much smaller than the distance between them $|z - z'|$. That is, if $|z - z'| \gg |q - q'|$, then $r - r'$ can be replaced by $z - z'$ in the denominator of (2.11). For the argument, the wavelengths that are contained within this distance must be considered. Thus, the terms that are retained are the ones that make the argument larger than 2π radians [1]. Then, $|r - r'|$ is expanded into a binomial series, and after reducing it by the Fresnel approximation, the following equation is obtained:

$$E_1(r) = \frac{k^2 e^{ikz}}{2\pi} \int_V d^3 r' e^{\frac{ik[(q-q')^2]}{2(z-z')}} \frac{n_1(r')}{(z-z')} \quad (2.12)$$

This equation is the Fresnel diffraction formula and is the solution to the diffusion equation when the source term is $-2k^2 n_2$. Now, in order to calculate how the amplitude and phase fluctuate when crossing the atmosphere relatively to the free space, the field E is rewritten as:

$$\frac{E}{E_0} = \frac{k^2 e^{ikz}}{2\pi} \int_V d^3 r' e^{\frac{ik[(\rho-\rho')^2]}{2(z-z')}} \frac{n_1(r')}{(z-z')} \quad (2.13)$$

This is the Fresnel diffraction formula and is the solution to the diffusion equation when the source term is $-2k^2 n_2$. In addition, the exponential term predicts the location of the interference fringes produced in the receiving plane in z by the interference of the incident plane wave and the scattered spherical wave (Huygen's wavelet) emitted from (ρ', z') .

Now, to calculate how amplitude and phase fluctuate when crossing the atmosphere relatively to the free space [11], the field $E = E_0 + E_1$ must be written in terms of both amplitude and phase as follows:

$$\frac{E}{E_0} = 1 + \frac{E_1}{E_0} = \frac{A}{A_0} e^{i(s-s_0)} \quad (2.14)$$

The last term results from $E = Ae^{is}$ and $E_0 = A_0 e^{is_0}$, where A_1 is the amplitude of E_1 . Then, its natural logarithm is taken. Since $\left| \frac{E_1}{E_0} \right| \ll 1$ and $\frac{A_1}{A_0} \ll 1$, the logarithm is expanded into a power series and just the first term is retained, as follows:

$$\frac{E_1}{E_0} \cong \frac{A_1}{A_0} + i(s - s_0) \quad (2.15)$$

Equation (2.15) implies that, when normalized by E_0 , amplitude ratio and phase fluctuations must be obtained from (2.12), from its real and imaginary parts, respectively [12]. Resulting in:

$$\frac{A_1}{A_0} = \frac{k^2}{2\pi} \int_V d^3 r' \cos \left[\frac{k(\varrho - \varrho')^2}{2(z - z')} \right] \frac{n_1(r')}{(z - z')} \quad (2.16)$$

$$(s - s_0) = \frac{k^2}{2\pi} \int_V d^3 r' \sin \left[\frac{k(\varrho - \varrho')^2}{2(z - z')} \right] \frac{n_1(r')}{(z - z')} \quad (2.17)$$

Where there is no wave attenuation as $n_1(r')$ is real. Now, log amplitude $\chi = \ln \left(\frac{A_1}{A_0} \right) \cong \frac{A_1}{A_0}$ (what is often measured experimentally) and $s - s_0 = s_1$ are defined by convention. It should be noted, though, that χ is identical to $\frac{A_1}{A_0}$ for weak scatterings as in the current case. Furthermore, as the analysis is considering a plane wave, statistical homogeneity of both amplitude and phase is expected to remain constant in the plane z [13]. On the other hand, χ and s are expected to fluctuate along the z axis (in the propagation direction), with a growth directly proportional to the distance travelled as it propagates through more refractive turbulence. Consequently, $n_1(r')$ is expanded into a two-dimensional Fourier-Stieltjes transform as follows:

$$n_1(r') = \int d\nu(K, z') e^{iK \cdot \varrho'} \quad (2.18)$$

Where the variation of $n_1(r')$ is transformed only in the plane transversal to the propagation direction and $d\nu$ is set as function of z' . Then, equation (2.17) is inserted into (2.15) and (2.16), and integration over $d\nu$ is substituted with that over $d^3 r'$ (since $d^3 r' = d^2 \varrho' dz'$) to obtain:

$$\left[\chi(r) \right]_{s_1(r)} = \frac{k^2}{2\pi} \int_0^z dz' \int d^2 \varrho' e^{iK \cdot \varrho'} \left\{ \frac{\cos \left[\frac{k(\varrho - \varrho')^2}{2(z - z')} \right]}{\sin \left[\frac{k(\varrho - \varrho')^2}{2(z - z')} \right]} \right\} \quad (2.19)$$

The integral over ϱ' is performed after substituting $\varrho'' = \varrho' - \varrho$. Now, since the sine and cosine terms are function of $|\varrho''|$, polar coordinates $\varrho'' = (\varrho'', \theta)$ can be used to integrate directly over θ . Additionally, the inner integrals are rewritten in terms of $I_{1,2}$, as follows:

$$I_{1,2} = e^{iK \cdot \varrho} \int_0^\infty d\varrho'' \varrho'' \left\{ \frac{\cos \left[\frac{k\varrho''^2}{2(z - z')} \right]}{\sin \left[\frac{k\varrho''^2}{2(z - z')} \right]} \right\} \int_0^{2\pi} d\theta e^{iK \varrho'' \cos(\theta - \psi)} \quad (2.20)$$

Where $K = (K, \psi)$ is let in the last term. The solution of the inner integration is the Bessel function $2\pi J_0(K\varrho'')$, which is then substituted in (2.20):

$$I_{1,2} = 2\pi e^{iK \cdot \varrho} \int_0^\infty d\varrho'' \varrho'' J_0(K\varrho'') \left\{ \frac{\cos \left[\frac{k\varrho''^2}{2(z - z')} \right]}{\sin \left[\frac{k\varrho''^2}{2(z - z')} \right]} \right\} \quad (2.21)$$

The remaining integral is the Hankel transform of the trigonometric term, which, when solved, (2.20) becomes:

$$I_{1,2} = \frac{2\pi(z - z')}{k} e^{iK \cdot \varrho} \left\{ \frac{\sin \left[\frac{k\varrho''^2}{2(z - z')} \right]}{\cos \left[\frac{k\varrho''^2}{2(z - z')} \right]} \right\} \quad (2.22)$$

The resulting expression is then inserted into (2.19) to obtain:

$$\left[\chi(r) \right]_{s_1(r)} = \int e^{iK \cdot \varrho} \left(k \int_0^z dz' dv(K, z') \left\{ \frac{\sin \left[\frac{K^2(z - z')}{2k} \right]}{\cos \left[\frac{K^2(z - z')}{2k} \right]} \right\} \right) \quad (2.23)$$

This equation is interpreted as a two-dimensional Fourier expansion as (2.18). Therefore, the terms in brackets represent the random spectral amplitudes of both amplitude and phase fluctuations [14].

As in any random process, the properties of the outcome of experiments were averaged. Consequently, the two-dimensional covariance functions $B_\chi(\varrho)$ and $B_s(\varrho)$ are used to settle the statistical homogeneity of amplitude and phase in the plane $z = \text{constant}$. (It should be noted that $\langle \chi \rangle = \langle s_1 \rangle = 0$ since $\langle n_1 \rangle = 0$ by definition and because only the first order term E_1 was retained.) Now, since both functions are often computed in the receiving plane located at $z = L$, they're defined as $B_\chi(\varrho, z) = \langle \chi(\varrho_1 + \varrho, z) \chi^*(\varrho_1 + z) \rangle$ and $B_s(\varrho, z) = \langle s(\varrho_1 + \varrho, z) s^*(\varrho_1 + z) \rangle$. Then, (2.23) is rewritten as:

$$\begin{aligned} & \begin{bmatrix} B_\chi(\varrho) \\ B_s(\varrho) \end{bmatrix} \\ &= k^2 \int \int e^{iK \cdot (\varrho_1 + \varrho) - iK' \cdot \varrho_1} \int_0^z dz' \int_0^z dz'' \left\{ \frac{\sin \left[\frac{K^2(z - z')}{2k} \right]}{\cos \left[\frac{K^2(z - z')}{2k} \right]} \frac{\sin \left[\frac{K'^2(z - z'')}{2k} \right]}{\cos \left[\frac{K'^2(z - z'')}{2k} \right]} \right\} \\ & \cdot \langle dv(K, z') dv^*(K', z'') \rangle \end{aligned} \quad (2.24)$$

Furthermore, the equation for the two-dimensional random amplitude dv is:

$$\langle dv(K, z') dv^*(K', z'') \rangle = \delta(K - K') F_n(K', z' - z'') d^2K d^2K'' \quad (2.25)$$

Where $F_n(K', z' - z'')$ is the two-dimensional spectral density of $n_1(r')$, defined by:

$$F_n(K', z' - z'') = \int_{-\infty}^{\infty} dK_z \phi_n(K', K_z) \cos[K_z(z' - z'')] \quad (2.26)$$

The ϕ_n term is the three-dimensional refractivity spectrum. Then, (2.24) is inserted into (2.23), where the K' integration is performed to obtain:

$$\begin{aligned} \begin{bmatrix} B_\chi(q) \\ B_s(q) \end{bmatrix} &= \int d^2K e^{iK \cdot q} \\ &\cdot \left(k^2 \int_0^z dz' \int_0^z dz'' \left\{ \frac{\sin \left[\frac{K^2(z-z')}{2k} \right]}{\cos \left[\frac{K^2(z-z')}{2k} \right]} \frac{\sin \left[\frac{K'^2(z-z'')}{2k} \right]}{\cos \left[\frac{K'^2(z-z'')}{2k} \right]} \right\} F_n(K, z' - z'') \right) \end{aligned} \quad (2.27)$$

The terms in brackets correspond to the two-dimensional spectral densities of χ and s , as demonstrated by their Fourier transform, which are then rewritten as $F_\chi(K, 0)$ and $F_s(K, 0)$.

Consequently, (2.27) is simplified in three phases [15]. First, it is noted from (2.26) that $F_n(K, z' - z'') = F_n(K, z'' - z')$. Second, since F_n correlates n_1 along adjacent planes along the z axis, only homogeneities whose size $2\pi K^{-1} \geq |z' - z''|$ provide significant contributions. Therefore, F_n is expected to decay sharply for $K|z' - z''| > 1$. Then, (2.27) is rewritten as:

$$\begin{bmatrix} F_\chi(K, 0) \\ F_s(K, 0) \end{bmatrix} = k^2 \int_0^L d\xi F_n(K, \xi) \int_{\frac{\xi}{2}}^{L-\frac{\xi}{2}} d\eta \left[\cos \left(\frac{K^2 \xi}{2k} \right) \mp \cos \left(\frac{K^2(L-\eta)}{k} \right) \right] \quad (2.28)$$

Where $L = z$ and the transformation of variables $\xi = z' - z''$, $2\eta = z' + z''$ is carried out. Finally, since $F_n \rightarrow 0$ for $\xi > K^{-1}$ and $K_{max} \sim 2\pi l_0^{-1}$ (where l_0 is the microscale of the turbulence), the argument of $\cos \left(\frac{K^2 \xi}{2k} \right)$ satisfies $\frac{K^2 \xi}{2k} \leq \frac{K}{2k}$, and $\frac{K}{2k} \ll 1$, respectively. Therefore, $\cos \left(\frac{K^2 \xi}{2k} \right) \sim 1$ for entire integration over ξ . Furthermore, by assuming $\xi \ll L$, its presence can be ignored within

the η integration limits, as K^{-1} has a maximum value of $\frac{L_0}{2\pi}$ (thus, $\xi \leq L_0$) for the Kolmogorov spectrum and region of integration $\xi \leq K^{-1}$. Lastly, if $L_0 \ll L$ then the ξ dependence is completely inserted into the η integral. This implies that the path must be much longer than the outer scale, which is satisfied when $L_0 \sim 1 \text{ m}$, near the ground [15]. Consequently, both integrals in (2.27) become independent and two expressions are obtained:

$$\int_0^\infty d\xi F_n(K, \xi) = \pi \phi_n(0, K) \quad (2.29)$$

Where the upper limit is replaced by infinity with negligible error, since $\xi \leq L_0$.

$$\begin{bmatrix} F_\chi(K, 0) \\ F_\varsigma(K, 0) \end{bmatrix} = \pi k^2 L \left[1 \mp \left(\frac{k}{K^2 L} \right) \sin \left(\frac{K^2 L}{k} \right) \right] \phi_n(K) \quad (2.30)$$

Chapter 3: The Kolmogorov Analysis.

The turbulence in the Earth's atmosphere is directly related to random variations in temperature and convective air movement, which alters the refractivity index of the air, both in position and in time. As optical waves propagate through the turbulent air, these waves are distorted by fluctuations in the refractive index of the atmosphere [16].

The equation of the refractive index of air referred to the optical frequencies is presented in equation (3.1). This equation will later be related to the turbulence equation to model its effects [18].

$$n - 1 = 77.6(1 + 7.52 \times 10^{-3}\lambda^{-2})(P/T)10^{-6} \quad (3.1)$$

In this equation, P represents pressure in millibars, T represents temperature in K degrees, and λ is the wavelength of light in μm (micrometers). Then, $n - 1$ is closely related to the deviation of the refractivity index. The typical value at sea level is 3×10^{-4} .

The present study is focused on analyzing the simulation of the temperature changes and analyzing their effects in the refractive index to lay the foundations for future mathematical modeling that will allow us to overcome these effects [17].

To perform the simulation of the phenomenon of turbulence, one approach is the analytical use of the Navier-Stokes equations, however, as mentioned in the introduction, the volume of equations and operations required to solve the model is enormous; due to its size, other calculation tools were sought. Even so, the model is the basis of the CFD (Computational Fluid Dynamics) simulations used in this study to generate temperatures, pressures and velocities all over the volume under analysis [18].

In this way, our analysis was based on an approach that demands less amount of information and can help us to understand the phenomenon of turbulence through the Theory of Energy Cascade, proposed by the Russian mathematician Andrey N. Kolmogorov. For this reason, it is known as the Kolmogorov's model [19].

Kolmogorov's idea is that when the kinetic energy enters a fluidic system, it does so through rotational systems called eddies. These eddies can be identified through their bigger diameters that are also called outer scale lengths, identified by L_o . whose size is larger when the process of transmitting kinetic energy starts into the system begins. Once the energy enters the major size eddies, it dissipates into smaller rotational systems until those systems reach the smallest diameters known as inner scale lengths or l_o . The Reynolds number corresponding to the eddies is $Re_o = u_o l_o$ [20].

After the eddies have reached their minimum size or the inner scale length, the rest of the kinetic energy dissipates as heat. All this phenomenon must occur within a range called inertial subrange, and Figure 2.1 shows how this dissipation occurs [21].

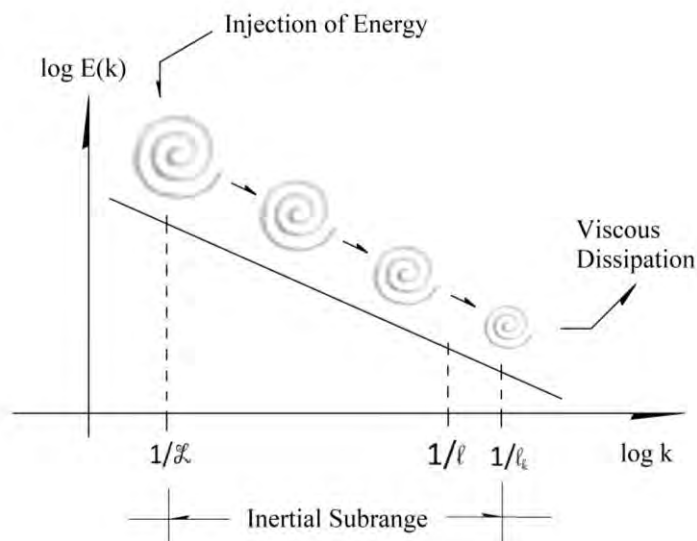


Figure 3.1: The Kolmogorov model of the energy cascade.

Figure 3.2 shows a turbulent system where the different sizes of eddies can be observed depending on the degree of kinetic energy dissipation.

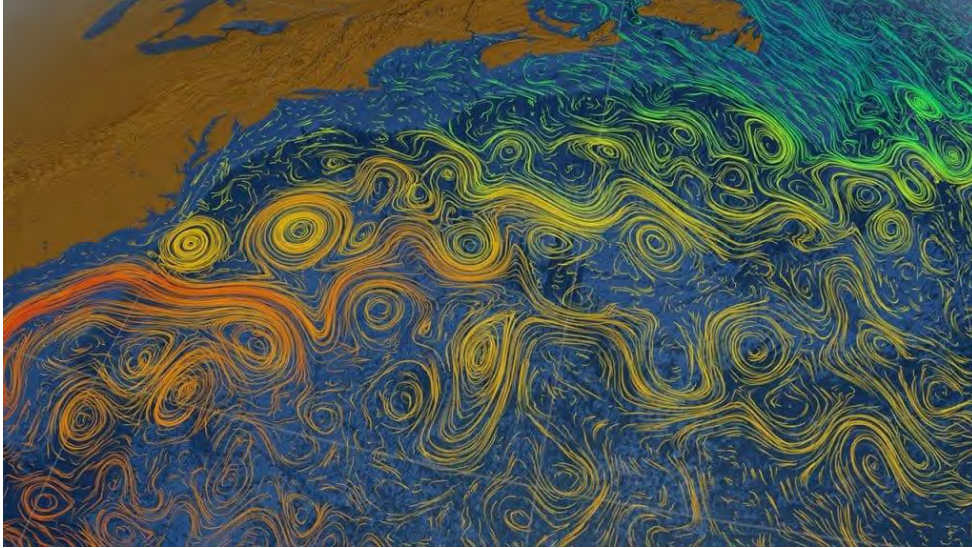


Figure 3.2: Rotational systems in turbulent media called eddies.

