



Finalist Reports 2018-2019



<https://supercomputingchallenge.org>



Printed by the
New Mexico Public Education Department
Compiled by
Los Alamos National Laboratory
High Performance Computing Group (HPC-ENV)

Cover: **Human Intelligence vs Machine Logic**

Team 12: Eldorado High /NexGen
Brendan Kuncel, Quentin Dye, Savannah Phelps

Sponsor: Karen Glennon
Winner in the Technical Poster Competition



Notification: These final reports are presented in an abridged form, leaving out actual code, color, and appendices where appropriate. Complete copies of most of the final reports are available from the archives of Supercomputing Challenge web site:
<https://supercomputingchallenge.org> .

New Mexico Supercomputing Challenge 2018 – 2019 Finalist Reports

Table of Contents

About the New Mexico Supercomputing Challenge

For more information, please visit our website at

<https://supercomputingchallenge.org> 2

2018—2019 Challenge Awards 4

Scholarship winners 7

Sponsors 8

Participants 9

Judges 13

Finalist Reports 15

1. *A Novel Computational Tool to Inform Cost-Effective Nutrition Interventions in Sub-Saharan Africa*, Los Alamos High School Team 28
(**Technical Writing Award winner**)
2. *Computer Simulation as a Tool for Risk Assessment of Coastal Areas*, Los Alamos High School Team 97
3. *Traffic Model*, Albuquerque High School Team 4
4. *Data-Based Approach to Estimating Ice-Shelf Melt Rates*, Los Alamos High School Team 30
5. *Detecting Shocks Waves with Artificial Intelligence*, Barranca Mesa Elementary School Team 5
6. *Using QAOA to Solve NP-Hard Problems on NISQ Computers*, Los Alamos High School Team 29
7. *What is the Radius of the Earth's Core?* Los Alamos Middle School Team 32
8. *Optimizing Flapping-Wing Flight*, New Mexico School for the Arts Team 56



Supercomputing Challenge Vision

The Vision of the Supercomputing Challenge is to be a nationally recognized program that promotes computational thinking in science and engineering so that the next generation of high school graduates is better prepared to compete in an information-based economy.

Supercomputing Challenge Mission

The Mission of the Supercomputing Challenge is to teach teams of middle and high schools students how to use powerful computers to analyze, model and solve real world problems.

About the Supercomputing Challenge

The Supercomputing Challenge (the Challenge) is an exciting program that offers a truly unique experience to students in our state. The opportunity to work on the most powerful computers in the world is currently available to only a very few students in the entire United States, but in New Mexico, it is just one of the benefits of living in the "Land of Enchantment."

The Challenge is a program encompassing the school year in which teams of students complete science projects using high-performance supercomputers. Each team of up to five students and a sponsoring teacher defines and works on a single computational project of its own choosing. Throughout the program, help and support are given to the teams by their project advisors and the Challenge organizers and sponsors.

The Challenge is open to all interested students in grades 5 through 12 on a nonselective basis. The program has no grade point, class enrollment or computer experience prerequisites. Participants come from public, private, parochial and home-based schools in all areas of New Mexico. The important requirement for participating is a real desire to learn about science and computing.

Challenge teams tackle a range of interesting problems to solve. The most successful projects address a topic that holds great interest for the team. In recent years, ideas for projects have come from Astronomy, Geology, Physics, Ecology, Mathematics, Economics, Sociology, and Computer Science. It is very important that the problem a team chooses is what we call "real world" and not imaginary. A "real world" problem has measurable components. We use the term Computational Science to refer to science problems that we wish to solve and explain using computer models.

Those teams who make significant progress on their projects can enter them in the competition for awards of cash and scholarships. Team trophies are also awarded for: Teamwork, Best

Written Report, Best Professional Presentation, Best Research, Creativity and Innovation, Environmental Modeling, High Performance, Science is Fun and the Judges' Special Award, just to name a few.

The Challenge is offered at minimal cost to the participants or the school district. It is sponsored by a partnership of federal laboratories, universities, and businesses. They provide food and lodging for events such as the kickoff conference during which students and teachers are shown how to use supercomputers, learn programming languages, how to analyze data, write reports and much more.

These sponsors also supply time on the supercomputers and lend equipment to schools that need it. Employees of the sponsoring groups conduct training sessions at workshops and advise teams throughout the year. The Challenge usually culminates with an Expo and Awards Ceremony in the spring at the Los Alamos National Laboratory or in Albuquerque. In 2017 we started alternating the Expo and Awards Ceremony between Albuquerque and Los Alamos.

History

The New Mexico High School Supercomputing Challenge was conceived in 1990 by former Los Alamos Director Sig Hecker and Tom Thornhill, president of New Mexico Technet Inc., a nonprofit company that in 1985 set up a computer network to link the state's national laboratories, universities, state government and some private companies. Sen. Pete Domenici, and John Rollwagen, then chairman and chief executive officer of Cray Research Inc., added their support.

In 2001, the Adventures in Supercomputing program formerly housed at Sandia National Laboratories and then at the Albuquerque High Performance Computing Center at the University of New Mexico merged with the former New Mexico High School Supercomputing Challenge to become the New Mexico High School Adventures in Supercomputing Challenge.

In 2002, the words "High School" were dropped from the name as middle school teams had been invited to participate in 2000 and had done well.

In the summer of 2005, the name was simplified to the Supercomputing Challenge.

In 2007, the Challenge began collaborating with the middle school Project GUTS, (Growing Up Thinking Scientifically), an NSF grant housed at the Santa Fe Institute.

In 2013, the Challenge began collaborating with New Mexico Computer Science for All, an NSF funded program based at the Santa Fe Institute that offers a comprehensive teacher professional development program in Computer Science including a University of New Mexico Computer Science course for teachers.

2018—2019 Challenge Awards

29th Annual New Mexico Supercomputing Challenge winners



Left to Right: **Logan Dare**, Los Alamos Middle, **Robert Straus**, Los Alamos High, **Lillian Petersen**, Los Alamos High, **Christie Djidjev**, Los Alamos High, **Rishi Tikare Yang**, Albuquerque High, (not pictured: Garyk Brix from Winston Churchill High School)

Tuesday April 30, 2019 was the culmination of the 29th Annual New Mexico Supercomputing Challenge held in Albuquerque at the Central New Mexico (CNM) main campus and the University New Mexico (UNM) Science and Technology Park.

First Place

Lillian Petersen from Los Alamos High School and Garyk Brix from Winston Churchill High School, Potomac, Maryland won first place for their project, “A Novel Computational Tool to Inform Cost-Effective Nutrition Interventions in Sub-Saharan Africa”, See their final report.

https://supercomputingchallenge.org/18-19/finalreports/28/Supercomputing_Challenge_Report.pdf.

They have presented their prior work at international, professional conferences on nutrition in Africa.

Their research has three components. From their executive summary:

Malnutrition contributes to nearly half of childhood deaths, while treatment reaches a small fraction of those in need. Treatment delivery is hampered by costly ingredients and inefficient supply chains. Here we develop a three-component tool to inform acute malnutrition treatment interventions.

First, we forecast the geospatial demand for acute malnutrition treatment using a machine learning algorithm.

Second, we optimize low-cost recipes for specialized nutritious foods while meeting nutritional standards. Recipes were optimized for both international production and local production in 24 sub-Saharan African countries, and both achieved ingredient cost reductions of up to 60% compared to current recipes.

Third, we model a supply chain of the optimal production and distribution of acute malnutrition treatment with both international and local factories while accounting for production and transportation costs.

Their machine learning algorithm uses 41 data sources ranging from real-time satellite imagery to static demographic data. The most significant predictor turned out to be female education level.

They have partnered with Valid Nutrition an NGO from Kenya (www.validnutrition.org) to prototype their recipes and verify the cost and nutritional content.

Second Place

Second place went to Robert Strauss from Los Alamos High School and Logan Dare from Los Alamos Middle School. The title of their project is “Simulation as a Tool for Risk Assessment of Coastal Areas”. See their final report at:

https://www.supercomputingchallenge.org/18-19/finalreports/97/Team_97_report.pdf

Their model was used to study the 2018 tsunami which devastated Palu city, Indonesia and the 2018 Anak Krakatoa tsunami. Their computational model successfully predicts both the areas of extreme damage and safe areas with little damage.

Our first and second place teams did astonishing professional level work!

We highly recommend you follow the above links and take a closer look at these projects.

Third Place

There was a tie for third place: Rishi Tikare Yang from Albuquerque High with a “Traffic Model” See the final report. <https://www.supercomputingchallenge.org/18-19/finalreports/4/TrafficModelFinalReport.pdf>

And, Christie Djidjev from Los Alamos High with a “Data-Based Approach to Estimating Ice-Shelf Melt Rates. See the final report.

https://www.supercomputingchallenge.org/18-19/finalreports/30/final_report_team_LAHS30.pdf.

Finalist Teams



All the finalist teams received plaques for their schools, a large banner suitable for hanging at their schools, and other gifts. Many “goodies” were distributed by Supercomputing Challenge sponsors.

Scholarships and Other Awards

A complete list of all winning student teams.

<https://supercomputingchallenge.org/18-19/expo/AllWinnersList.pdf>



Scholarship winners

Scholarships worth \$27,400 were awarded at the Supercomputing Challenge Awards Ceremony to 13 seniors: Christie Djivjev, Quentin Dye, Destiny Erickson, Delaney Galligan, Ariana Garcia, Occam Kelly-Graves, Jeremy Lawrence, Danny Luong, Torrey Luong, Nigone "Owen" Phommachack, Elijah Pelofske, Alexis Smith, and Manoj Subedi.

Many other awards were distributed ranging from random \$100 gifts for finishing the academic marathon to team prizes for teamwork, programming prowess, and environmental impact, etc.

All final reports are online.

<https://www.supercomputingchallenge.org/18-19/finalreports/submitted.php>

Activities

On Monday, April 29, 2019, student teams presented their projects to panels of judges.

Demonstrations of technology and tours throughout the day were provided by:

- Sandia National Laboratories,
- RiskSense,
- Harry Henderson, Rio Rancho Public Schools,
- Air Force Research Lab,
- Fuse Makerspace,
- CNM Ingenuity,

- campus tours of CNM and UNM,
- Old Town Walking tours, and
- individual team visits to Albuquerque museums.

In conjunction with the judging Expo and Award Events, the Supercomputing Challenge held a networking event at the Smith Brasher Auditorium, CNM, with the New Mexico Tech Council.

About the Supercomputing Challenge

“The goal of the yearlong event is to teach student teams how to use powerful computers to analyze, model and solve real-world problems,” said David Kratzer, Executive Director.

“Participating students improve their understanding of technology by developing skills in scientific inquiry, modeling, computing, communications, and teamwork.”

The New Mexico Supercomputing Challenge teaches written and oral communication, collaboration with peers and professionals, critical thinking including research and coding including computer modeling to middle and high school students throughout the state. Any New Mexico middle-school or high-school student, including home-schooled students are eligible to participate in the Supercomputing Challenge. Students follow their own interests to choose a topic to model.

Sponsors

The Supercomputing Challenge this year is mainly sponsored by:

- Los Alamos National Laboratory,
- New Mexico Consortium,
- Sandia National Laboratories,
- VanDyke Software, and
- National Science Foundation Space and Trustworthy Program.

A complete list of sponsors and supporters of the Challenge is on the website at <https://supercomputingchallenge.org/18-19/sponsors.php>.

Teams Finishing the Challenge and submitting final reports:

Team 1, Academy For Technology and the Classics, *Detecting Developmental Delays*

Team Members: Etta Pope, Shyla Sharma, Odin Frostad

Sponsor: Jenifer Hooten

Team 4, Albuquerque High, *Traffic Model*

Team Member: Rishi Yang

Sponsor: Joan Newsom

Team 5, Barranca Mesa Elementary, *Detecting Shocks Waves with Artificial Intelligence*

Team Member: Andrew Morgan

Sponsor: Nathaniel Morgan

Team 6, Capital High, *The Traveling Salesman Problem*

Team Members: Ariana Garcia, Arath Oscar Flores, Carlos Lopez, Melissa Gill

Sponsor: Irina Cislaru

Team 7, Capital High, *The Strength of Crystals*

Team Members: Destiny Erickson, Hansel Chavez, Michael Montano

Sponsor: Irina Cislaru

Team 10, Multi Schools-Cottonwood Classic Prep/Cleveland Middle School,
Albuquerque Fire Department Wait Times

Team Members: Ayvree Urrea, Kiara Onomoto

Sponsor: Karen Glennon

Team 11, Multi Schools-Eldorado High/Manzano High/Sandia High,
How Does Nutrition Affect Depression?