3.1. Structure Tensor.

Based on the presence of the eddies in the turbulent environment, Kolmogorov proposes the structural equation of velocity.

The idea is to consider the velocities of two different points in the space which distance is defined by the vector $\mathbf{r}$ [22]. He has found that the mean-square velocity of the velocity difference could be described by a form called structure function, which is defined as:

$$D_{ij}(\mathbf{r}) = \langle [v_i(\mathbf{r}_1 + \mathbf{r}) - v_i(\mathbf{r}_1)][v_j(\mathbf{r}_1 + \mathbf{r}) - v_j(\mathbf{r}_1)] \rangle \quad (3.2)$$

The structure equation in this case is referred to velocities in the different components. Important assumptions of homogeneity and isotropy are necessary to make this equation valid. The homogeneity establishes that the difference in velocity is only related to the displacement vector r . The isotropy of the magnitude r is the most important concept.

The previous assumptions lead to the description of the turbulence as a single structure function D_{rr} , thus, the following equation can describe it:

$$D_{rr} = \langle [v_r(r_1 + r) - v_r(r_1)]^2 \rangle \quad (3.3)$$

Considering that r is within the inertial subrange, thus D_{rr} has the universal form

$$D_{rr} = C_v^2 r^{2/3} \quad l_o < r < L_o \quad (3.4)$$

C_v^2 is the structural constant of velocity, which has a direct relationship to the total amount of energy in the turbulent flow [23]. As mentioned above, this equation holds for the values of r that lie at the microscale level l_o to the outer turbulence scale L_o .

The microscale l_o corresponds to the lowest eddie size values in the limit of energy dissipation through viscosity effects. The outer scale L_o corresponds to the largest size of the scale for which the eddies can be considered isotropic. Near the ground the l_o has an order of size of millimeters while L_o has an order of meters. Both amounts show an increase with height at floor level [25].

3.2. Turbulence Model of Refractivity Fluctuations

The crucial point is to relate the turbulent model that Kolmogorov proposes to the refractivity equation. A key concept to establish this relation is the conservative passive additive defined by Professor Valerian Tatarskii [14]. Conservative quantities are the ones related to an identified air parcel and that parcel will keep the changes referred to those quantities no matter how much the parcel moves into space.

This is the reason why the way of analyzing temperature, pressure and speed must be based on the fluctuations of these properties. Because the variations in these properties follow a pattern similar to the variation in the velocity that was previously demonstrated through the velocity tensor, therefore the variations in temperature can be expressed using the structural constant of thermal variation.

$$D_T = C_T^2 r^{2/3} \quad (3.5)$$

Additionally, because refractivity has also a direct relationship with the velocity variations, the structural refractivity constant can as well be expressed using the same structural relation.

$$D_n = C_n^2 r^{2/3} = 8\pi \int_0^\infty k^2 \Phi_n(k) \left[1 - \frac{\sin(kr)}{kr} \right] dk \quad (3.6)$$

The previous relationship established with the concept of structural refractivity constant and the power spectral density of Kolmogorov (Φ), it is possible to determine the complete function of the power spectral density. This function can be calculated by inverting the previous equation in and solving it for the variable.

$$\Phi(k) = \frac{1}{4\pi^2 k^2} \int_0^\infty \frac{\sin(kr)}{kr} \frac{d}{dr} \left[r^2 \frac{d}{dr} D_n(r) \right] dr \quad (3.7)$$

k is the angular frequency measured in rad/m and is equal to $2\pi/\lambda$. Then, working with the previous equations, the following function can be obtained. This equation is valid in the Kolmogorov's inertial subrange [21].

$$\Phi_n(k) = 0.033 C_n^2 k^{-11/3} \quad \frac{2\pi}{L_o} < k < \frac{2\pi}{l_o} \quad (3.8)$$

The power spectral density is the probability function associated with atmospheric turbulence. It is important to comment that there are turbulent atmospheric observations that deviate from the Kolmogorov's model, so it has also been necessary to find a statistical model that more accurately represents the effects of atmospheric turbulence [19].

Finally, relating the structural equations for the constants of temperature and refractivity, we have the final equation for the structural constant of refractivity as a function of pressure, temperature and the structural constant of thermal variation [23].

$$C_n^2 = \left(\frac{79 P}{T^2} \times 10^{-6} \right)^2 C_T^2 \quad (3.9)$$

The variations in temperature and pressure, as well as the structural variable of the temperature, were used to perform the calculations of the structural constant of refractivity for the present project.

3.2. Effects of Turbulence on Optical Propagation

Changes in the refractive index due to fluctuations in temperature are regularly smooth random functions of space and time. An auxiliary figure to understand the effects of variations in the refractive index is interpreting the presence of rotational systems or eddies as lenses [16].

In this model, the atmosphere could be assumed as a large number of random lenses with different shapes and sizes that move randomly into the space. The Figure 3.3 shows this phenomenon where the laser beam that leaves the emitting device, finds the different eddies that act as lenses and thus suffers variation in the direction due to different diffraction indices along the path, until it reaches the detector device [18].



Figure 3.3: Changes in laser direction due to the presence of eddies.

In the previous sections, the induced atmospheric effects on the refractive index were already characterized. In this section we will observe the propagation of light as a flat wave front that moves through turbulent flow. The phase change is produced by changes in the refractive index of an atmospheric layer of thickness δh and height h as shown in the following equation [20]:

$$\varphi(x) = k \int_h^{h+\delta h} dz n(x, z) \quad (3.10)$$

The phase change can be affected by different variables in addition to the refractive index, for example layers that are thicker than the turbulent cell. This is the reason why Gaussian statistics are used to estimate the phase changes, given the statistical properties of the refractive index.

Therefore, we can start this analysis by deriving the statistical behavior of a wavefront.

$$U = e^{i\omega(x)} \quad (3.11)$$

The coherence function $B_h(\rho)$ needs to be denoted in terms of phase structure function with the purpose of doing this [18].

$$B_h(r) = \langle U(x)U^*(x+r) \rangle \quad (3.12)$$

$$Bh(r) = \langle e^{(i[\varphi(x) - \varphi(x+r)])} \rangle \quad (3.13)$$

Due to the Gaussian behavior of $[\varphi(x) - \varphi(x+r)]$, then the following relationship can be applied

$$\langle e^{(\alpha\chi)} \rangle = e^{\left(\frac{1}{2}\alpha^2\langle\chi^2\rangle\right)} \quad (3.14)$$

Therefore, the following equations can be derived [10]:

$$B_h(\rho) = e^{\left[-\frac{1}{2} \langle |\varphi(x) - \varphi(x+r)|^2 \rangle\right]} \quad (3.15)$$

$$B_h(\rho) = e^{\left[-\frac{1}{2} D_\varphi(r)\right]} \quad (3.16)$$

Due to the chaotic behavior of the turbulent atmosphere, the stochastic processes such as velocity or temperature cannot be approximated as stationary. In any case, these stochastic processes can still be analyzed statistically if they have stationary increases. Instead of focusing on the stochastic process based on time or a point in space $x(r)$, we center the analysis on the function $x(r + r_1) - x(r_1)$, that has a behavior close to stationary.

The coherence function and the structure function can be simplified using the Fried Parameter.

$$B_h(\rho) = \exp\left[-3.44 \left(\frac{\rho}{r_o}\right)^{5/3}\right] \quad , \quad D_\varphi(\rho) = 6.88 \left(\frac{\rho}{r_o}\right)^{5/3} \quad (3.17)$$

The coherence function and the phase structure function depend on the Fried Parameter [24], which is a function of the intensity of the turbulence, the zenith angle and the wavelength. This parameter will be widely used in characterizing the effects of atmospheric turbulence on the phase changes of the laser propagating into that medium.

3.3. Behavior of the Refractive Index C_n^2 in Open Atmosphere

Although the concept of refractivity index was previously referred to, with this chapter the purpose is to delve a little deeper into the concept commenting on important references about the behavior of it in the Earth's atmosphere depending on the time of day, the time of year and the height of the measurement [2].

As mentioned previously, the purpose of this analysis is to calculate the variability of the optical parameter denominated refractive index constant C_n^2 relative to Reynolds number.

C_n^2 is the refractive index structure constant and is calculated in the formula (3.9).

The C_n^2 is dependent of the altitude and of the time in the day. The Figure 3.4 shows a variation of the constant C_n in a lapse of 24 hrs. It can be observed from a range of 0.15×10^{-6} that happens where the sun is more brilliant to 1×10^{-6} particularly during the night [10].

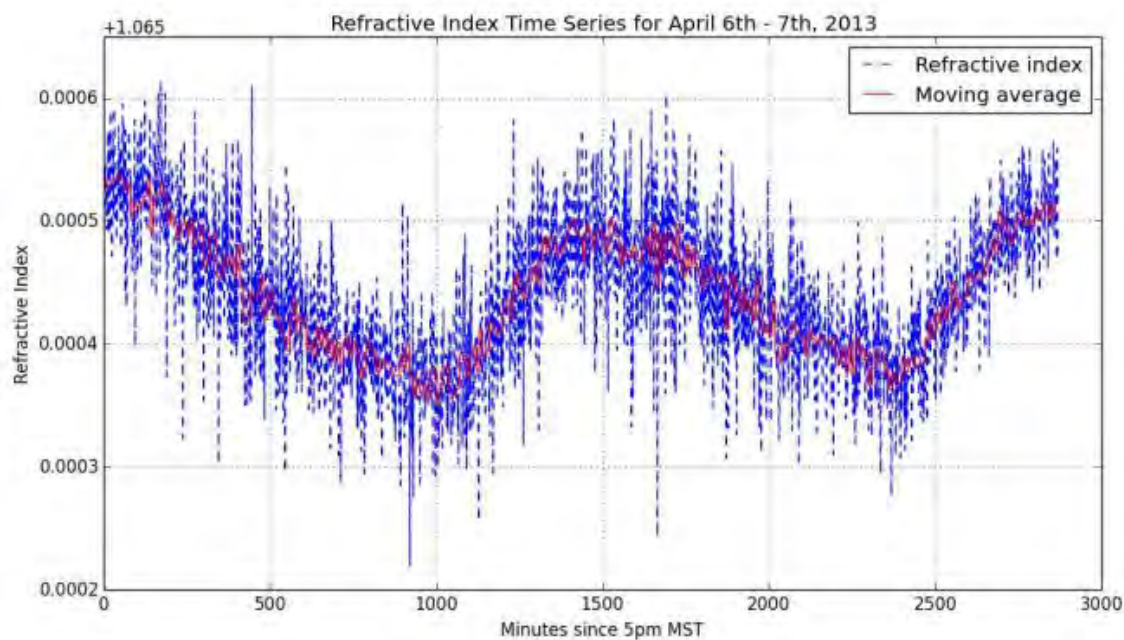


Figure 3.4: Distribution of the C_n value during a typical spring day in Hawaii.

The cycle of C_n during the day, represents a similar behavior on the surface of the earth (in the approximate range between 30 to 100 m).

The fluctuation of temperatures, and therefore of C_n , is very reduced at night and the first hours of the morning. After the sun heats the soil, the process of atmospheric convection starts and, consequently, the temperature variations.

The instability due to convection generates temperature variations [5]. The process by which these temperature variations occurs starts when the materials are heated by the sun. Then the air in contact with these surfaces heats up, loses density, and rises, colliding with the cold layers in its path and this mixture of layers of cold and hot air causes irregularities in the temperature.

The heating process as the atmosphere irregularities continue during the day, which in turn cause an increase on C_n , until the end of the afternoon, when the solar heat falls, consequently the C_n values also decrease.

Typical C_n^2 values on a sunny day at a height of 2 m above the ground, are within a range of $10^{-17} \text{ m}^{-2/3}$ to $10^{-12} \text{ m}^{-2/3}$. Some other variations in the previous figure have to do with sudden presence of clouds, when the solar heat falls, consequently the C_n values also decrease, since the grown quickly cools down and the structure of the temperature in the air becomes uniform rapidly [15].

Chapter 4: Numerical Simulation of Laser Propagation.

The laser can be understood as an electromagnetic wave that suffers distortion in turbulent environment by the rotational phenomenon mentioned previously and that is called eddies. This distortion can be understood based on the theory of geometrical optics. The effect of the eddies in the optical waves, as also mentioned previously, is similar to thin lenses that can modify their waves [12].

During the current work, simulation was done to measure the phase variance of a laser beam that propagates along horizontal paths in a maritime atmosphere between the top of two mountains where two important sites are placed, the Mauna Loa NOAA observatory and the AFRL AEOS telescope on Haleakala in Hawaii. The distance in between them is 150 km and the height of the mountain, where the laser emitter is placed, is 3.1 km. On the other hand, the height of the mountain of the Haleakala, where the receptor is placed, is 3.4 km.

The equations to be modeled and solved through numerical methods have to be created in a geometric structure using a series of planes generated in a Cartesian system, in such a way that the entire laser path, as well as its phase changes can be understood plane by plane. Then the laser path is placed at the z coordinate, the vertical axis is y and the depth is x . The distance in between the mountains is defined over the z axis and the height of the mountains is defined over the y axis [14].

The Figure 4.1 represents the way in which the Cartesian axes were established on the volume under study. It is important to mention that I am including the definition of *observation plane* and *source plane* into the diagram. The source plane is represented by the emission point of the laser beam. The observation plane is placed in certain position along the z axis where I have

decided to perform the analysis of velocity, pressure and temperature as well as refractivity conditions.

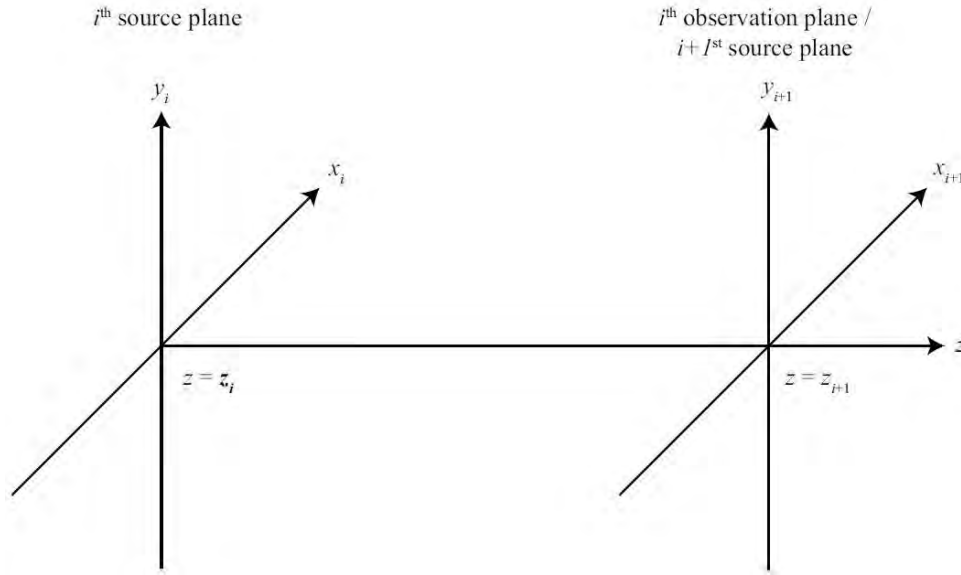


Figure 4.1: Definition of the source plane and observation plan inside a cartesian system

The geometry used in the figure establishes $r_1 = (x_1, y_1)$ the coordinates for the plane of the source and $r_2 = (x_2, y_2)$ for the plane of observation or receiver. The distance between the two planes is Δz . The figure shows the basic problem, given the optical plane of the source field $U(x_1, y_1)$, is necessary to define the field of the observation plane $U(x_2, y_2)$. The solution is found in the fraction integral given by Fresnel in the following equation [13]:

$$U(x_2, y_2) = \frac{e^{ik\Delta z}}{i\lambda\Delta z} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(x_1, y_1) e^{i\frac{k}{2\Delta z}[(x_1-x_2)^2 + (y_1-y_2)^2]} dx_1 dy_1 \quad (4.1)$$

In order to make a better observation of the laser phase behavior along the route, it is necessary to generate a certain number of planes, then the diagram shows an i^{th} observation plan.

In this way, the z axis has been divided by a number of planes to establish points of observation. Then each plane represents partial states of observation that allow the calculation of the atmospheric conditions such as velocity, temperature and pressure, based on previous results to finally calculate the diffractive index structure constant C_n^2 on each case [14].

In addition, a very powerful tool to analyze the behavior of laser's diffraction is through the generation of phase screens. In the previous paragraph, there is a mention that the generation of intermediate planes allows the calculating of C_n^2 , so it is worth mentioning also that it is possible to generate phase screens. The phase screens are 2D maps, as seen of Figure 4.2, that through colors demonstrate the conditions of the laser phase and allow to observe its behavior along the path.

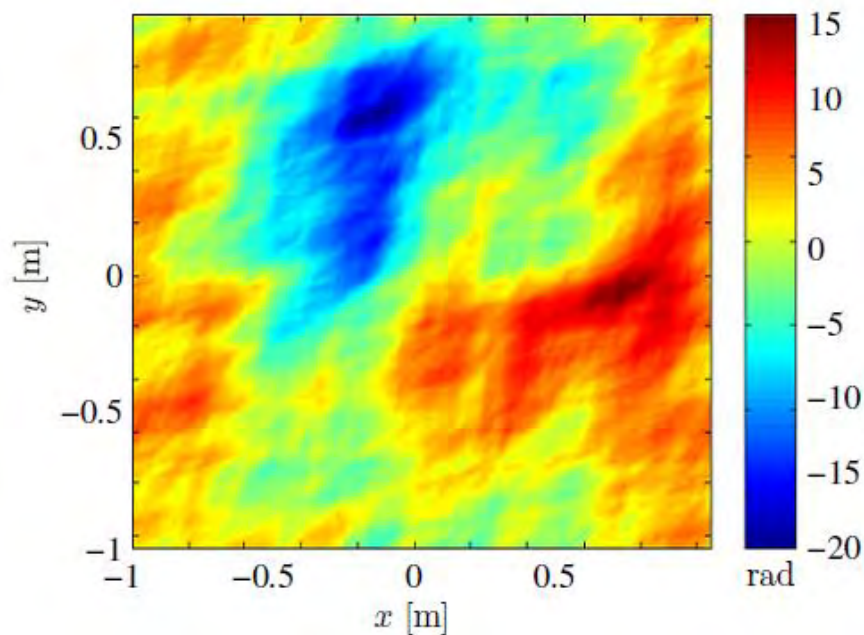


Figure 4.2: Distribution of the laser phase screen in a 2D map.

4.1. Fourier Transforms: from Analytical to Numerical.

In order to solve the Fresnel equation using numerical methods, an important tool is the Fourier transform, since linear analysis and Fourier are essential tools for studying wave optics. Fourier spatial function $g(x)$ and its inverse [25] are shown in the equation (4.2):

$$G(f_x) = \mathcal{F} \{g(x)\} = \int_{-\infty}^{\infty} g(x) e^{-i2\pi f_x x} dx \quad (4.2)$$

The first step to perform the conversion to numerical analysis is the discretization of the Fourier transform into a Riemann sum:

$$G(f_{xm}) = \mathcal{F} \{g(x_n)\} \quad (4.3)$$

$$G(f_x) = \sum_{n=-\infty}^{\infty} g(x_n) e^{-i2\pi f_{xm} x_n} (x_{n+1} - x_n), \quad m = -\infty, \dots, \infty \quad (4.4)$$

where n and m are integers. Computational calculations can only work with a finite number of samples N , so we set a δ interval such that $x_n = n\delta$ [26]. Then, the frequency of the interval is $\delta_f = 1/(N\delta)$ so that $f_{xm} = m\delta_f = m/(N\delta)$.

$$G\left(\frac{m}{N\delta}\right) = \mathcal{F} \{g(n\delta)\} \quad (4.5)$$

$$G\left(\frac{m}{N\delta}\right) = \delta \sum_{n=-N/2}^{N/2-1} g(n\delta) e^{-i2\pi mn/N}, \quad n = -N/2, 1 - N/2, \dots, N/2 - 1 \quad (4.6)$$

The use of positive indices in the rearrangement of samples is similar to what occurs in the Discrete Fourier Transform [27]. The equation (4.7) shows the result:

$$g_{n'} = \frac{1}{N\delta} \sum_{m'=1}^N G_{m'} e^{i2\pi(m'-1)(n'-1)/N}, \quad n' = 1, 2, \dots, N \quad (4.7)$$

The use of the discretization of the optical waves of the laser using the Fourier transform allowed us to apply numerical methods to predict the phase changes of the laser that were experiencing while propagating in turbulent media [28].

Chapter 5: Governing Equations of Fluid Mechanics for Turbulent Conditions.

As commented from the beginning of this work, the medium in which the laser beam is moving is experiencing turbulence due to the conditions of velocity, temperature and pressure.

One of the most accurate ways to understand the fluidic phenomenon is through the equations of Louis Navier and George Gabriel Stokes. These scientists set out to describe the motion of viscous fluids by applying Newton's second law to motion fluids and assuming that the stress in the fluids is the result of a sum of diffuse viscosity and pressure [29].

These numerical relations are called Navier-Stokes equations, which are based on the assumption of an incompressible fluid with ρ and μ uniform and $\nabla \cdot U = 0$ [30]:

$$\rho \frac{DU_j}{Dt} = \mu \frac{\partial^2 U_j}{\partial x_i \partial x_i} - \frac{\partial P}{\partial x_j} - \rho \frac{\partial \Psi}{\partial x_j} \quad (5.1)$$

The Reynolds number, $Re = uL/\nu$, is the unique free parameter if the forces are not present. In the case of our study, where we focus our research on a turbulent atmosphere, the value of the Reynolds number in the 10^6 units.

The typical wind velocities in Maui, where this study is based on, are within the range of 5 to 6 m/s, and since the average height of the mountains of 3250 m, and with a laser dispersion range of 3 m that takes de part of the characteristic, the Reynolds number has range of values from 1.09×10^6 to 3.47×10^6 [31].

Within this range of Reynolds values, fluidic behavior is known as chaotic and for this reason the equations that apply have a particular different effect from the classical theory of

laminar flow where the Navier-Stokes equations have a prediction within a notorious level of accuracy [32].

An essential characteristic of turbulent flow is that is rotational, meaning that it has a certain level of vorticity. Vorticity $\omega(x, t)$ is the curl of the velocity

$$\omega = \nabla \times U \quad (5.2)$$

and is equal to twice the average of the fluid rotation at (x, t) . The equation that defines the evolution of vorticity is obtained by calculating the curl of the Navier-Stokes equations [33]:

$$\frac{D\omega}{Dt} = \nu \nabla^2 \omega + \omega \cdot \nabla U \quad (5.3)$$

The term related to pressure $\left(-\nabla \times \nabla \frac{p}{\rho}\right)$ vanishes due to incompressible flows or of constant density.

The equation that shows the evolution or rotation of an infinitesimal linear element of certain material $s(t)$ is the (5.4):

$$\frac{ds}{dt} = s \cdot \nabla U \quad (5.4)$$

On this equation, without considering the viscous term, it can be seen that is identical to the vorticity equation (5.3) [34]. Therefore, in the inviscid flow, the vorticity vector has the same behavior of an infinitesimal linear element of material, analysis that is known as the Helmholtz theorem [35].

If the average of the strain produced by the velocity gradients acts to contract the linear material element aligned with ω , then the magnitude of ω increases in the same manner. This phenomenon is known as *vortex stretching*, and is a very important process in turbulent flows, and $\omega \cdot \nabla U$ is denoted as the term of vortex stretching [36].

Chapter 6. Fundamentals of Computational Fluid Dynamics.

The simulation of fluids in motion using Computational Fluid Dynamics, is based on the governing equations of fluid flow representing fundamentals of the laws of conservation of physics [37]:

1. Conservation of the fluid mass.

2. The ratio of change of momentum that is equal to the sum of the forces in a fluid particle.

This statement comes from the Newton's Second Law.

3. The rate of change of energy is equal to the sum of the rate of the addition of heat and of the work done on a fluid particle. This statement comes from the first law of thermodynamics.

6.1. Conservation of Mass.

The essential concept of mass conservation is to define mathematical equations to demonstrate that the rate of the mass increase in the fluid of the element is equal to the net change on the mass flow of fluid within the element [38]. The Figure 6.1 shows the mass flow vectors entering and leaving the element volume.

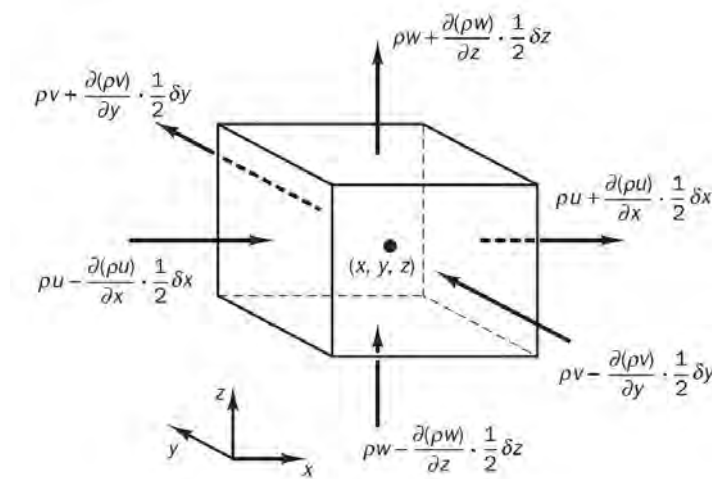


Figure 6.1: Mass flow vectors entering and leaving an element volume.

The rate of mass increase within the element is now equal to the net change in mass flow within the element across its faces. All the terms resulting from the mass balance are arranged in the left hand of the equation. It yields [39]

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (6.1)$$

using compact vector notation

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{u}) = 0 \quad (6.2)$$

Considering that we are assuming incompressible fluid conditions, therefore the final equation of conservation of mass will be

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (6.3)$$

6.2. Conservation of Momentum.

The Newton's second law defines that the rate of change of the momentum of a fluid particle is equal to the sum of the forces on the particle. The rate of change of momentums per unit volume in a fluid particle at x , y , and z is given by [39]

$$\rho \frac{Du}{Dt} \quad (6.4)$$

$$\rho \frac{Dv}{Dt} \quad (6.5)$$

$$\rho \frac{Dw}{Dt} \quad (6.6)$$

Two types of forces on fluid particles can be distinguished: *surface forces* and *body forces*. Within the surface forces are those of pressure, viscosity and gravity, and within the body forces are the centrifugal, Coriolis and electromagnetic forces.

For practical purposes, contributions due to surface forces are considered as part of the general terms of the momentum equations and to include the effects of body forces, source terms are used. The Figure 6.2 shows the components of the surface forces as shear stresses.

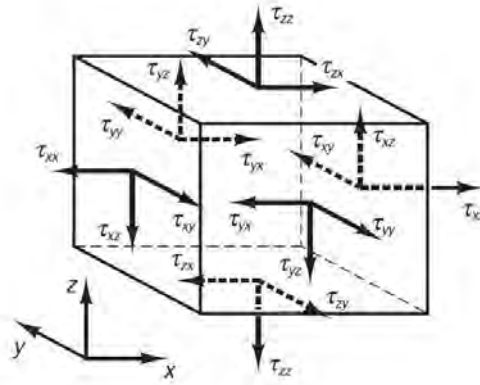


Figure 6.2: Components of the surface forces as shear stresses.

The x, y and z components of the momentum equation respectively are defined as follows

$$\rho \frac{Du}{Dt} = \frac{\partial(-p + \tau_{xx})}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx} \quad (6.7)$$

$$\rho \frac{Dv}{Dt} = \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial(-p + \tau_{yy})}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + S_{My} \quad (6.8)$$

$$\rho \frac{Dw}{Dt} = \frac{\partial(-p + \tau_{xx})}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx} \quad (6.9)$$

6.3. Conservation of Energy.

The energy equation is derived from the first law of thermodynamics, which states that the rate of change of the energy of a fluid particle is equal to the ratio of the heat added to the particle plus the ratio of the work done on the particle [40].

$$\rho \frac{DE}{Dt} \quad (6.10)$$

6.3.1. Work.

The ratio of work done on the fluid particle by a surface force is equal to the product of the force and the velocity component in the direction of the force. The ratio of the total work done on a fluid particle by a surface force is as follows [40]:

$$\begin{aligned} [-div(pu)] + \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} \\ + \frac{\partial(v\tau_{zy})}{\partial z} + \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z} \end{aligned} \quad (6.11)$$

6.3.2. Heat conduction.

The net heat flux in a fluid particle in any of the three x, y, and z directions is given by the difference between the ratio of the input heat flux to the output heat flux on each respective Cartesian axis. The Figure 6.3 shows the heat flow in each of the geometric axes [41].

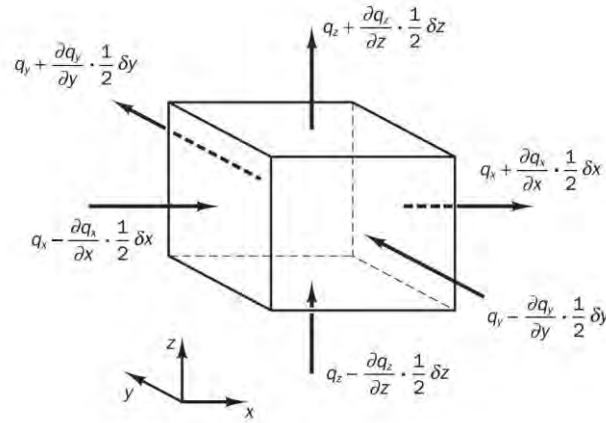


Figure 6.3: Mass flow vectors entering and leaving an element volume.

The following equations show the Fourier's law of heat conduction taking as a reference the local temperature gradient.

$$q_x = -k \frac{\partial T}{\partial x} \quad (6.12)$$

$$q_y = -k \frac{\partial T}{\partial y} \quad (6.13)$$

$$q_z = -k \frac{\partial T}{\partial z} \quad (6.14)$$

These equations can be written in vector form as follows

$$\mathbf{q} = -k \text{ grad } T \quad (6.15)$$

6.3.3. General Energy Equation.

Considering the general sum of body and surface forces, the general equation for energy balance is as follows [41]:

$$\begin{aligned} \frac{\partial(\rho h_0)}{\partial t} + \text{div}(\rho h_0 \mathbf{u}) = & \text{div}(k \text{ grad } T) + \frac{\partial p}{\partial t} + \left[\frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} \right. \\ & \left. + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} + \frac{\partial(v\tau_{zy})}{\partial z} + \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z} \right] + S_h \end{aligned} \quad (6.16)$$

6.3.4. General Equilibrium Equation.

All the effects caused by viscous stress in the internal energy equation are described by the dissipation function, which can be described using the Newtonian model for the viscous effects in the equation of internal energy. It can be described as follows [42]:

$$\rho \frac{Di}{Dt} = -p \text{ div } \mathbf{u} + \text{div}(k \text{ grad } T) + \Phi + S_i \quad (6.17)$$

Chapter 7: CFD Simulation.

7.1. Model Design.

To create the model, it was necessary to limit the original geometry. Because the height of both mountains is around 3 km, we decided to double the height to 6 km to cover a more complete volume. The dimensions were 6 km height, 2 km width and 150 km of length.

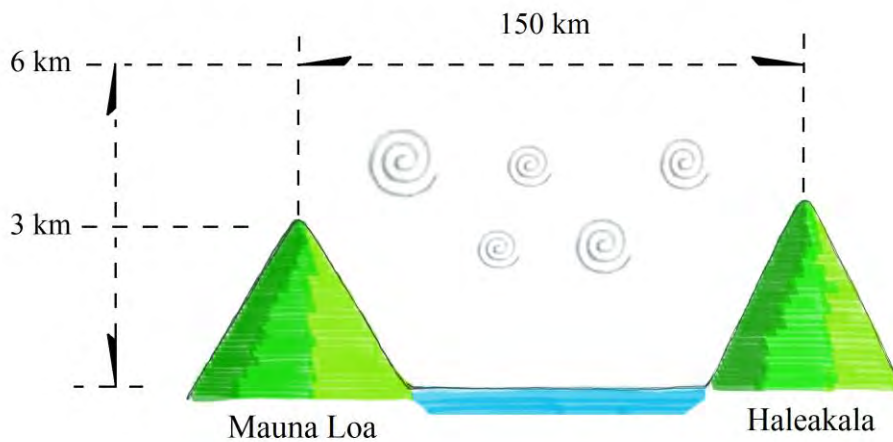


Figure 7.1: General geometry under analysis.

7.1.1. Scaling the Model.

The initial 3D model was created in a software named NX. Due to the large geometry of the real model where the length is 150 km, to fit the geometry into a 3D computational model, it was decided to perform a scaling down of 10,000: 1. The Figure 7.2 shows the dimensions for the scaling down step.

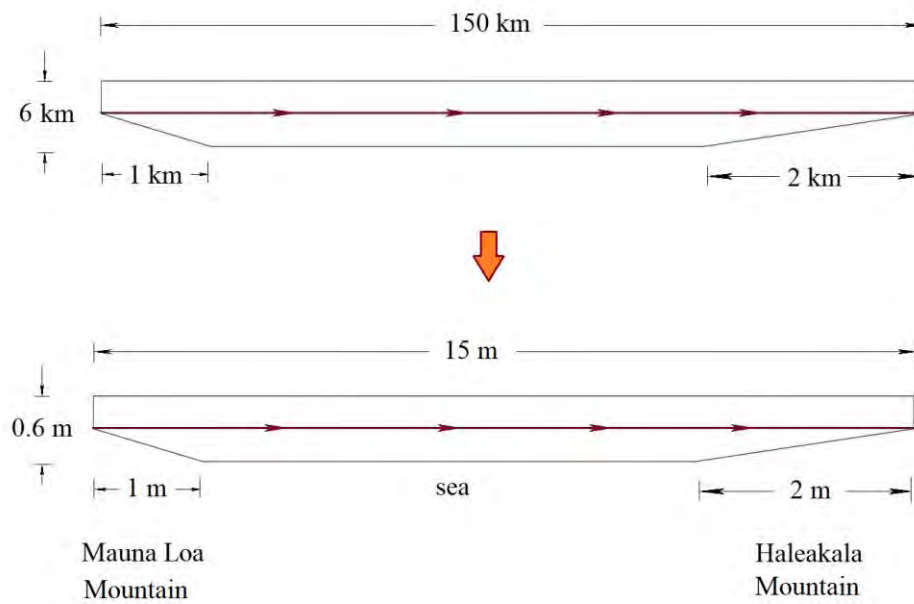


Figure 7.2: Scaling down the geometry.