Team Members: Nancy Avila, Gwenevere Caouette, Tiffany Chau, Kyreen White

Sponsors: Karen Glennon, Sharee Lundsford

Team 12, Multi Schools-Eldorado High/NexGen, *Human Intelligence vs. Machine Logic*

Team Members: Brendan Kuncel, Quentin Dye, Savannah Phelps

Sponsor: Karen Glennon

Team 20, Jackson Middle School, *Glioblastoma*

Team Members: Isarel Chirino, Santiago Herrera

Sponsor: Sharee Lundsford

Team 21, La Cueva High, *Projectile Motion and Drag Forces*

Team Members: David Feng, Blake Watson, Luke Xue, Brad Zhang

Sponsor: Terri Jones

Team 22, Las Cruces High, *Water Usage Simulator Anabelle Fortin*

Team Member: Hannah Himelright

Sponsor: Lauren Curry

Team 26, Las Cruces High, *Parking Safety and Convenience*

Team Members: Tanner Nielsen, Phillip Ortiz, Lazaro Tiqui

Sponsor: Lauren Curry

Team 27, Las Cruces High, *Using Epidemiology to Model Computer Viruses*

Team Members: Steven Fraga, Karson Million, Ben Widner, Breanna Widner

Sponsor: Lauren Curry

Team 28, Los Alamos High, *A Novel Computational Tool to Inform Cost-Effective Nutrition Interventions in Sub-Saharan Africa*

Team Members: Lillian Petersen, Garyk Brixi

Sponsor: Allan Didier

Team 29, Los Alamos High, *Using QAOA to Solve NP-Hard Problems on NISQ Computers*

Team Member: Elijah Pelofske

Sponsor: Allan Didier

Team 30, Los Alamos High, *Data-Based Approach to Estimating Ice-Shelf Melt Rates*

Team Member: Christie Djidjev

Sponsor: Mark Petersen

Team 32, Los Alamos Middle School, *What is the Radius of the Earth's Core?*

Team Member: Veronica Sofia Parra

Sponsor: Curtis Terrill

Team 45, House High/Melrose High, *Turtles vs. Pollution*

Team Members: Evelyn Garrett, Alexis Smith, Gracie Sanchez, Madison Garrett

Sponsor: Alan Daugherty

Team 46, Melrose High, *Till the Cows Come Home*

Team Members: Hunter Hollaway, Rilee Priest

Sponsor: Alan Daugherty

Team 47, Melrose High, *4U App*

Team Members: Chantilly Fulgham, Erika House

Sponsor: Alan Daugherty

Team 48, Melrose Middle, *Crop Dust*.

Team Members: Heidi Macfarlane, Eoghen Knight, Harvey Peacock

Sponsor: Alan Daugherty

Team 49, Melrose Middle, *Mustangs of America*

Team Members: Lily Macfarlane, Lilly Gallagher

Sponsor: Alan Daugherty

Team 53, Mesilla Valley Leadership Academy, *Kissing Disease*
Team Members: Stephanie Hernandez, Jaqueline Jimenez, Jelli Reyna
Sponsor: Tracie Mikesell

Team 54, Mesilla Valley Leadership Academy, *Michael the “human”*
Team Members: Diego Romero, Alberto Castaneda, Jouquin Salizar, Uriel Meledez
Sponsor: Tracie Mikesell

Team 56, New Mexico School for the Arts, *Optimizing Flapping-Wing Flight*
Team Member: Occam Kelly Graves
Sponsor: Jennifer Black

Team 59, New Mexico Tech Upward Bound Math & Science,
The Mission of Plastic Decomposition: Fungi Edition
Team Members: Seth Griffen, Jeremy Lawrence, Danny Luong, Torrey Luong
Sponsors: Karen Glennon, Kurtis Griess

Team 60, New Mexico Tech Upward Bound Math & Science, *The Smart Farm*
Team Members: Manoj Subedi, Jerrel White, Naomi Ramos, Alex Phommachack, Nigone Phommachack
Sponsors: Karen Glennon, Kurtis Griess

Team 72, Santa Fe High, *Air Pollution on a U.S Scale*
Team Members: Owen Attlesey, Mason Nichols
Sponsor: Brian Smith

Team 74, Sarracino Mid, *Cellular Automata*
Team Members: Sky Sessions, Angelica Jaquez, Elena Prieto
Sponsors: Teresa Apodaca, Lauri Capps

Team 75, Sarracino Mid, *Angry Boards*
Team Members: Elizabeth Moellenbrock, Andrew Moellenbrock
Sponsors: Teresa Apodaca, Lauri Capps

Team 80, Shiprock High, *Campus Alert System*
Team Members: Adam Pettigrew, Mya Harvey, Shi'son Tsosie, Lawrence Kee, Destiny Domingo
Sponsor: Natalie Johnston

Team 81, Socorro High, *Mangrove Madness*
Team Members: Elias Zheng, Cody Johnston, Rio Sessions
Sponsor: Jay Garcia

Team 82, Taos High, *Moving Water*
Team Members: Jenay Barela, Eben Bellas, Kineo Memmer
Sponsor: Tracy Galligan

Team 83, Taos High, *Doctors Assistant Program*

Team Members: Sawyer Solfest, Henry Tosta, Skylar Spriggs, J Ryan Cox

Sponsor: Tracy Galligan

Team 84, Taos High, *Avalanche Dog*

Team Members: Liam Davis, Eric Hernandez, Grace Goler

Sponsor: Tracy Galligan

Team 86, Media Arts Charter School, *Creating a Balanced Ecosystem Simulation*

Team Member: Jacob Bowden

Sponsor: Creighton Edington

Team 87, Media Arts Charter School, *Finding the Battleships*

Team Member: Seungbin Chung

Sponsor: Creighton Edington

Team 97, Los Alamos High,

Computer Simulation as a Tool for Risk Assessment of Coastal Areas

Team Members: Robert Strauss, Logan Dare

Sponsor: Mark Petersen

Team 1001, Los Lunas High, *Ready Set Grow!*

Team Member: Delsin Jaramillo

Sponsor: Anne Loveless

Team 1006, Taos High, *Stem Cell Therapy: Modeling the application of renewable stem cells on bodies affected by cancer*

Team Members: Delaney Galligan, Kyle Totman, Cyrus O' Hern

Sponsor: Tracy Galligan

Judges

Richard Alex, Eastern New Mexico University
Zeinab Akavan, University of New Mexico
Ed Angel, Professor Emeritus UNM
Camber Arnhart, University of New Mexico
Charlene Arias, Sandia National Laboratories
Richard Barrett, Sandia National Laboratory
Carly Beneke, Descartes Labs
Cheri Burch, NM Network of Women in Science
Chuck Burch, Conoco Phillips/Retired
Robin Caine-Rizza, Eastern New Mexico University
Jaime Cherry, PED Dual Credit Administrator
Sheri Clark, Clark Consulting Group, LLC
Steve Cox, Northern New Mexico College
Ken Cradock, Eastern New Mexico University
Jesse Crawford, Sandia National Laboratories
Varsha Dani, University of New Mexico
Geoff Danielson, Sandia National Laboratories
Mike Davis, Cray Inc
Howard DeLaCruz-Bancroft, University of New Mexico
Sharon Deland, Sandia National Laboratories
Celia Einhorn, Supercomputing Challenge
Drew Einhorn
Tony Espinoza, University of New Mexico
Quinton Flores, Eastern New Mexico University
Chris Garasi, Sandia National Laboratories
Susan Gibbs, Project GUTS
Kaley Goatcher, TransCore
Ann Gomez
Shadron Gudmunson, New Mexico Institute of Mining and Technology
Stephen Guerin, Simtable
Nels Hoenig, PNM
David Janecky, Los Alamos National Laboratory
Philip Jones, Los Alamos National Laboratory
Sylvain Jones, New Mexico Institute of Mining and Technology
Mia Kalish, New Mexico State University
Gloria Kindig
Amy Knowles, New Mexico Institute of Mining and Technology
Chris Koch, Google
Rita Kuo, New Mexico Institute of Mining and Technology
Laura Kurtzberg, Descartes Labs
Maximo Lazo, Central New Mexico Community College
Wing Lin, New Mexico Institute of Mining and Technology
Chris Lindemann, Eastern New Mexico University
Mya Longmire, New Mexico Institute of Mining and Technology

Joan Lucas, University of New Mexico-Los Alamos
Analyssa Martinez, New Mexico State University
Heidi Morning, Los Alamos National Laboratory
Rocky Navarrete
Stephen Pate, New Mexico State University
Veena Parboteeah, Eastern New Mexico University
Mark Petersen, Los Alamos National Laboratory
Jen Phifer, New Mexico Institute of Mining and Technology
Bruce Pitt, Mountain Vector Energy
David Price, New Mexico State University
Maureen Psaila-Dombrowski, Los Alamos National Laboratory
Jacquelyn Quesenberry
Ramya, New Mexico Institute of Mining and Technology
Eric Ried, Presbyterian Healthcare Services
Dana Roberson
Teri Roberts, Los Alamos National Laboratory/retired
Tom Robey, Gaia Environmental Sciences
Shaka Rucker, Central Consolidated School District
Albert Simon, Retired teacher
David Torres, Northern New Mexico College
Michael Trahan, Sandia National Laboratories
Geoff Valdez, PNM
Sarah Wall, Eastern New Mexico University
Chrysm Watson-Ross, University of New Mexico
Myhisha Webb, Pulpo Latin Network, Univision
Scott Wilson, Applied Technology Associates
Quincy Wolford, University of New Mexico
Simon Woodruff, Woodruff Scientific, Inc.
Janelle Zamie, Eastern New Mexico University



Do you want to become a supporter of the Supercomputing Challenge?
Please email us at consult@supercomputingchallenge.org for details.

A Novel Computational Tool to Inform Cost-Effective Nutrition Interventions in Sub-Saharan Africa

New Mexico Supercomputing Challenge

Final Report

April 3, 2019

Team 28

Los Alamos High School

Team Members

- Lillian Petersen
- Garyk Brixi

Project Mentor

- Daniela Moody

A Novel Computational Tool to Inform Cost-Effective Nutrition Interventions in Sub-Saharan Africa

Lillian Kay Petersen, Garyk Brix

1. Executive Summary

Malnutrition contributes to nearly half of childhood deaths, while treatment reaches a small fraction of those in need. Treatment delivery is hampered by costly ingredients and inefficient supply chains. Here we develop a three-component tool to inform acute malnutrition treatment interventions. First, we forecast the geospatial demand of acute malnutrition treatment using a machine learning algorithm. Second, we optimize low-cost recipes of specialized nutritious foods while meeting nutritional standards. Recipes were optimized for both international production and local production in 24 sub-Saharan African countries, and both achieved ingredient cost reductions of up to 60% compared to current recipes. Third, we model a supply chain of the optimal production and distribution of acute malnutrition treatment with both international and local factories while accounting for production and transportation costs. Three optimization scenarios were considered: using current factories to treat a set demand, building new factories to treat a set demand, and building new factories to maximize the number of cases treated on a budget. Our model suggests that optimized recipes could reduce total procurement costs by 25%, with additional reduction through optimizing supply logistics. The tool can assess relative location suitability for SNF production; compare the cost-effectiveness of different recipes; and identify cost drivers, such as the high import and export costs in sub-Saharan Africa. Used in conjunction, forecasting demand, optimizing recipes, and modelling efficient supply chains can better inform policy makers and donor organizations. Optimal supply chains could reach more children with life-saving treatment within existing budgets while supporting sustainable agriculture and future food security in developing countries.

2. Introduction

Child malnutrition prevalence remains unacceptably high in developing countries, causing nearly half of all childhood deaths. Globally, 16 million children under the age of five suffer from severe acute malnutrition (SAM) and 51 million suffer from moderate acute malnutrition (MAM) [1,2]. SAM and MAM are measures of wasting, or weight for height, and are defined as the percent of the population below two and three standard deviations from a healthy population's average.

Children suffering from SAM previously required hospitalization and treatment with therapeutic milk. Due to the difficulty and expense of hospitalizing large numbers of patients, intervention coverage was commonly under 10% and mortality remained high. Since 2000, over 70 countries have implemented community-based management of acute malnutrition (CMAM) programs using specialized nutritious foods (SNF) [3]. These specialized food packets are developed to meet correct macro and micronutrient compositions to help malnourished children under age five gain weight. Through outpatient care of children not suffering from complications, CMAM has improved treatment coverage and effectiveness. By using ready-to-use therapeutic foods (RUTF) to treat SAM and ready-to-use supplementary foods (RUSF) or Super Cereal Plus (SC+) to treat MAM, CMAM led to major improvements in the survival of children with acute malnutrition. CMAM improved treatment coverage from under 10% to over 70% of children suffering from SAM. [4,5].

However, current RUTF, RUSF, and SC+ remain costly, largely due to expensive ingredients, transportation, and misaligned policy. The standard RUF recipes use milk powder and peanut paste with vegetable oil, sugar, and micronutrient supplements [6]. Expensive milk powder accounts for over half the final cost [7], thus stressing developing countries' limited health budgets. It has become a major obstacle for scaling up treatment of SAM and MAM [8], preventing the meeting of basic nutrition needs in developing countries and hindering integration of SNF in national health programs [2,9]. Thus, there remains a major need to reduce the cost of RUF through reduced use of milk powder and peanuts, while still meeting all macro and micronutrient requirements.