The Figure 7.3 shows the final dimensions of the model. The length was trimmed so the lateral dimensions were clearly observed.

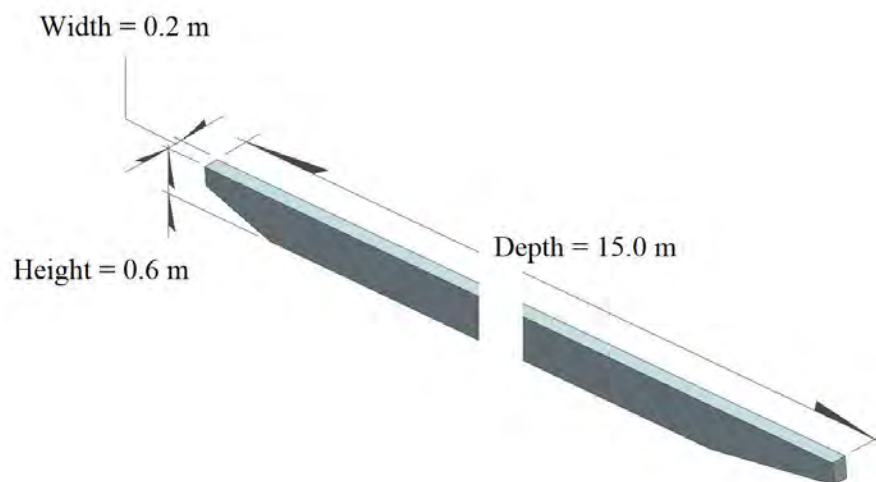


Figure 7.3: Final Geometry dimensions.

7.1.2. Justification of Modeling Scaling Down.

For the model to be analyzed using the CFD ANSYS Fluent simulation software, a scaling of 10,000: 1 was necessary.

The scaling dimensional factor was calculated based on the original length and the proposed length of the model

$$\frac{L_o}{L_s} = \frac{150,000 \text{ m}}{15 \text{ m}} = 10,000 \quad (7.1)$$

L_o is the original length and L_s is the length in the scaled model. Both of the dimensions are in meters.

7.1.2.1. Velocity conditions

Following these scaling conditions, the next step was to define the values of the properties to be used in the scaled model, so I started with the velocity.

Based on the range of 5 to 16 m/s of the original model I performed the scaling down of the velocity of 5 m/s.

$$v_s = \frac{v_o}{10,000} = 5 \times 10^{-4} \text{ m/s} = 0.5 \text{ mm/s} \quad (7.2)$$

However, because the appropriate conditions for turbulence are the same in the original and in the scaled model regardless of the dimensions, scaling down velocities 10,000 times does not produce the turbulence conditions required for my analysis. Not even scaling down the

velocities 10 times. If the turbulent conditions cannot be reproduced, the effects over the diffraction index cannot be calculated.

For this reason, I decided to maintain the same velocities that generate the same turbulent conditions of the original geometry even though the volume of fluid as well as the inlet area have decreased 10,000 times.

7.1.2.2. Temperature conditions

The basis for performing the analysis of the changes in the constant of diffraction index C_n^2 are the temperature gradients. In a 10,000:1 scaled model there is virtually no temperature change between the bottom and the top of the model, not even one-degree celsius.

The temperature in the original model varies from 25 C at sea level to 5.5 C degrees at the lasers path. For this reason, it was decided to repeat this linear temperature gradient in the scaled model in order to simulate the temperature variations affected by the turbulence conditions.

7.1.2.3. Pressure conditions

Because pressure conditions require to be linked to temperature to maintain adequate density conditions, it was also decided to set the original manometric pressures over all the model.

7.1.2.4. Testing

I performed numerous tests, more than 1000 CFD Simulations, to verify that the boundary conditions explained on this scaling justification are the closest possible to simulate the original model conditions.

The Figures 7.4 and 7.5 show the comparative simulation of the velocity in the case of scaling down the temperature and pressure conditions vs maintaining the original ones. The low precision presented by the case of the scaled conditions can be clearly observed with respect to the original ones.

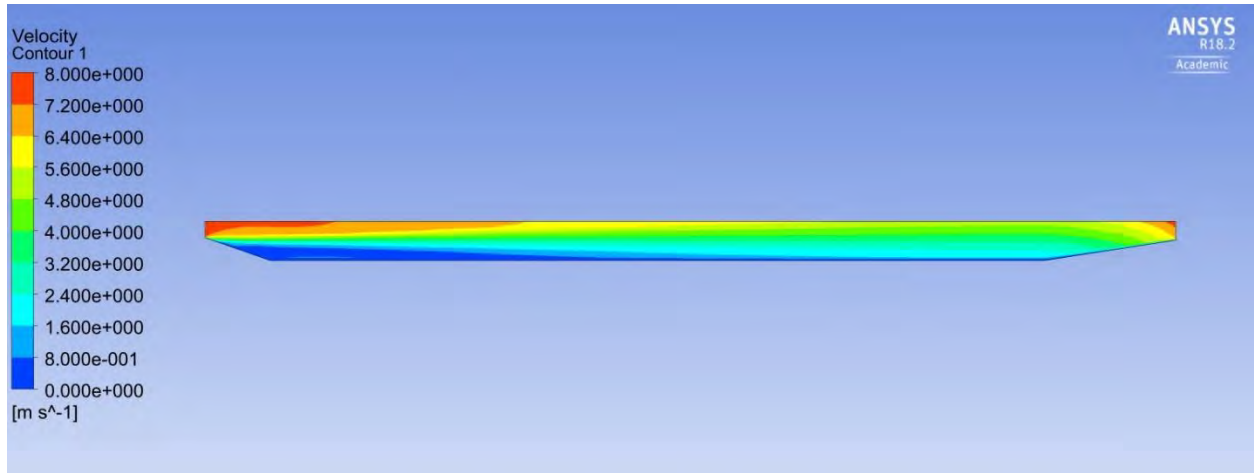


Figure 7.4. Velocity simulated with scaled conditions.

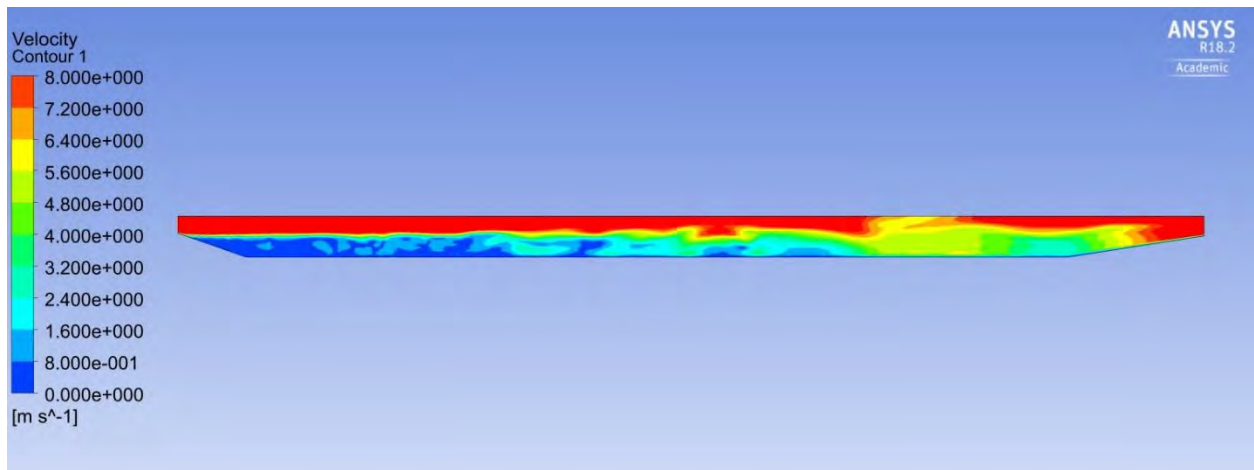


Figure 7.5. Velocity simulated with original conditions.

Additionally, Figures 7.6 and 7.7 show the behavior of the temperature with scaled conditions vs original one. There is practically no profile of temperature variations in the case of the scaled model, but the profile for the case of temperatures in the original model is very clear.

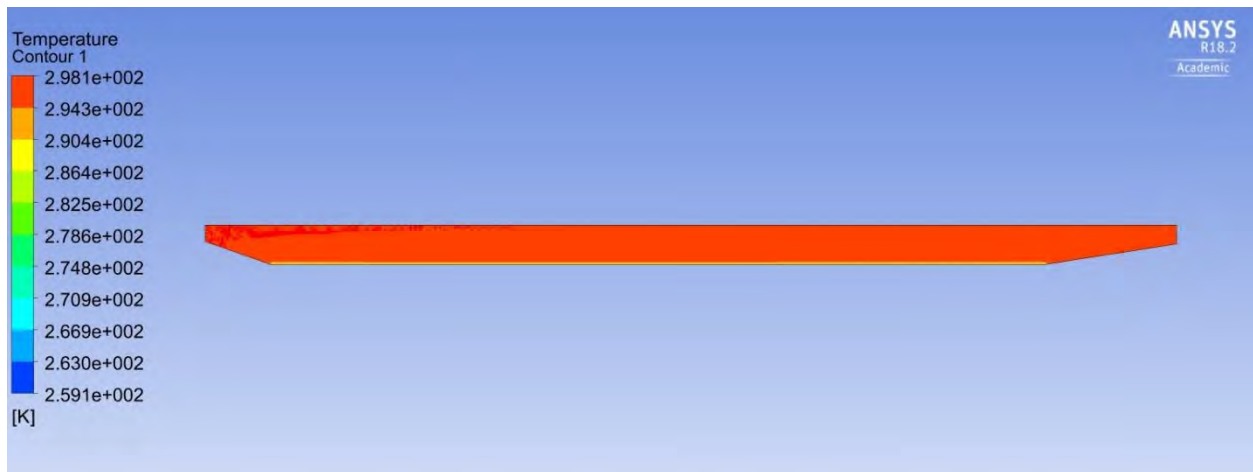


Figure 7.6. Temperature variations simulated with scaled conditions.

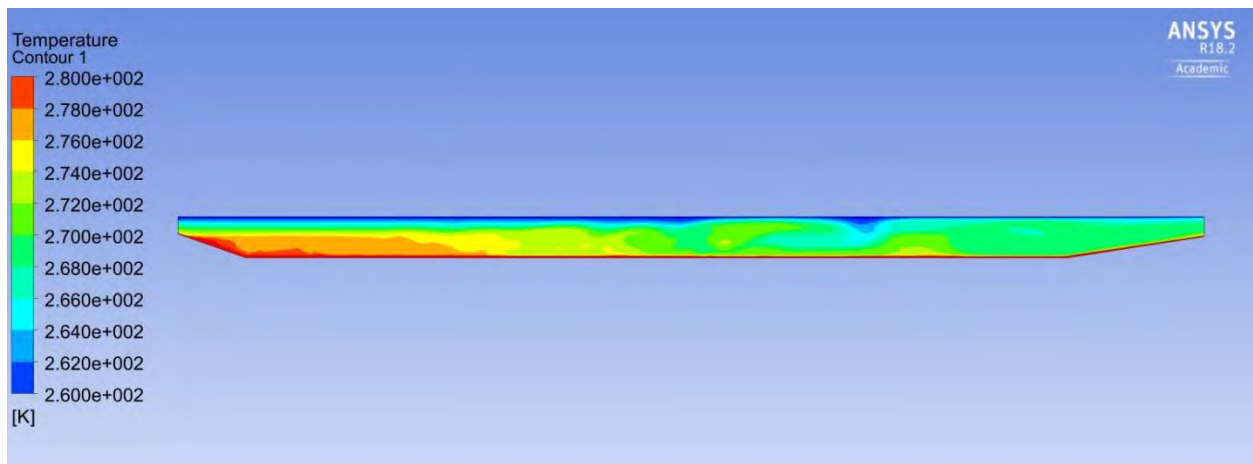


Figure 7.7. Temperature conditions simulated with original conditions.

These series of 1000 simulations allowed me to define that effectively, in the case at hand, the use of the velocity, temperature and pressure conditions of the original geometry, it is the best option to simulate the turbulence conditions necessary to necessary to obtain the adequate information for my study.

This allows me to conclude that the simulation that was carried out during the present work has an acceptable degree of precision.

7.1.2. 3D Modeling

The CFD simulation software that was used to obtain the temperature, pressure and velocity profile was the ANSYS Fluent [44]. This tool has several sections that are shown in the Figure 7.8 and will be explained below.

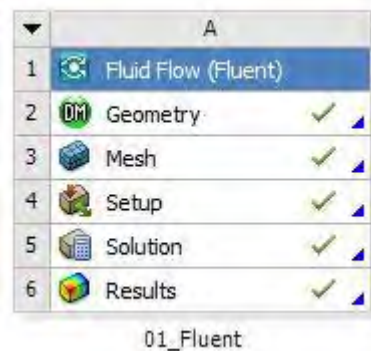


Figure 7.8: ANSYS Fluent Sections for fluid analysis.

In the *Geometry* section the tool can generate the geometric profile that needs to be analyzed or imported in general *.iges* format from any other design software. In my case I used NX to generate the geometry and import it to ANSYS through its module called *DesignModeler*. The Figure 7.9 shows the geometric profile of the case.

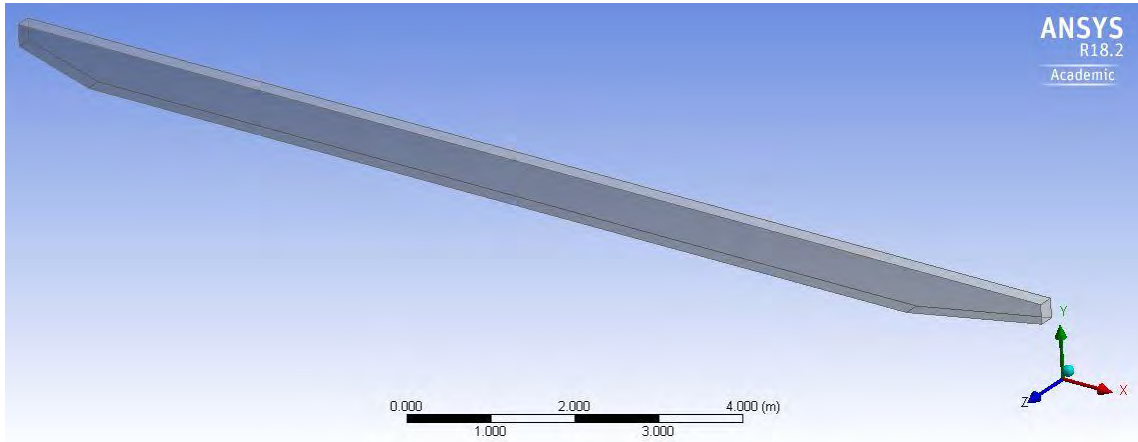


Figure 7.9: 3D ANSYS solid model of the case under analysis.

7.2. Meshing.

The section named *Mesh* is where each part of the model is defined; after this, the sensitive mesh can be generated. The mesh allows the generation of the general simulation. In the Figure 7.10 the named sections can be seen inside the design tree. In the Figure 7.11 the named sections are shown on the solid model [44].

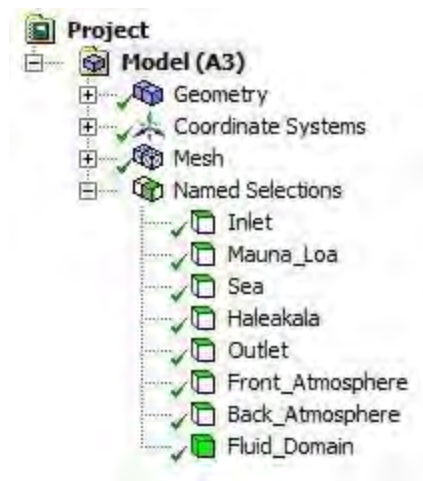


Figure 7.10: Design tree for the ANSYS Meshing section.

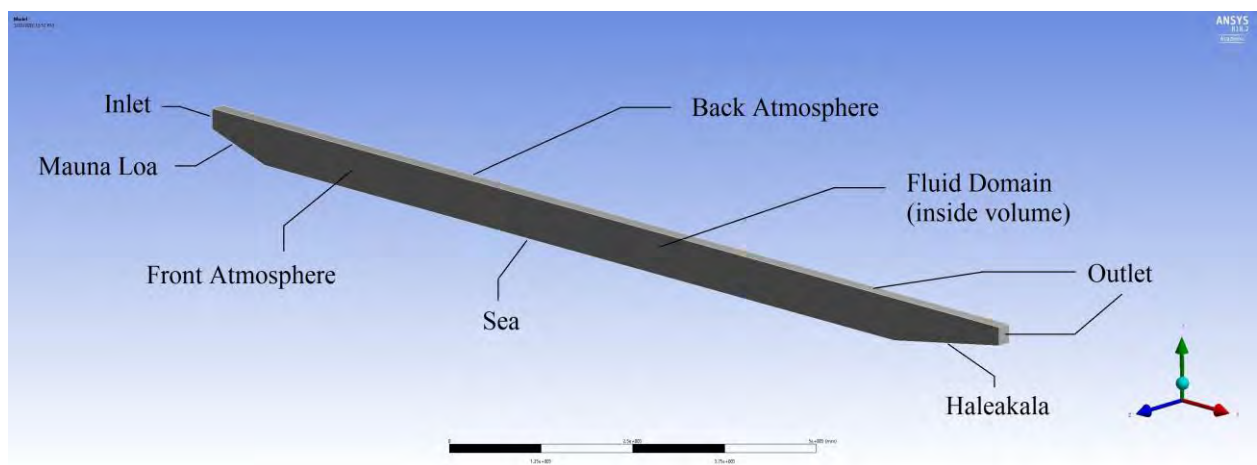


Figure 7.11: Designation of named sections according to the original model.