Transport and logistics are also costly, involving international shipments from manufacturing sites mainly in France and the United States [10]. Insufficient lead time of malnutrition forecasts often leads to crucial delays in policy negotiations, planning, producing, and trucking of SNF. This delay sometimes forces aid organizations to rely on air transport of SNF during emergencies, bringing transportation costs from 10% to 39% of the total treatment cost [9,11]. Greater lead time of SAM and MAM forecasts could help increase the cost effectiveness of SNF logistics, allowing more people to receive treatment.

Recent research supports context-specific approaches to the treatment and prevention of malnutrition, including localizing SNF production close to malnourished patients [12]. Local formulae can improve acceptability, while local production may enhance availability and supply chain efficiency [12]. However, current RUTF recipes rely on ingredients that are locally unavailable (e.g. milk powder, peanuts, oil) and subject to import tariffs in sub-Saharan Africa. Domestic RUTF cost currently averages at \$50 per carton (150 servings), compared to \$44 for internationally procured RUTF [12], meaning local

production is currently less cost-effective. Using local ingredients could increase the cost effectiveness of local production for more cost effective treatment [13].

The cost of ingredients, logistics, and transport hinders developing countries from properly treating malnourished children. The current supply of SNF is insufficient, with RUTF meeting the needs of only 16% of children suffering from SAM [2]. This figure falls significantly short of the United Nation First 1000 Days goal to reduce malnutrition during the critical prenatal to 2-year-old time frame that “can cause irreversible damage to a child’s physical growth and brain development.” [14].

We aim to create a tool to inform policy makers on optimal production sites and distribution networks of SNF to best treat all malnourished children. By predicting future malnutrition prevalence, reducing ingredient cost, optimizing logistics, and localizing production, we aim to propose a more economic and sustainable system of acute malnutrition relief, aligned with long term development and food security goals.

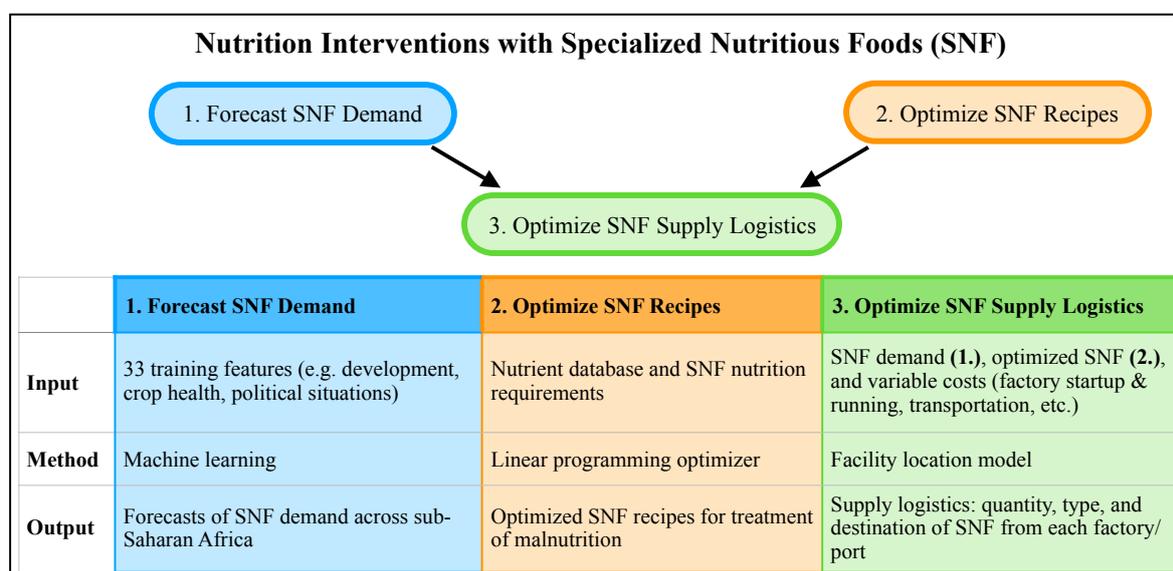


Figure 1. A flowchart detailing the three components of the computational tool.

3. Methods

The goal of this project is to inform SNF production decisions to sustainably increase treatment and reduce costs. It consists of three components: prediction of acute malnutrition prevalence (Section 3.1); optimization of SNF formulae (Section 3.2); and modeling the production and distribution of SNF (Section 3.3). The flowchart in Figure 1 illustrates an overview of all three parts of the project.

All of the python code for this manuscript was written by the authors and totalled to 2000 lines. The code can be found at https://github.com/lillianpetersen/Nutrition_Intervention.

3.1. Forecasting SNF Demand

The first step in this project is forecasting demand of SNF to feed into the supply chain model (Figure 1). Inaccurate or untimely forecasts of malnutrition prevalence often lead to crucial delays

in planning and production of SNF, which can force aid organizations to use air transportation and dramatically increase the cost of treatment.

We used a machine learning algorithm to predict future geospatial malnutrition prevalence across sub-Saharan Africa based on 33 training features (Table 1). After performing a training and testing trial to validate the accuracy of the predictions, we predicted malnutrition prevalence across sub-Saharan Africa to year 2021.

3.1.1. Malnutrition Data and Training Features

The malnutrition data that we used as ground-truth synthesizes weight, height, and age data from numerous surveys across sub-Saharan Africa [15]. The released data set includes gridded acute malnutrition prevalence from 2000-2015 at a 5km resolution (Figure 2).

We assembled a training data set for machine learning of malnutrition prevalence. The 33 training variables may be split into five categories: development, economics, political situations, climate, and crop health. Table 1 lists the training features and Figure 3 highlights some examples.

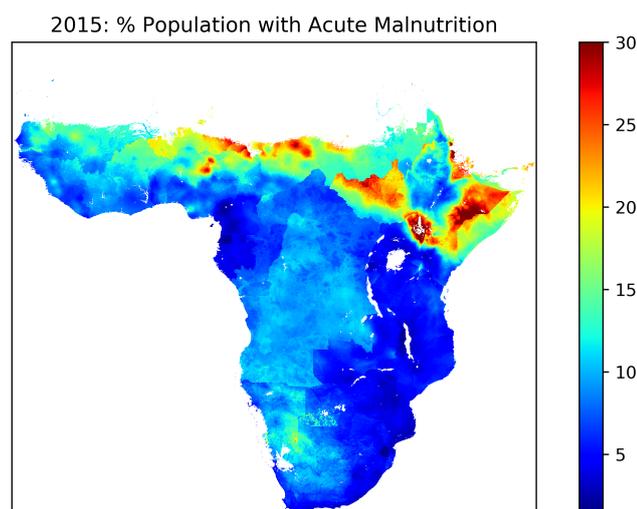


Figure 2. Acute malnutrition prevalence across sub-Saharan Africa in 2015. This data is used as ground-truth in our model. Malnutrition prevalence is highest in South Sudan, Ethiopia, Niger, Nigeria, and Kenya.

3.1.2. Scenarios using the Logistic Model

Before entering the features into the model, the data had to be processed into the correct format. Many of the variables came as tiff files and were already gridded. These variables were interpolated to the same grid as the malnutrition prevalence data using a bivariate spline approximation technique. Other data was retrieved by country (e.g. World Bank data). We used a national identifier grid [32] to overlay these variables onto a grid.

A few datasets came as latitude/longitude points or shapefiles (i.e. coastlines and fatalities from conflicts). We interpolated coastlines into a raster dataset of distance from coasts by calculating the

Table 1. Training features used in machine learning algorithm to predict acute malnutrition.

| Category | Training Features |
|----------------------|---|
| Development | <ol style="list-style-type: none"> 1. Female education, years of attainment, gridded [16] 2. Percent of school-aged children in school [17] 3. Percent of population with access to electricity [17] 4. Percent of females with a secondary school education, national [17] 5. Human Development index [18] 6. Gridded population [19] 7. Travel time to the nearest urban center [20,21] 8. Low degree and high degree of settlements [19,22] 9. Built up land cover types [22] |
| Economics | <ol style="list-style-type: none"> 1. Agriculture as a percentage of GDP [17] 2. Net official development assistance (ODA) per capita [17] 3. Gridded subnational estimates of GDP PPP per capita [18] 4. Imports per capita [17] |
| Political Situations | <ol style="list-style-type: none"> 1. Political stability and absence of violence [17,23] 2. Government effectiveness [17,23] 3. Conflicts and fatalities from conflicts [24] |
| Climate | <ol style="list-style-type: none"> 1. AVHRR-derived forest cover [25] 2. Distance to coasts and inland coasts [26] 3. Elevation [27] 4. Elevation roughness [27] 5. AVHRR-derived bare ground [25] |
| Crop Health | <ol style="list-style-type: none"> 1. Mean annual precip [28] 2. Crop yield [17] 3. Crop production per capita [29] 4. Diversity of Crop Systems [30] 5. AVHRR-derived NDVI [25] 6. Irrigated area (area actually irrigated) [31] 7. Irrigated area (area equipped for irrigation) [31] |

distance from the nearest coast for every pixel. To convert the conflict points onto a grid, we used a market potentials index. The market potentials equation applied to conflicts is

$$MP_i = \sum_{j=0}^J \frac{\text{fatalities}_j}{\text{dist}_{i \rightarrow j}^\theta} \quad (1)$$

where i is the current pixel, j is the latitude/longitude point of the conflict, and fatalities is the fatalities at point j . The power θ in this calculation is chosen to be 1.2. In this way the conflicts database was converted into a raster dataset. All of the training features combined made a total of 13 gigabytes of input data for the malnutrition forecasting component.

3.1.3. Machine Learning and Predictions

Next we trained a random forest regressor, a machine learning algorithm in the python library scikit learn, on the features to predict malnutrition prevalence. The data was split into a training set of 80% and a testing set of 20%. To avoid overfitting by spatial correlation, the testing data was removed in boxes across sub-Saharan Africa. The model was trained on the previous year's indicators (e.g. 2015 prevalence was trained on 2014 features). We then validated the accuracy of the predictions by

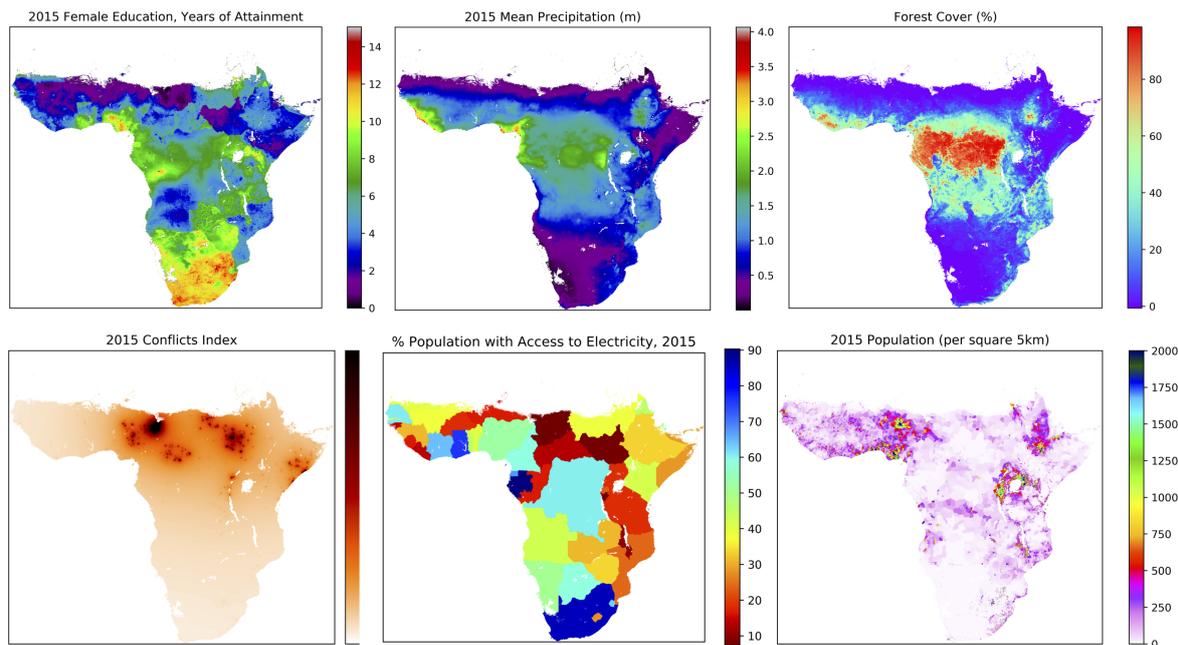


Figure 3. An example of six of the 33 training variables for prediction of malnutrition. The year 2015 is shown, but data was evaluated for each year, 1999–2020.

comparing the predicted malnutrition prevalence to the testing set. Validation data is presented in the results Section 4.1.