After the corresponding sections have been named, the mesh size is established through the command called *edge sizing*. In this case an edge size of 20 mm was defined. The figure shows the edges of the model selected with a division of 20 mm.

The reason I selected a 20 mm for the elements dimension is because it was the smallest possible dimension to reach the maximum limit of 500,000 element in the ANSYS Fluent because I used a student license. This selection on Figure 7.12 precedes to the generation of the mesh.

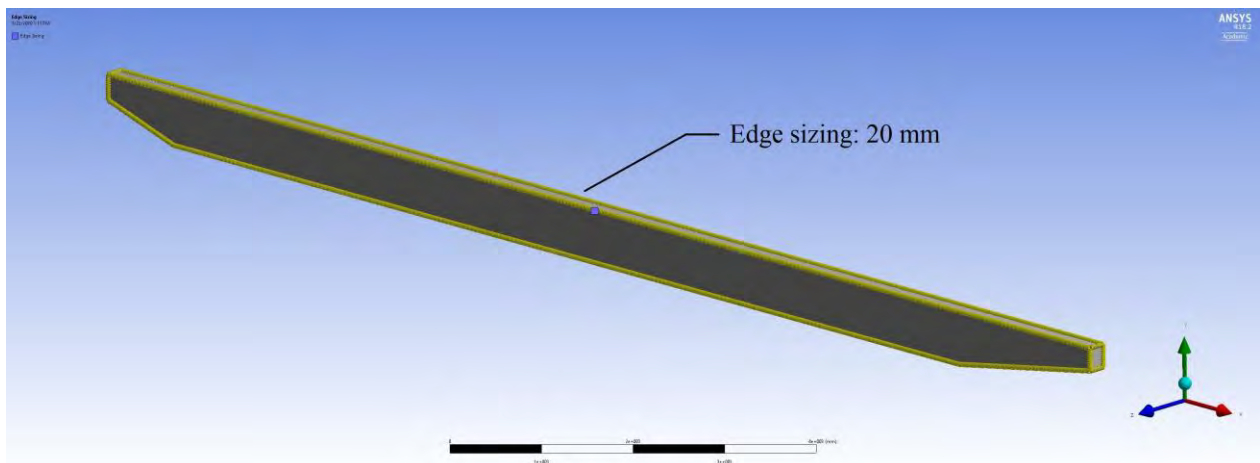


Figure 7.12: Edge dimensioning for defining the element size of the mesh.

In the Figure 7.12 the final mesh can be seen. It has 450,000 tetrahedral elements and 550,000 nodes. This mesh represents components of a matrix that will receive the values of the fluid properties using the fundamental fluid equations.

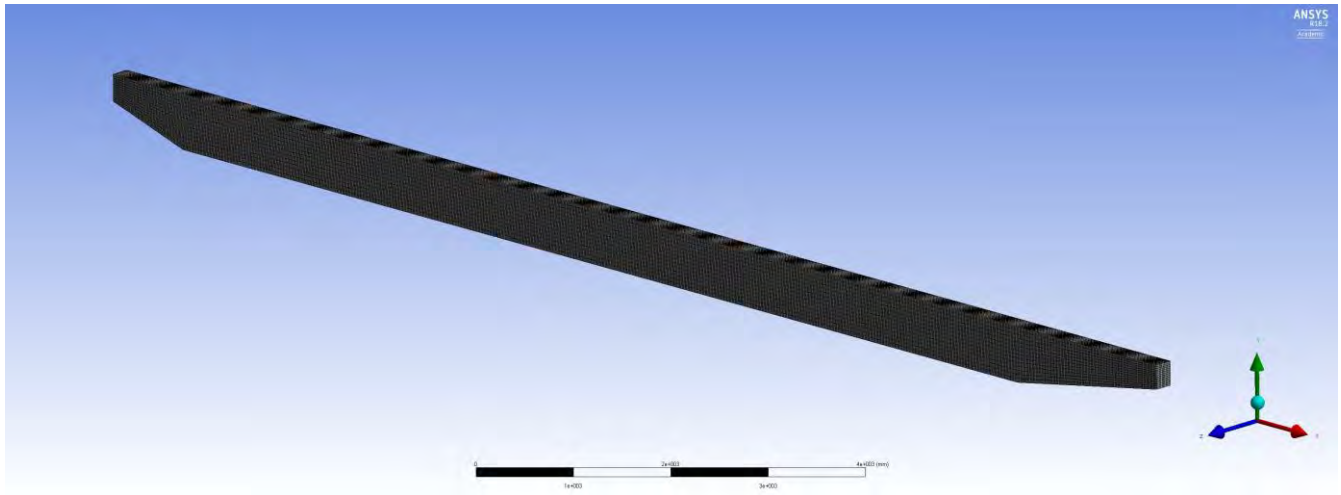


Figure 7.13: 3D mesh with 20 mm side elements.

7.3. Defining Boundary Conditions and Setting Up the Model.

In the *Set up* section all the boundary conditions are defined. These data are fed on the three divisions shown in the Figure 7.14.

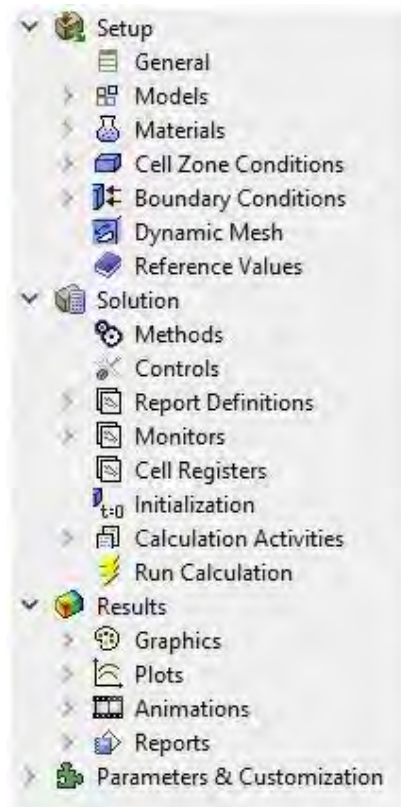


Figure 7.14: Components of the *Set up* section in ANSYS Fluent.

In the *Models* section of the *General* modules, the type of analysis that will be performed is defined. In my case, since it involves heat flow, it is necessary to indicate that it will be using in the Energy's mathematical modules [43]. On the other hand, the mathematical model that I considered most suitable for its precision when simulating turbulent conditions is the LES (Large Eddie Simulations). I selected this option inside the *Viscous Model* section.

In the *Materials* section I specify *air* as the working fluid. In the *Cell Zone Conditions* the pressure and temperature conditions were specified according to the range of values that are defined in the *Chapter 9. Problem Statement* of this dissertation.

Finally, the model is initialized and run with all the data supplied [43]. At the end, a data profile with the velocity, temperature and pressure by coordinate can be generated in a .csv file in the *Solution* section [45]. The Figure 7.15 shows the generated profile of the wind velocities and the Figure 7.16 shows the profile of atmospheric temperatures.

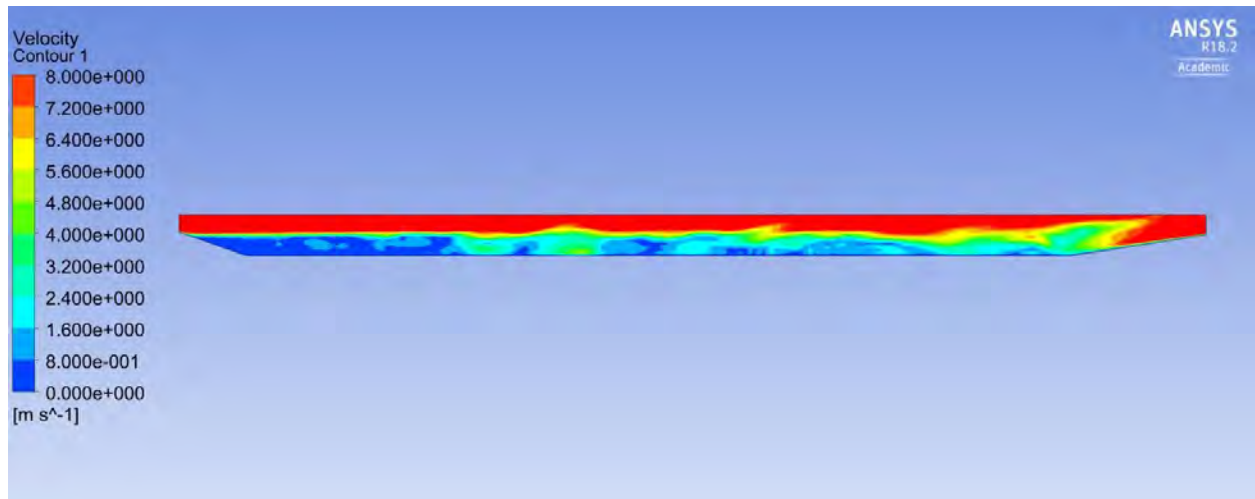


Figure 7.15: Results of the velocity distribution of a 8 m/s inlet analysis.

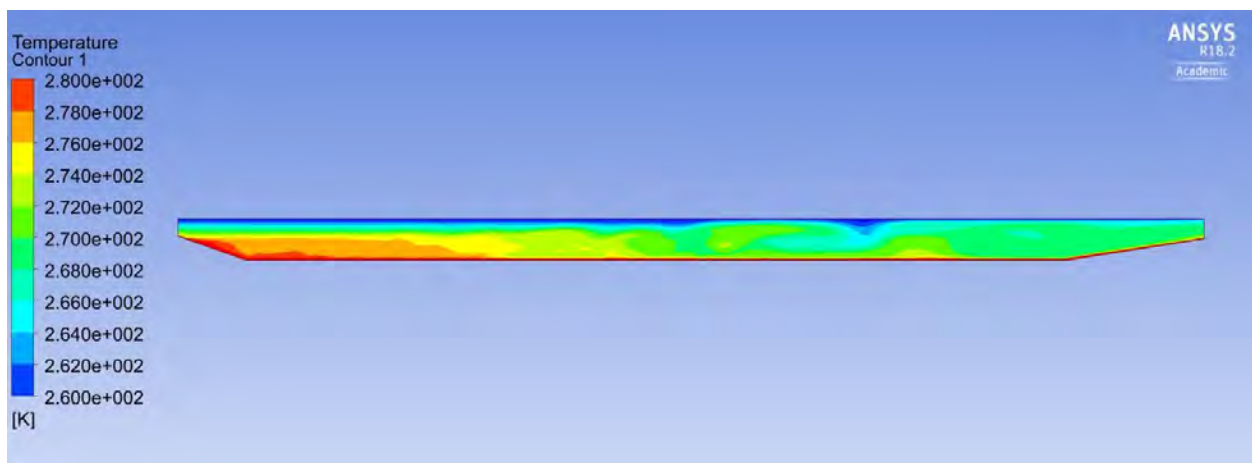


Figure 7.16: Results of the temperature distribution of a 8 m/s inlet analysis.

Chapter 8: Deep Learning Analysis.

Much of the current communication technology is based on machine-learning analysis. From systems used to identify objects in images, store historic searches to make later recommendations to users, filtering social networks, as well as scientific applications to predict physical phenomena [46].

Deep Learning can be defined as a neural network with more than two layers. A neural network is a set of algorithms that is named this way, because it behaves like how human neurons learn, meaning, through repetitive approaches relating certain experiences to certain results. They are designed to recognize patterns [47]. As neurons do, these algorithms learn from data to relate initial conditions to final responses identifying the pattern. At the end, the network delivers the outputs based on the inputs and the pattern that previously trained the network.

The most common deep learning analysis is the one known as supervised learning, which in every set of data that is fed into the system, it compares the results to verify how accurate the model is predicting the outputs and then it goes into the next layer [47]. The Figure 8.1 demonstrates the model:

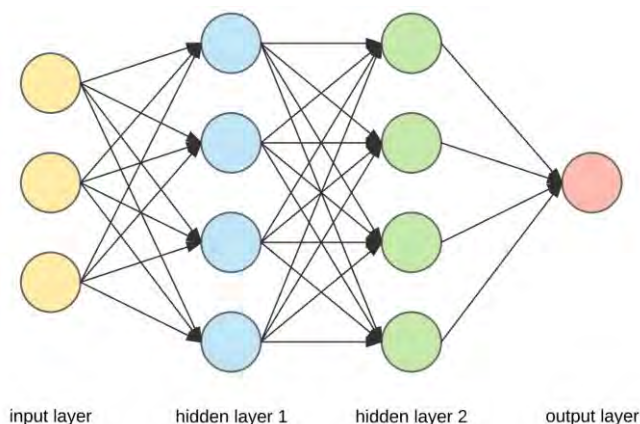


Figure 8.1: Graphic definition of the neural network algorithm.

As can be seen in Figure 8.1, the neural network is made up of a variety of nodes that are connected and communicated with each other. Each node takes multiple inputs and makes dot products with each weight as observed in the equation (8.1). Then it applies an activation function to the product of the inputs and generates an output [61].

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + [b] = x_1w_1 + x_2w_2 + x_3w_3 + b \quad (8.1)$$

This product is then used as input for the activation function. The activation function acts as a switch and depending on the function, the value can be between 0 and 1 or -1 and 1. It is a common practice that for the activation function, any of the following is used: Sigmoid or ReLu (Rectified Linear Unit) [61].

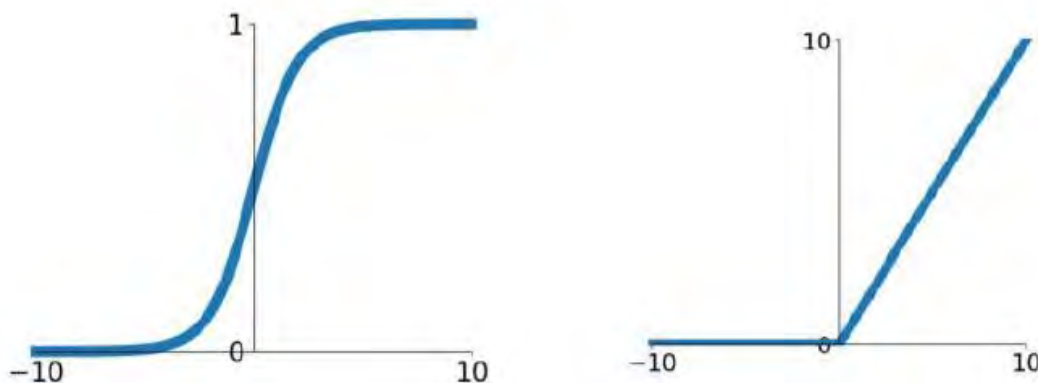


Figure 8.2: Sigmoid Function (Left), ReLu Function (Right)

Once the activation function is calculated, the output denoted by h is then determined.

$$h_1 = f(x_1w_1 + x_2w_2 + x_3w_3 + b) \quad (8.2)$$

The activation function is denoted by f . In neural networks there are two components: the linear component that is the linear algebra and the non-linear component that is the activation function [61].

The hidden layers are giving weights to the different inputs according to the previous training of neural network. Based on this weighing, the algorithm is filtering the data to get the best result possible [48].

$$h = f \left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right) \quad (8.3)$$

A certain number of hidden layers help to reduce the size of the error; nevertheless, is important to identify and optimum number of hidden layers and number of internal units because it is possible to increase the error if we continue increasing the number of hidden layers infinitely.

Since the beginning of the pattern recognition, the goal of the researchers has been to replace features designed by the engineer's hand with multilayered networks, but despite its simplicity, this concept was not understood until the mid-1980s. Multilayered architectures can be trained by a simple descending stochastic gradient [49].

The backpropagation procedure to calculate the gradient of an objective function with respect to the weights of a group of multilayered modules is nothing more than a practical application of the chain rule used to calculate derivatives. The key point is that the derivative or gradient of the objective function with respect to the input of a module can be calculated by working back from the gradient with respect to the output of the module (or the input of the subsequent module) [50].

The process of the equation (8.3) is a matrix multiplication followed by an activation function; this will solve the hidden layer and the process will be repeated to determine the adequate weights for the problem. At the beginning of constructing the neural network, the weights are initialized through a Gaussian random number generator. At the start of the construction of a neural

network, the weights are initialized using a Gaussian random generator. The goal is to adjust the neural network weights to get better predictions every time. This is accomplished by varying the weights to minimize the error between the input and the predicted data. To achieve this, it is needed to implement a backpropagation descending gradient.

$$w = w_o - \alpha * \Delta error \quad (8.4)$$

The descending gradient method is based on the gradient that makes a change to the new weight value w . It takes the current position w_o and adjust it with the error weighted by a step size α . The crucial component in a neural network is the loss function. Optimizing the algorithm is based on minimizing this loss function. This type of algorithm is called of *supervised training*, so the algorithm is intended to learn from a series of data to make predictions.

For a given training pair, the input data will be called x^z and the desired output or prediction will be y^z . Thus, J is the loss for the training of that pair x^z .

$$J(w, b, x, y) = \frac{1}{2} \|y^z - y_{pred}(x^z)\|^2 \quad (8.5)$$

The variable $y_{pred}(x^z)$ is the output layer of the neural network in. To generalize the cost function and include more than one training pair, it is necessary to minimize the function for all m training pairs:

$$J(w, b) = \frac{1}{m} \sum_{z=0}^m \frac{1}{2} \|y^z - y_{pred}(x^z)\|^2 \quad (8.6)$$

To determine the new weights the descending gradient and bias for each weight will be calculated through the equations (8.7) and (8.8):

$$w_{ij}^{(l)*} = w_{ij}^{(l)} - \alpha \frac{\partial}{\partial w_{ij}^{(l)}} J(w, b) \quad (8.7)$$

$$b_i^{(l)*} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(w, b) \quad (8.8)$$

The way neural networks work is through an initialization of the weights. Then, the information is feedforward to calculate the predicted value. At the end, the predicted value and the fed value can be taken to determine the loss function. Through the backpropagation and the descending gradient, weights and bias can be adjusted [61].

From the above calculations, the importance of the activation function can be appreciated. Whenever the output of the activation function is close to zero, there will be no information to adjust the weights when using the descending gradient. This is how neural networks learn and adjust to exclude unnecessary values. When the weights are not adjusted, we realize that our numerical network has converged.

In the case at hand, this type of multilayered architecture was used to predict parameters located in cartesian coordinates x, y and z but with no equidistant spaces, so an equidistant model was designed and the neural network algorithm was trained to predict what would be the values of that same geometrical model would but on an equidistant set of cartesian coordinates [51].

8.1. Neural Network Analysis.

8.1.1. Mathematical Background.

In the present study, the first part has consisted in generating data of the profile of atmospheric temperatures, pressures, and velocities. Subsequently, to calculate the structural constant of the refractive index, the following equation is required [52]:

$$D_T = \langle [T_r(r_1 + r) - T_r(r_1)]^2 \rangle \quad (8.4)$$

After the structural function of the temperature has been calculated, the equation 3.5 is used to obtain C_T^2 . This equation was previously defined in Chapter 3:

$$D_T = C_T^2 r^{2/3} \quad (3.5)$$

Then, clearing C_T^2 from the equation I have

$$C_T^2 = D_T / r^{2/3} \quad (8.5)$$

Finally, with the set of values obtained from the structural constant of the temperature C_T^2 in addition to the values of Pressure and Temperature at the height of the laser's path, I use the equation 3.9 to obtain C_n^2 [53]:

$$C_n^2 = \left(\frac{79 P}{T^2} \times 10^{-6} \right)^2 C_T^2 \quad (3.9)$$

Thus, through this process I obtain the value of the structural diffraction constant C_n^2 .

On the other hand, the reason I relied on the use of neural networks was to predict the temperature on equidistant coordinates to be able to calculate the structural function of temperature. It can be observed from the equation (8.4), that the quadratic difference requires a constant value of positional vector r . For the equation to have accuracy, the data has to be placed on an equidistant mesh.

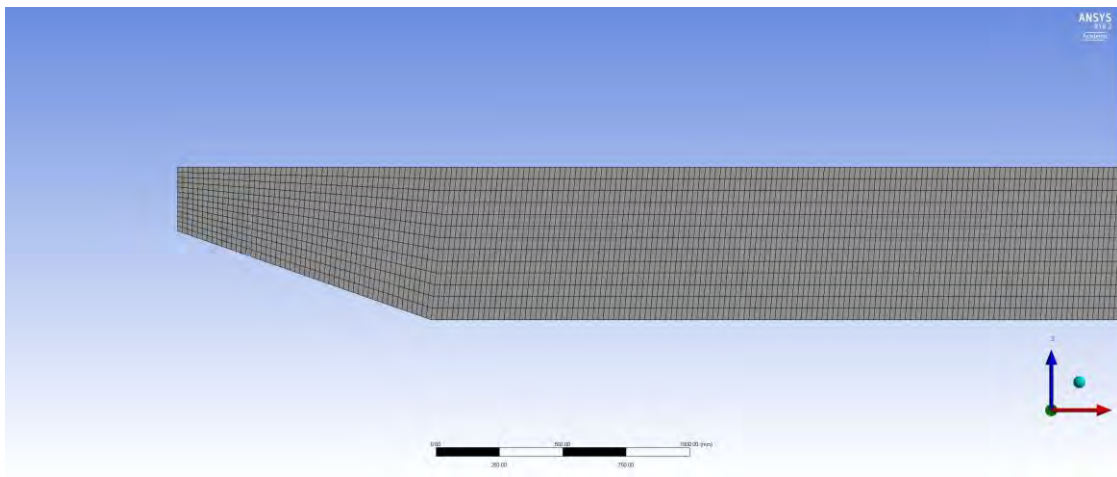


Figure 8.3: Close view of the non-equidistant mesh generated in ANSYS Fluent.

The Figure 8.3 shows the detail of the mesh. It is possible to observe notorious variations in the element sizes which side specification is 20 mm.

8.2. Neural Network Process.

8.2.1. Importing Data.

The procedure to transform the non-equidistant mesh to an equidistant set of coordinate values is using a neural network algorithm. Thus, that step after running the cases from the ANSYS Fluent, was to export the data of velocity, temperature, pressure of the entire volume in .csv format. This data was then imported into a script in Python language.

Inside the Python code, that was created in a Spyder platform, the imported .csv file was used to create a matrix of data. Parallel in the script, an equidistant array of coordinates with side 20 mm was created. Figure 8.4 shows the matrix.

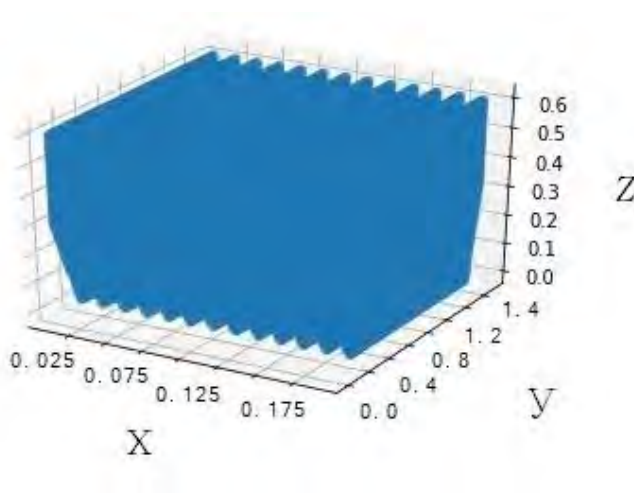


Figure 8.4: Equidistant points created to simulate the original model.

In the figure 8.4, each coordinate is represented by a blue sphere, the union of all these spheres is the diagram presented.

8.2.2. Training the Algorithm

The next step is training the neural network algorithm. The algorithm used was the *MLPRegressor* that belongs to the *sklearn.neural_network* library. *MLPRegressor* means *Multi-Layer Perceptron Regresor* and is a model that optimizes the squared-loss using stochastic gradient descent [52].

8.2.2.1. Selecting the Data.

The process for prediction begins with the selection of data from the .csv file imported from ANSYS Fluent to create two groups of variables (X , Y) selected for training. One of them, the X , containing the set of coordinates and the other one corresponding to the temperature. This process is shown in equations (8.6) and (8.7).

$$X = df.loc[:, ['x - coordinate', 'y - coordinate', 'z - coordinate']] \quad (8.6)$$

$$Y = df.loc[:, ['temperature']] \quad (8.7)$$

8.2.2.2. Training.

Later comes the training section. Based on the set of values X and relating it to Y results, the training comes with a set of values over the 80% of the training data and testing this prediction with 20% of the data from the same set of values. Equation (8.8) shows this process.

$$\begin{aligned} X_{train}, X_{test}, Y_{train}, Y_{test} &= train_test_split(X, Y, random_state \\ &= 42, test_size = 0.20) \end{aligned} \quad (8.8)$$

8.2.3. Prediction Process.

The next step is to use the neural network algorithm to make the prediction. This algorithm is the *MLPRegressor* and its commands can be seen in equations (8.9) and (8.10). The algorithm uses the solver *adam* and the activation function is *relu*. In the commands shown, **6** hidden layers were used as well as **13** neurons inside each hidden layer,

$$Y_clf = MLPRegressor(hidden_layer_sizes = (13, 13, 13, 13, 13, 13), \quad (8.9)$$

$$solver = 'adam', activation = 'relu')$$

$$Y_predict = Y_clf.predict(X_test) \quad (8.10)$$

8.2.4. Optimization

The number of hidden layers as well as the number of neurons per hidden layer mentioned in section 8.2.3, were carefully chosen through an optimization process. This process was carried out through calculating the value of the error in the prediction.

The error in the prediction was calculated by the difference between the predicted and the initial values previously fed into the algorithm. This process were done using different cases of number of hidden layers vs number of neurons by varying both concepts from 2 to 20, respectively [50]. This difference is shown in equation (8.11).

$$err_sum = Y_predict - Y_test_flat \quad (8.11)$$

To avoid negative numbers, the error was squared as shown in the equation (8.12)

$$err_sum = err_sum ** 2 \quad (8.12)$$

The minimum value of error was found to be **6** hidden layers with a size of **13** elements each layer.

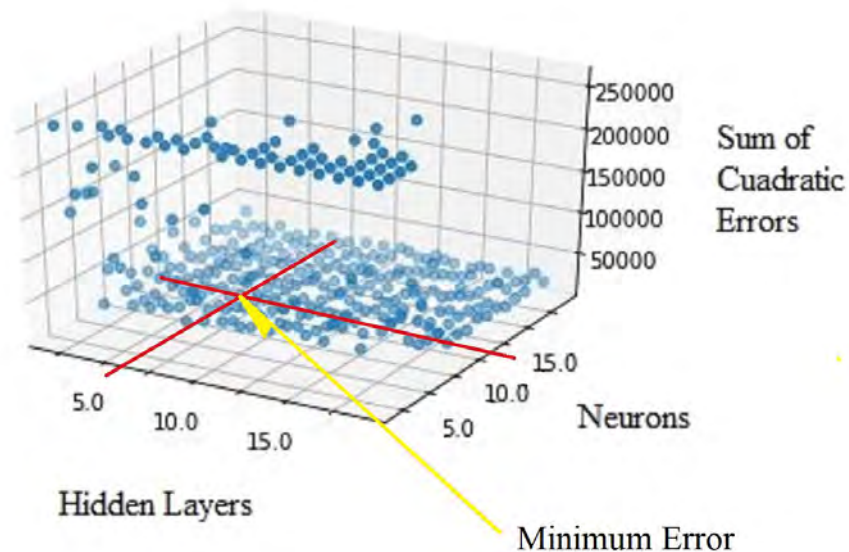


Figure 8.5: Distribution of the errors Hidden Layers vs Neurons.

The Figure 8.5 shows the distribution of the errors when the number of hidden layers is varied with respect to the number of elements per layer. It can be seen that errors are very high when the number of hidden layers is varied but the number of elements per layer is kept low like from 2 to 5 elements, especially if the number of layers is high and the number of elements is low [50].

8.3 Mesh comparison

Although the resultant mesh will be presented with all the results in Chapter 10, Figure 8.6 shows the objective pursued with the use of the neural algorithm.

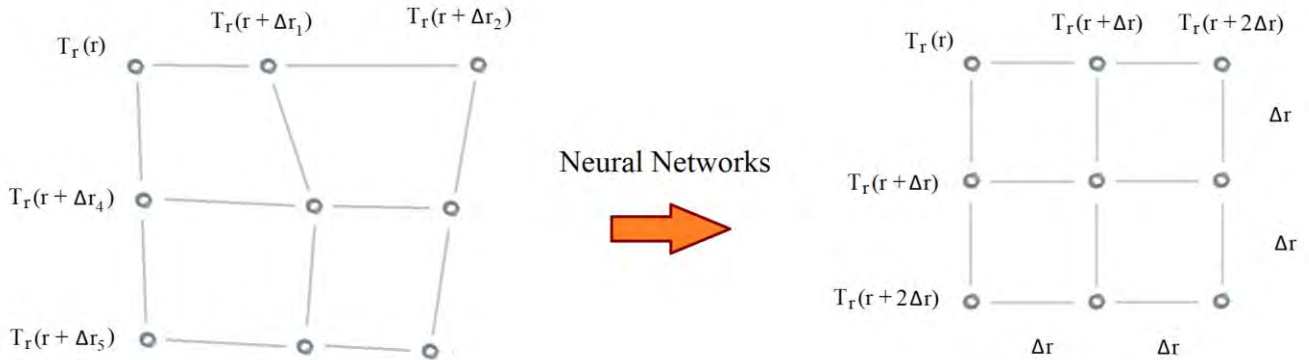


Figure 8.6: Mesh comparison of meshes before and after Neural Network algorithm.

Chapter 9: Problem Statement.

The problem statement is based on the fact that in the presence of strong turbulence in the atmosphere, a laser's phase propagating in this medium, is affected and consequently the quality of the remote sensing and the imaging systems based on it.

The present study is focused on finding new correlations between turbulent atmospheric conditions like velocity, temperature and pressure, and optical variables such as C_n^2 [54], this in order to improve the quality of the laser detected in the receiver system.

The motivation of this study is based and inspired by preliminary work as well as a project supported by the Air Force Office of Scientific Research (AFOSR) by the hypothesis that the theories developed to analyze turbulent fluid dynamics problems with high Reynolds turbulence, will help to improve characterization theories in developing new techniques to understand the effects of deep turbulence on long distance laser propagation.



Figure 9.1. Laboratory at the top of the mountain Mauna Loa.

The goal has been to develop a clear understanding of the problem and to optimize methods of mitigating the effects of strong turbulence on optical systems through a critical examination of the atmospheric variables related to their effects on optical variables.

The atmosphere's refractive index randomly evolves over space and time. This effect causes light to be randomly distorted as it propagates. As a result, optical systems that rely on light propagating through the atmosphere must overcome a great challenge. Figure 9.2 shows typical systems that use optical communication.



Figure 9.2. Systems that use optical communication.

9.1. Objective.

The proposed work has been based on investigating the impact of deep turbulence on the propagation of the laser beam using the phase screen approach for plane wave propagation through a long path [44].

9.2. General conditions.

The propagation geometries are shown in the Figure 9.3. The laser beacon platform is located on Mauna Loa mountain at elevation of 3,058 m and the receiver telescope at the summit of Haleakala at 3,397 m. The distance in between the two mountains is 150 km.

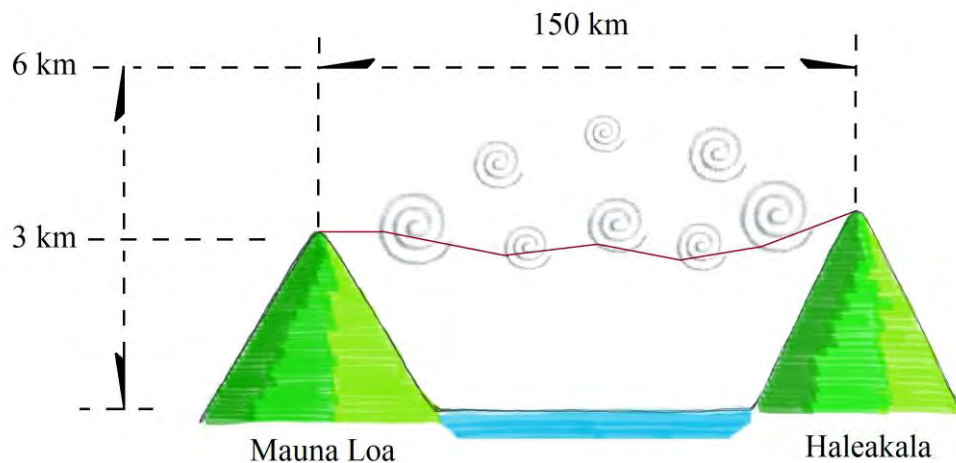


Figure 9.3. General Geometry.

The velocity range that was used for the analysis is 5 to 16 m/s, which is taken as the basis of a typical Hawaiian summer day where the wind velocities between these mountains ranges from 10 to 35 mph. Due to the height of the laser beacon in Mauna Loa, the ambient temperature at that point is 5.5 °C during the daylight varying from a temperature of 15 °C on the sea surface and 25 °C on the base surface of the mountains [40].

Similarly, the variations in pressure are from 101,325 Pa at sea level to 70,630.21 Pa at a height of 3,397 m where the laser emitting device is located.

To carry out my research, I needed to simulate the turbulent behavior of atmospheric conditions among the previously mentioned mountains. Therefore, I needed a CFD simulator and I chose the ANSYS Fluent because of its precision and that it has accurate turbulent flow analysis options [44]. The design process was extensively discussed in Chapter 7 so it will only be mentioned in this chapter as a general summary.

Another big challenge apart from deciding the scaling down the model was to replicate the original conditions of temperature and pressure all over the scaled model [44]. The situation is that in the original geometry, these properties change with the height, then I needed to create a User Defined Function, or UDF, to ensure these variations. The UDF function was used in the inlet, outlet, top, front, and back atmosphere.

The Figure 9.4. shows the boundary conditions of pressure, temperature, and velocity in each section, as well the UDF to specify the variation of these properties with respect to height.