After validating the model, we predicted malnutrition prevalence for 2016–2021. When a training variable was missing for one of these years, we extended a trend from the previous years. The caseload of malnutrition was calculated by multiplying the prevalence by the population under 5 for each grid cell, and from the caseload the expected demand of SNF may be calculated (see Section 3.3.1).

3.2. Optimizing SNF Recipes

For the second component of the computational tool, we created optimized formulae for treatment of acute malnutrition by meeting all nutritional requirements while minimizing cost (Figure 4). First, we created optimized recipes using international ingredient costs. Then we created optimized recipes for 24 sub-Saharan African countries using local ingredients and prices. The optimized recipes are then fed into the third component of this project, the supply chain model of SNF production and distribution (Section 3.3).

3.2.1. Linear Programming Tool

Building on Brixi (2018) [33], we created a linear programming (LP) tool to optimize for low cost and local contexts in compliance with applicable nutrient and formulae composition standards. The

Optimizer 1. Specialized Nutritious Foods

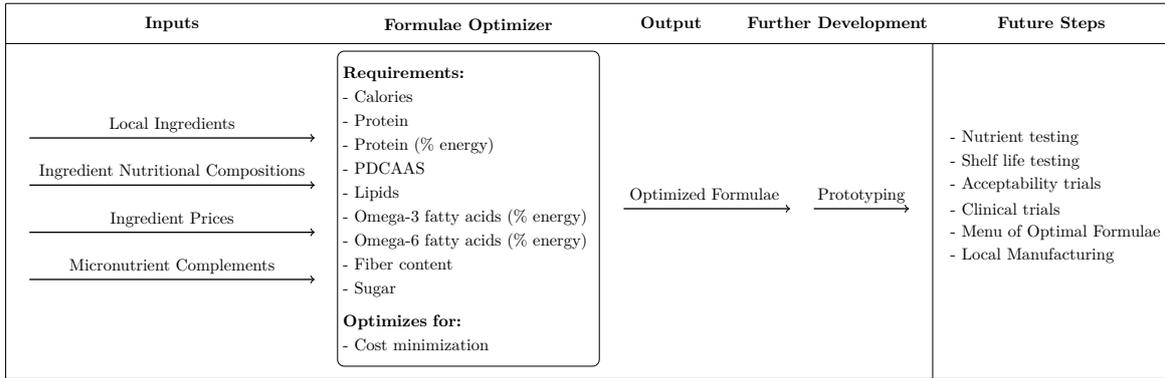


Figure 4. A flowchart of the inputs, constraints, and outputs of the SNF optimizer.

tool minimizes cost, specifying the optimal quantity of each ingredient for low cost while meeting established nutritional requirements and other constraints. The tool's linear objective function is

$$Y = \sum_{i=0}^n b_i \cdot B_i \quad (2)$$

where Y is the total cost of ingredients, b_i is the cost per gram of ingredient i , and B_i is the amount of ingredient i in grams.

Compared to previous optimization efforts, our tool includes an expanded range of ingredients and constraints. The optimization tool lets users set the macro and micronutrient constraints according to patient conditions. Protein Digestibility Corrected Amino Acid Score (PDCAAS), micronutrient supplements, and water efficiency optimization constraints are included.

PDCAAS is automatically calculated and constrained in order to ensure protein quality without the need for costly dairy ingredients. This is expressed as

$$Q_a = \sum_{i=0}^n C_i a_i d_i \quad (3)$$

where Q_a is total quantity of essential amino acid a , C_i is quantity of ingredient i , a_i is quantity of amino acid a per gram of ingredient i , and d_i is protein digestibility factor of ingredient i . The tool assures that protein quality is met using the criteria

$$\frac{Q_a}{P} \geq S g_a \quad (4)$$

where P is total protein quantity in the formula, S is the goal PDCAAS score expressed as a decimal, and g_a is goal quantity of amino acid a per gram of reference protein. The tool can automatically include amino acid supplements as ingredients to meet this goal PDCAAS. Furthermore, the tool is able to modify formulae in response to shifts in prices and availability.

For water efficiency, the tool limits the water footprint of ingredients using UNESCO-IHE water footprint data to facilitate local production in water scarce regions [34].

The tool's accuracy was validated using the ingredient composition of current peanut RUTF and previously published dairy-free RUTF [35,36]. The nutritional values calculated by our tool were compared with previously published laboratory analysis.

3.2.2. Optimizing SNF for International Production

SNF were developed using the optimization tool. We set constraints according to standards for each SNF product. To ensure satisfactory PDCAAS, we used measured amino acid profiles from previous laboratory analysis of RUTF, and estimated amino acid profiles based on composition of RUSF and SC+, alongside technical specifications. The constraints, shown in Table 2, reflect the current standards and practice, as well as clinical trials of recovery from SAM and MAM.

At this stage, we used the tool to optimize alternative RUTF, RUSF, and SC+ formulae for international production using average international commodity prices over the past five years, adjusted for processing.

3.2.3. Prototyping and Testing Optimized SNF

Prototyping and testing were conducted in collaboration with Valid Nutrition, a research NGO, and partner facility in Kenya. Optimized SC+ formulae were manufactured in Kenya with locally sourced soybeans and corn flour and imported sugar, oil, Ajinomoto amino acid supplements, and micronutrient premix. Samples were sent to SGS lab Nairobi for nutrient analysis.

3.2.4. Optimizing SNF for Local Production

After confirming the accuracy of the LP tool through prototyping, we optimized recipes in 24 sub-Saharan African countries with available local ingredient price data. For each country, we optimized RUTF, RUSF, and SC+ formulae using locally grown crops, and compared our optimized formulae with current products for nutrition, ingredient cost, and water efficiency. For the comparison, the ingredient cost and water efficiency value of current formulae were calculated using the same databases as the optimized formulae to ensure consistency.

3.3. *Optimizing SNF Supply Logistics*

Next, we created a computer model to optimize SNF production and distribution networks in sub-Saharan Africa based on the optimized local and international SNF formulae presented in the previous section (Section 3.2). The optimizer treats the demand of acute malnutrition while accounting for associated costs, including ingredients, production, and transportation of SNF. After minimizing cost, the tool returns the optimal placement and capacities of factories and ports; the type, quantity, and destination of SNF from each port and factory; and the total procurement cost (Figure 5).