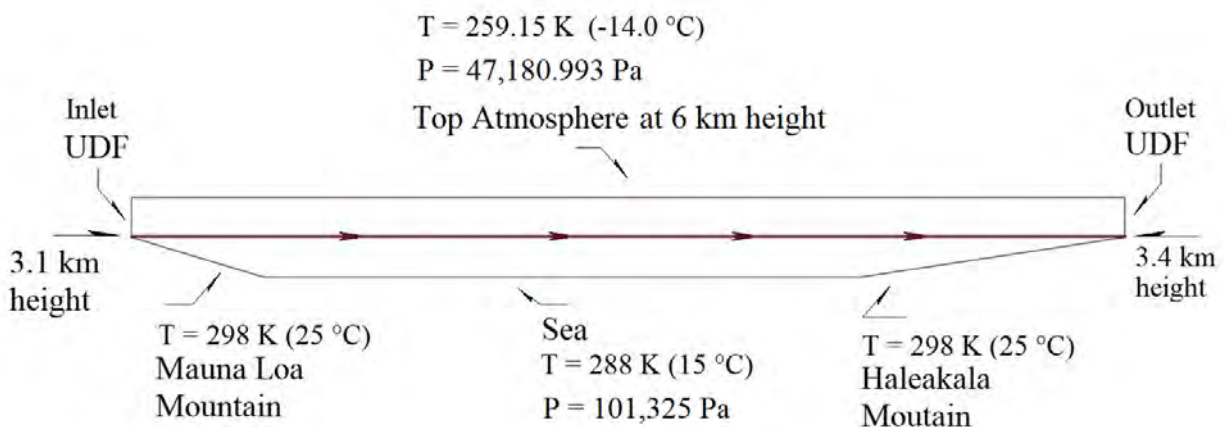


Figure 9.4: Temperature and pressure data in the specified sections.

Equations (9.1) and (9.2) were the ones used to generate the code for the variation of temperatures and pressures.

$$T(h) = 298.15 + 65 h \quad (9.1)$$

$$P(h) = 101325 h [1 - (2.25577 \times 10^{-5})]^{5.25588} \quad (9.2)$$

Figure 9.5 shows the code in C language that was generated with formulas (9.1) and (9.2) to create the UDF that was later uploaded into the *Set up* section of the ANSYS Fluent to run the 45 cases of analysis.

```

1  #include "udf.h"
2  DEFINE_PROFILE(temperature_profile, thread, nv)
3  {
4      float pos[3];
5      float z;
6      face_t f;
7      begin_f_loop(f, thread);
8      {
9          F_CENTROID(pos, f, thread);
10         z = pos[2];
11         F_PROFILE(f, thread, nv) = 298.15 - 65 * z;
12     }
13     end_f_loop(f, thread);
14 }
15
16 #include "udf.h"
17 DEFINE_PROFILE(pressure_profile, thread, nv)
18 {
19     float pos[3];
20     float z;
21     face_t f;
22     begin_f_loop(f, thread);
23     {
24         F_CENTROID(pos, f, thread);
25         z = pos[2];
26         F_PROFILE(f, thread, nv) = 101325 * pow((1 - 0.0000225577 * (z * 10000)), 5.25588);
27     }
28     end_f_loop(f, thread);
29 }

```

Figure 9.5. Codes to generate the UDF in C language for temperature and pressure.

Chapter 10: Analysis.

During the analysis stage, the simulation calculations are run in the CFD software which uses the information previously fed. The case of analysis is initialized and subsequently run with a time step size of 0.002 and a number of time steps of 100,000 with a maximum of iterations per time step of 5. In order to handle this high volume of calculations, HPCs (High Performance Computers) is used for solving the cases [55].

The cases of analysis were generated based on the typical velocity range in this area of Hawaii, where the values have been identified from 10 to 35 mph. Using the conversion to International Units System and with the purpose of using a step size of 0.25 m/s, a range from 5 to 16 m/s was established. Thus, forty-five cases of analysis were defined.

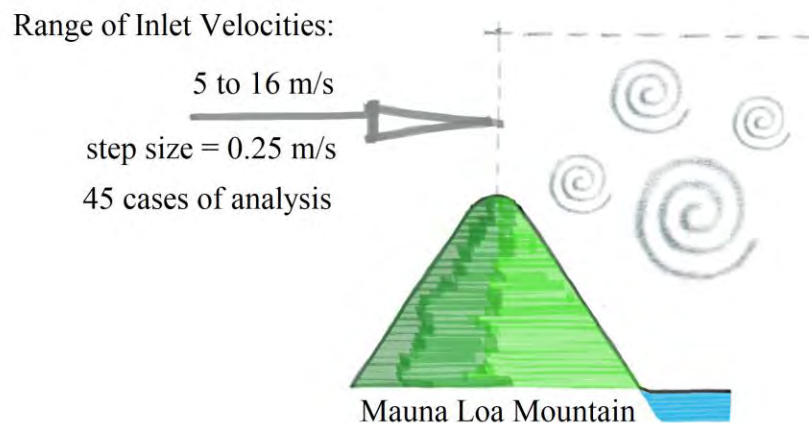


Figure 10.1. Range of velocities for the 45 cases of analysis.

The Figure 10.2 shows the general velocity profile when the case was calculated based on an inlet velocity of 5 m/s on the Large Eddie Simulation method with high turbulent conditions.

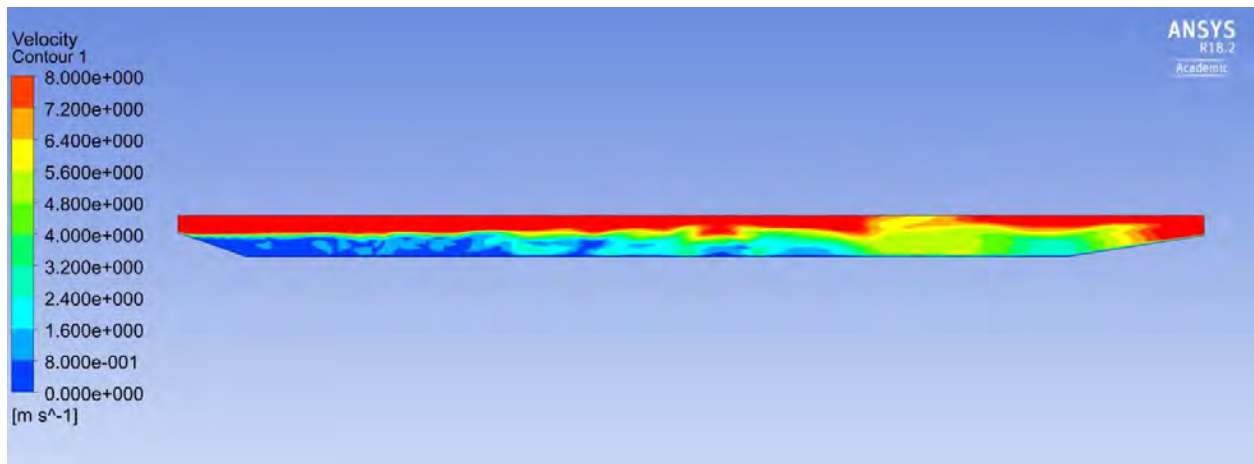


Figure 10.2. Velocity behavior in the turbulent environment.

Each case is run in the Fluent section. After this, from the Results section, the data related to velocities, temperatures and pressures over each coordinate is exported and stored into a general .csv file, so it can be used later on other stages of analysis [56].

The .csv files with the information of coordinates, velocities, temperatures and pressures is defined in non-equidistant coordinates, as it can be seen in the figure 10.3. Thus, it is submitted to a code written in Python language that uses training and prediction tools based on neural algorithms previously defined into the *Chapter 8* about Deep Learning Analysis section [57].

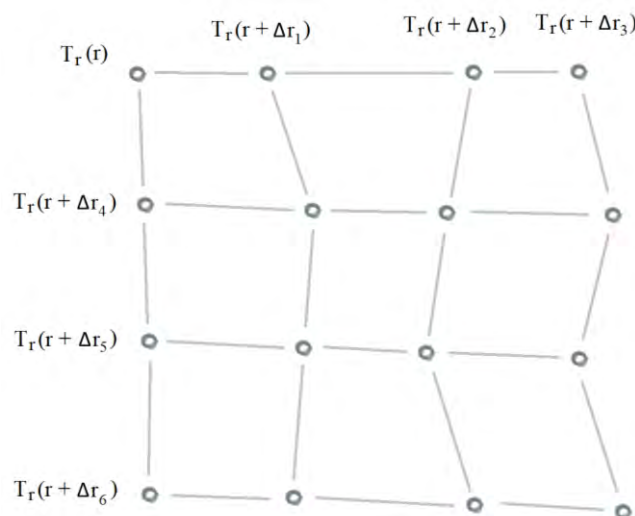


Figure 10.3. Non equidistant mesh from the ANSYS File.

The neural network algorithm generates a .csv file with the data for velocity, temperature and pressure adjusted to equidistant positions as it can be seen on figure 10.4 [48].

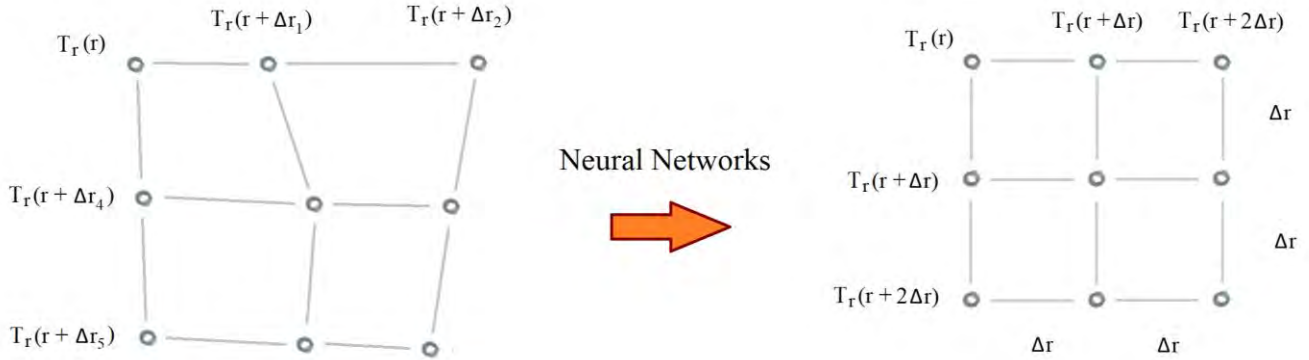


Figure 10.4. Transition from Non-Equidistant to Equidistant Mesh.

The reason of generating equidistant positions is to be able to use the structure function of the temperature which is based on doing the summation of the square difference of the temperatures depending on the position vector r_1 and using a distance r whose value must remain constant during all the summation [10]:

$$D_T = \sum [T_r(r_1 + r) - T_r(r_1)]^2 \quad (10.1)$$

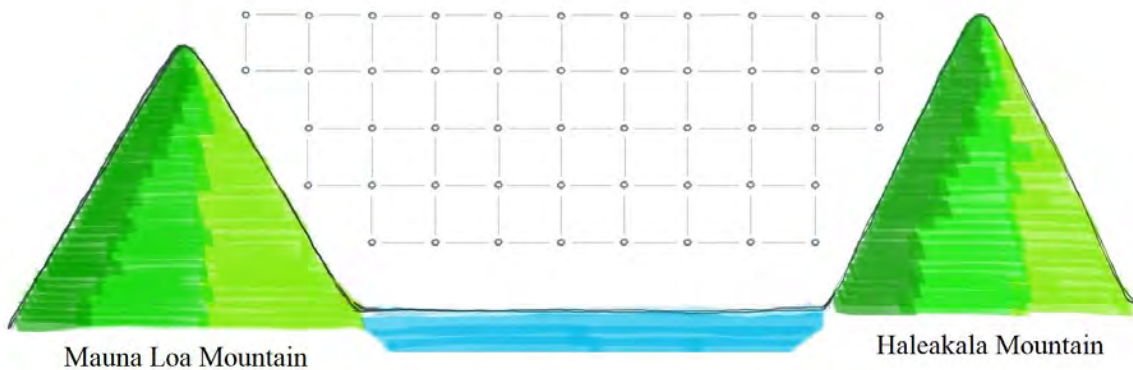


Figure 10.5. Temperature information in equidistant positions.

Once the structural function of temperature has been calculated, the following equations are used to obtain C_T^2

$$D_T = C_T^2 r^{2/3} \quad (3.5)$$

$$C_T^2 = D_T / r^{2/3} \quad (10.2)$$

The result of the calculations is the structural temperature index C_T^2 and can be considered a universal way of calculating spatial temperature variations. After the C_T^2 was calculated using the python scripts, a new .csv file is created. Thus, the next step is using this information to calculate the refractive index for each case of analysis using the equation (3.9).

$$C_n^2 = \left(\frac{79 P}{T^2} \times 10^{-6} \right)^2 C_T^2 \quad (3.9)$$

The values of P and T (Temperature and Pressure) used for the equation, are the characteristic readings at the laser emission point. In the results section, the range of values of C_n^2 come from these calculations.

Turbulence conditions in open atmosphere begin at a Reynolds number of 5×10^5 . As previously mentioned, the Reynolds number calculation comes from the equation [60].

$$Re = v L / \nu \quad (10.3)$$

The parameters that make up the equation are v , which is the wind velocity in a range from 5 to 16 m/s; L , which is the characteristic dimension of the flow zone, equal to 3 m in the scaled model and ν that is the kinematic viscosity of the air of 1.38×10^{-5} m/s at the temperature of the emission zone which is 5.5 °C or 278.65 K.

The Figure 10.6 specifies the values of the Reynolds numbers corresponding to the range of velocities from 5 to 16 m/s. This range is from 1.09×10^6 to 3.47×10^6 . Thus, the entire range is above the transition value of 5×10^5 that indicates the change from laminar to turbulent flow. Therefore, the entire analyzed phenomenon is at the level of turbulence.

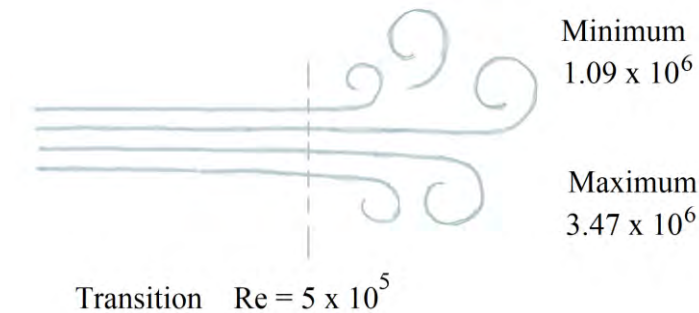


Figure 10.6. Range of Reynolds numbers.

In the Chapter 11, the series of calculations made for the Reynolds number are presented, as well as those corresponding to the refractive index C_n^2 and the graph that relates both parameters.

Chapter 11: Results.

As a result of the analysis previously carried out, the following Table 11.1 was obtained, from which the Reynolds numbers and the calculations made for C_n^2 can be observed [1].

Table 11.1: Reynolds vs C_n^2

<i>Item</i>	<i>Reynolds</i>	C_n^2
1	1,085,384	1.1093E-16
2	1,139,653	6.7988E-17
3	1,193,922	9.3066E-17
4	1,248,191	8.7333E-18
5	1,302,460	1.3023E-16
6	1,356,729	4.1944E-17
7	1,410,999	1.4309E-16
8	1,465,268	6.0641E-17
9	1,519,537	1.1440E-16
10	1,573,806	6.4804E-17
11	1,628,075	8.8587E-17
12	1,682,344	5.7416E-17
13	1,736,614	7.4705E-17
14	1,790,883	3.4308E-17
15	1,845,152	1.4612E-16
16	1,899,421	4.6558E-17
17	1,953,690	1.5267E-16
18	2,007,959	6.3539E-17
19	2,062,229	1.2321E-16
20	2,116,498	5.8475E-17
21	2,170,767	1.5050E-16
22	2,225,036	5.6085E-17
23	2,279,305	1.7614E-16
24	2,333,575	7.5427E-17
25	2,387,844	2.2133E-16
26	2,442,113	7.0411E-17
27	2,496,382	1.5254E-16
28	2,550,651	6.5826E-17
29	2,604,920	1.4452E-16
30	2,659,190	1.0770E-16
31	2,713,459	1.7352E-16
32	2,767,728	8.1912E-17
33	2,821,997	1.8998E-16
34	2,876,266	8.5792E-17
35	2,930,535	1.5846E-16
36	2,984,805	1.0408E-16
37	3,039,074	1.9199E-16
38	3,093,343	1.0095E-16
39	3,147,612	1.8985E-16
40	3,201,881	9.5771E-17
41	3,256,151	1.8040E-16
42	3,310,420	1.1728E-16
43	3,364,689	1.7846E-16
44	3,418,958	1.2195E-16
45	3,473,227	2.1001E-16

The figure 11.1 shows the behavior of the Reynolds number with respect to the value of the refractive index constant C_n^2 :

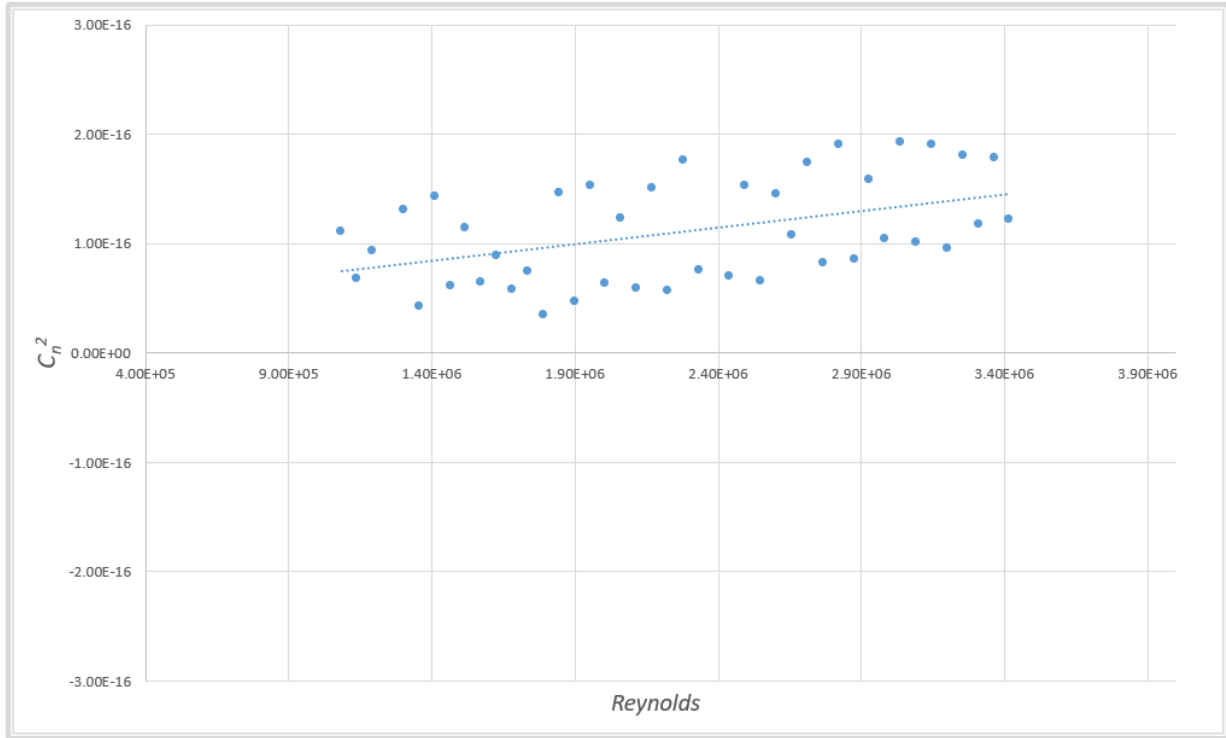


Figure 11.1: Re vs C_n^2 trend.

The equation (11.1) describes the trend of C_n^2 with respect to the Reynolds number.

$$C_n^2 = 3 \times 10^{-23} Re + 4 \times 10^{-17} \quad (11.1)$$

The positive relation between the Reynolds number and the C_n^2 can be seen from the graph and from the equation. It is a positive linear trend with a noticeable dispersion. When analyzing the graph, the size of this dispersion as well as certain patterns get the attention [2]. This could be explained on the base that the temperature values in a state of turbulence, show a lot of variations due to the chaotic nature of the phenomenon.

Nevertheless, the trend is clearly observed, for this reason I consider this estimation good enough for understanding the relation between the Reynolds number and the C_n^2 .

11.1. Correlation between Re and C_n^2 .

Based on the Table 11.1 of the Reynolds number vs C_n^2 , the correlation between both variables has been calculated. The value obtained is **0.511**.

A correlation analysis indicates that the values in the range from 0.5 to 1.0 have a strong positive correlation between both variables [1], meaning that they are likely to move in the same direction.

These values reinforce the initial objective of finding a relationship between the two variables that would allow predicting, above all, the value of C_n^2 with the Reynolds values as well as obtaining a mathematical relation between the two variables [2].

Chapter 12: Conclusions.

The equation shown in the present study as a result of the analysis of atmospheric conditions to determine the relationship between the Reynolds number and the refractive index C_n^2 represents an adequate platform for those who adjust the laser communication systems in the Mauna laboratory. Based on the equation, they can program automatic adjustments directly related to the instantaneous wind velocity present at their location.

Since the conditions to generate the equation were based on maximum wind velocities at a typical temperature scale, it follows that this equation allows the necessary adjustments on the optical systems to be used from the minimum to the critical turbulent conditions.

The use of a virtual platform such as the ANSYS Fluent, whose operation is based on the Navier-Stokes equations, represents an extremely useful tool to generate the profile of temperatures and velocities at different positions in the space at a given time, however, the necessary scaling of the model, represents a weakness that affects the accuracy of the predictions.

Additionally, the inaccuracy of the spatial positions in the sensitive mesh that CFD models create, despite having internal adjusting tools, requires the use of additional computational scripts to generate the required proportional spaces for analyzing the data.

This is where Neural Network algorithms, with their training and prediction steps, come with high accuracy to solve this need, as they are able to use data on the basis of a non-equidistant mesh to be trained and then predict a series of data on the basis of an equidistant mesh.

Finally, it can be said that the use of the Python programming platform was decisive in order to use all the data collected from our CFD Simulator (ANSYS Fluent), encode the neural network algorithms and work on the data obtained from this codification to perform the calculations of C_T^2 and C_n^2 in order to obtain the equation that relates Reynolds vs C_n^2 .

Chapter 13: Future Work.

Part of the future work that needs to be done is comparing the results that were generated here through virtual simulators with an experimental database with real atmospheric conditions between the Mauna Loa and the Haleakala mountains.

Additionally, it is required to implement the equation into the optical systems currently operating into the laboratories to carry out tests and the corresponding adjustments as part of a typical implementation process.

The present analysis was performed on the basis of an experimental ambient temperature scale based on a set point of 25 Celsius and a range of maximum wind velocities, however, future work may also include a range of average velocities as well as a range of nighttime ambient temperatures to know the behavior of C_n^2 at night.

References

- [1] M. Vorontsov, V. S. Rao Gudimetla, G. Carhart, T. Weyrauch, S. Lachinova, E. Polnau, J. Reiersen, L. Beresnev, J. Liu, and J. F. Riker. 2011. "Comparison of turbulence-induced scintillations for multi-wavelength laser beacons over tactical (7 km) and long (149 km) atmospheric propagation paths." Kehie, Hi. 2011. U. S. Air Force Research Laboratory.
- [2] M. Vorontsov et al., "Characterization of Atmospheric Turbulence Effects Over 149 km Propagation Path Using Multi-Wavelength Laser Beacons," Proc. AMOS Conference, 2010.
- [3] Abdilghanie, A. M., Collins, L. R., and Caughey, D. A., 2009, Comparison of Turbulence Modeling Strategies for Indoor Flows, J. Fluids Eng., 131(5).
- [4] A. Quirrenbach, "The Effects of Atmospheric Turbulence on Astronomical Observations," Exp. Astron., 2009.
- [5] L. C. Andrews and R. L. Phillips, Laser Beam Propagation through Random Media. 1000 20th Street, Bellingham, WA 98227-0010 USA: SPIE, 2005.
- [6] L. A. Chernov, Wave propagation in a random medium. Translated from the Russian by R. A. Silverman. New York: McGraw-Hill, 1960.
- [7] R. L. Phillips and L. C. Andrews, "Measured statistics of laser-light scattering in atmospheric turbulence," J. Opt. Soc. Am., vol. 71, no. 12, p. 1440, 1981.
- [8] A. Consortini, F. Cochetti, J. H. Churnside, and R. J. Hill, "Inner-scale effect on irradiance variance measured for weak-to-strong atmospheric scintillation," J. Opt. Soc. Am. A, vol. 10, no. 11, p. 2354, 1993.

- [9] D. Dayton, B. Pierson, B. Spielbusch, and J. Gonglewski, "Atmospheric structure function measurements with a Shack-Hartmann wave-front sensor," *Opt. Lett.*, vol. 17, no. 24, p. 1737, 1992.
- [10] R. Kerr, "Experiments on Turbulence Characteristics and Multiwavelength Scintillation Phenomena," *J. Opt. Soc. Am.*, vol. 62, no. 9, pp. 1040–1049, 1972.
- [11] E. Velar, J. Haddon, and J. Haddon, "Measurement and Modeling of Scintillation Intensity to Estimate Turbulence Parameters in an Earth-Space Path," *IEEE Trans. Antennas Propag.*, vol. 32, no. 4, pp. 340–346, 1984.
- [12] Clifford, S. F. "The classical theory of wave propagation in a turbulent medium." *Laser beam propagation in the atmosphere*. Springer, Berlin, Heidelberg, 1978. 9-43.
- [13] Tatarskii, Valerian Ilitch. "The effects of the turbulent atmosphere on wave propagation." Jerusalem: Israel Program for Scientific Translations, 1971 (1971).
- [14] Schmidt, Jason. "Numerical simulation of optical wave propagation with examples in MATLAB." Society of Photo-Optical Instrumentation Engineers, 2010.
- [15] Yuksel, H., Harris, J., Tang, Y., Gammon, R., and Davis, C., 2008, Aperture Averaging and Correlation Function Measurements in Strong Atmospheric Turbulence for Optical Wireless Applications, *Proc.SPIE*.
- [16] A. Kolmogorov, In *Turbulence, Classic Papers on Statistical Theory*. New York: Interscience, 1961.
- [17] A. Zilberman, E. Golbraikh, and N. S. Kopeika, "Propagation of electromagnetic waves in Kolmogorov and non-Kolmogorov atmospheric turbulence: three-layer altitude model.," *Appl. Opt.*, vol. 47, no. 34, pp. 6385–91, 2008.

- [18] J. P. Bos, M. C. Roggemann, and V. S. Rao Gudimetla, “Anisotropic non-Kolmogorov turbulence phase screens with variable orientation,” *Appl. Opt.*, vol. 54, no. 8, p. 2039, 2015.
- [19] V. S. R. Gudimetla, R. B. Holmes, and J. F. Riker, “Analytical expressions for the logamplitude correlation function for plane wave propagation in anisotropic non-Kolmogorov refractive turbulence,” *J. Opt. Soc. Am. A*, vol. 29, no. 12, p. 2622, 2012.
- [20] C. Smith, “GoldsteinUnwrap2D_r1.” MATLAB 7.11 (R2010b), 2010. Markov, V., Khizhnyak, A., Woll, D., Trolinger, J., and Eaton, F., 2006, “Optical Phase Conjugation for Atmospheric Turbulence Assessment,” *Proc.SPIE*.
- [21] R. S. Lawrence, G. R. Ochs, and S. F. Clifford, “Measurements of atmospheric turbulence relevant to optical propagation,” *J. Opt. Soc. Am.*, vol. 60, no. 6, pp. 826–830, 1970.
- [22] B. E. Stribling, “Optical propagation in non-Kolmogorov atmospheric turbulence,” in *Proceedings of SPIE*, 1995, vol. 2471, pp. 181–196.
- [23] J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation with Examples in MATLAB*. 2010.
- [24] E. Masciadri and P. Jabouille, “Improvements in the optical turbulence parameterization for 3D simulations in a region around a telescope,” *Astron. Astrophys.*, vol. 376, no. 2, pp. 727–734, 2001.
- [25] Shadloo, M. S., and Yildiz, M., 2011, “Numerical Modeling of Kelvin–Helmholtz Instability Using Smoothed Particle Hydrodynamics,” *Int. J. Numer. Methods Eng.*, 87(10), pp. 988–1006.
- [26] T. Chacón Rebollo and R. Lewandowski, *Mathematical and Numerical Foundations of Turbulence Models and Applications*. New York, NY: Springer New York, 2014.

- [27] F. H. Harlow and J. E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Phys. Fluids*, vol. 8, no. 12, p. 2182, 1965.
- [28] D. L. Walters and D. K. Miller, "Evolution of an upper-tropospheric turbulence event—Comparison of observations to numerical simulations," in *Preprints, 13th Symposium on Boundary Layer Turbulence*, 1999, pp. 157–160.
- [29] Vassilicos, J. C., 2015, "Dissipation in Turbulent Flows," *Annu. Rev. Fluid Mech.*, 47(1), pp. 95–114.
- [30] Tsubokura, M., 2005, "Subgrid Scale Modeling of Turbulence for the Dynamic Procedure Using Finite Difference Method and Its Assessment on the Thermally Stratified Turbulent Channel Flow," *J. Appl. Mech.*, 73(3), pp. 382–390.
- [31] Mizuno, Y., Yagi, T., and Mori, K., 2018, "Contribution of Attached Eddies to the Intensity and Turbulent Transfer of Velocity Fluctuations in Wall-Turbulence," *Fluid Dyn. Res.*, 50(4), p. 45513.
- [32] White, C. M., & Mungal, M. G. (2008). *Mechanics and prediction of turbulent drag reduction with polymer additives*. *Annu. Rev. Fluid Mech.*, 40, 235-256.
- [33] G. Weiss, "Wave Propagation in a Turbulent Medium. V. I. Tatarski. Translated by R. A. Silverman. McGraw-Hill, New York, 1961. 285 pp. Illus. \$9.75," *Science* (80-.), vol. 134, no. 3475, pp. 324–325, Aug. 1961.
- [34] J. W. Strohbehn, "Line-of-sight wave propagation through the turbulent atmosphere," *Proc. IEEE*, vol. 56, no. 8, pp. 1301–1318, 1968.

- [35] F. G. Schmitt, "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity," *Comptes Rendus - Mecanique*, vol. 335, no. 9–10. pp. 617–627, 2007.
- [36] Bernard, Peter S., and James M. Wallace. *Turbulent flow: analysis, measurement, and prediction*. John Wiley & Sons, 2002.
- [37] DalBello, T., Dippold, V., Georgiadis, N. J., and Center, N. G. R., 2005, *Computational Study of Separating Flow in a Planar Subsonic Diffuser*, National Aeronautics and Space Administration, Glenn Research Center.
- [38] Emori, Richard I., and Dieterich J. Schuring. *Scale models in engineering: Fundamentals and applications*. Elsevier, 2016.
- [39] Heller, Valentin. "Scale effects in physical hydraulic engineering models." *Journal of Hydraulic Research* 49.3 (2011): 293-306.
- [40] Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.
- [41] Anderson, J. D., & Wendt, J. (1995). *Computational fluid dynamics* (Vol. 206). New York: McGraw-Hill.
- [42] Y. Zhiyin, "Large-eddy simulation: Past, present and the future," *Chinese J. Aeronaut.*, vol. 28, no. 1, pp. 11–24, 2015.
- [43] F. Ducros, F. Nicoud, and T. Poinso, "Wall-adapting local eddy-viscosity models for simulations in complex geometries," *Conf. Numer. Methods Fluid Dyn.*, pp. 1–7, 1998.
- [44] Ansys, "Ansys fluent 12.0 Theory Guide," ANSYS Inc., no. April, pp. 49–53, 2009.

- [45] Menter, Florian R. "Two-equation eddy-viscosity turbulence models for engineering applications." *AIAA journal* 32.8 (1994): 1598-1605.
- [46] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117.
- [47] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444
- [48] R. S. Michalski, I. Bratko, and A. Bratko, Eds., *Machine Learning and Data Mining; Methods and Applications*. New York, NY, USA: John Wiley & Sons, Inc., 1998.
- [49] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, "GeePS : Scalable deep learning on distributed GPUs with a GPU-specialized parameter server."
- [50] T. Chen et al., "MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," pp. 1–6.
- [51] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems."
- [52] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," pp. 1–10, 2016.
- [53] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and Checkerboard Artifacts," *Distill*, vol. 1, no. 10, Oct. 2016.
- [54] V. S. Gudimetla, "Estimating Cn2 (Refractive index structure constant) in Deep Turbulence using Unwrapped Phase," in *Imaging and Applied Optics*, 2013, p. PTu2F.1.
- [55] Miettinen, A., and Siikonen, T., 2015, "Application of Pressure- and Density-Based Methods for Different Flow Speeds," *Int. J. Numer. Methods Fluids*, 79(5), pp. 243–267.

- [56] Galántai, A., 2000, “The Theory of Newton’s Method,” J. Comput. Appl. Math., 124(1), pp. 25–44.
- [57] Ypma, T. J., 1995, “Historical Development of the Newton-Raphson Method,” SIAM Rev., 37(4), pp. 531–551.
- [58] Baines, P. G., and Mitsudera, H., 1994, “On the Mechanism of Shear Flow Instabilities,” J. Fluid Mech., 276, pp. 327–342.
- [59] A. Wheelon, Electromagnetic Scintillation I. Geometrical Optics, 1st ed. New York: Cambridge University Press, 2001.
- [60] J. M. McDonough, “Lectures in Elementary Fluid Dynamics: Physics, Mathematics and Applications,” Analysis, p. 163, 2009.
- [61] Jimenez, Diego Alberto Lozano. "Implementing Large Eddy Simulation to Numerical Simulation of Optical Wave Propagation." (2018).

Vita

My name is Luis Fernando Rodriguez Sanchez and I am pursuing a doctoral degree in Mechanical Engineering at the University of Texas at El Paso under Dr. Vinod Kumar as my faculty advisor. Before starting my doctoral studies at UTEP, I practiced for ten years as professor of the Mechanical Engineering Department at the prestigious university Tec de Monterrey in Mexico. During this period, I had the opportunity to serve as director of the undergrad programs of Mechanical Engineering and Automotive Engineering.

Before starting my labor as a professor, I gained extensive experience for ten years in the fields of automotive and electronics manufacturing. In the automotive production lines, I focused above all in the redesign of structural systems such as the chassis' components of the Chrysler's heavy-duty trucks. As for the electronic production lines I focused on the implementation of new projects in the manufacturing lines of the company named Flextronics.

The experience gained in the industry as well as in the educational environment, allows me to be involved very strongly into the research projects because I have the knowledge of the productive branch as well as the one of the educational branch where it is possible to focus on a large number of projects that the industry requests from the universities.

Contact Information: fernando.rodriguez@live.com