We optimized supply logistics for three scenarios that reflect the possible uses of the tool:

Optimizer 2. Acute Malnutrition Treatment Supply Chain

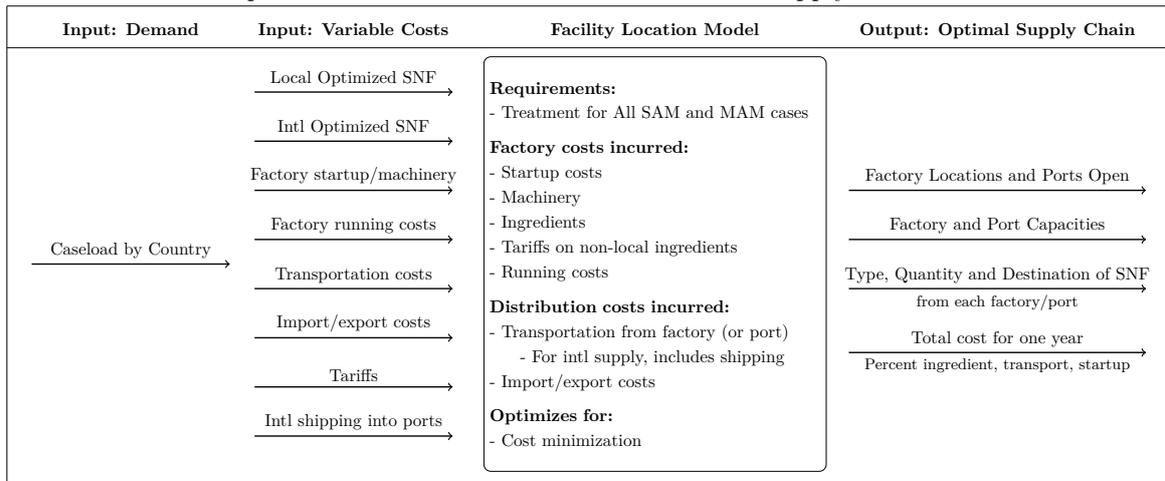


Figure 5. A flowchart of the inputs, costs, constraints, and outputs of the supply chain optimizer.

1. Current factories and prices; cost is optimized to treat all demand;
2. New factories can be established; cost is optimized to treat all demand;
3. New factories can be established; number of cases is optimized on a budget

3.3.1. Quantifying SNF Demand

The first step is to calculate the SNF demand for treatment of SAM and MAM for each country. We calculated the annual treatment demand to quantify demand by country in the model. Prevalence of severe and moderate acute malnutrition was obtained for 43 mainland sub-Saharan African countries using national surveys [2]. Although survey data is currently used for the demand, the facility location model is built to accept future demand forecasts from Section 3.1.

In order to account for the known underestimation of caseload based on prevalence, We adjusted prevalence using the incidence correction factor K [37–39].

$$\text{Incidence} = \text{Prevalence} \cdot K \quad (5)$$

$$K = \frac{\text{duration over which prevalence was estimated}}{\text{avg duration of untreated disease}} \quad (6)$$

$$\text{For SAM: } K = \frac{365 \text{ days}}{45 \text{ days}} \quad (7)$$

For both SAM and MAM, survey data estimated prevalence over a one year period. Using current estimates for the duration of untreated malnutrition, we calculated the incidence correction factor K for SAM and MAM [37–39]. The incidence of malnutrition per country was then multiplied by the

quantity of SNF needed per case to estimate the annual demand for the respective treatment product by country [39–41]:

$$\text{Annual treatment demand} = \text{Incidence} \times \text{Duration of treatment} \times \text{Treatment Dosage} \quad (8)$$

The annual SNF demand was then increased by 10% to account for possible underestimation based on previous findings [37]. Compared to SAM, MAM requires a lower amount of treatment, which reflects the lower incidence correction factor as well as lower amount of treatment needed per case. Figure 6 displays the final by-country demand for treatment used in the model, where a sachet is a single serving packet.

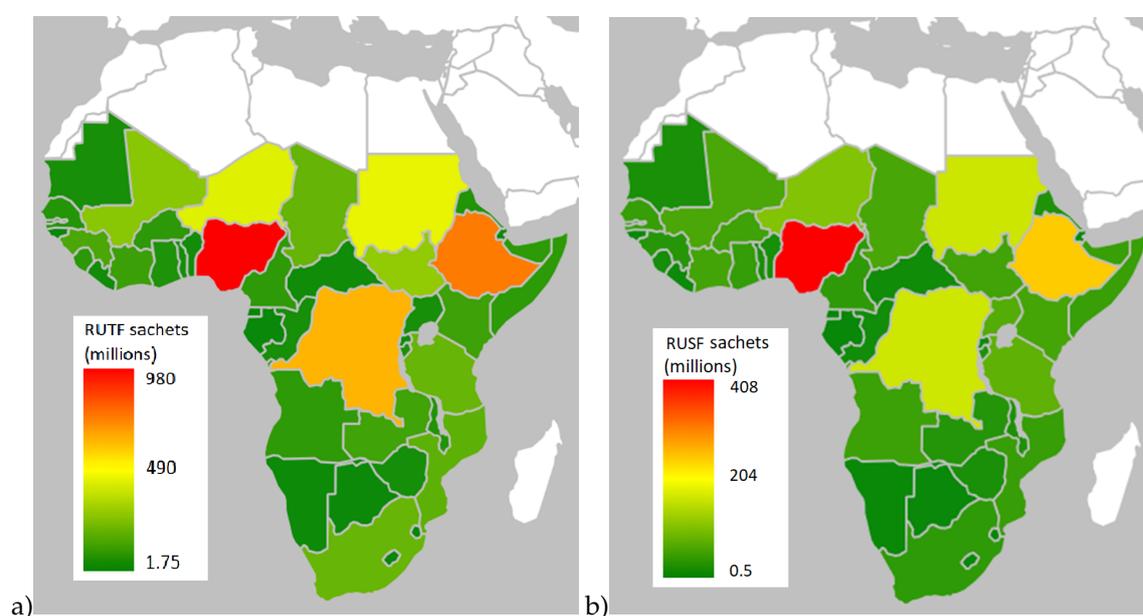


Figure 6. The estimated demand for treatment of severe (a) and moderate (b) acute malnutrition, based on UNICEF surveys [2].

3.3.2. Variable Costs

The optimizer is dependent on variable costs of SNF production and distribution. These include: ingredient costs of local and international optimized SNF, factory startup and machinery costs, factory running costs, transportation costs, import and export costs, tariffs on non-local ingredients, and shipping costs from international suppliers. These variable costs are sourced from literature, UNICEF reports, and personal communications with Valid Nutrition.

The model includes local production cost information to quantify the fixed and operating costs. Interviews with SNF production experts indicated that ingredient cost is the most significant and most variable operating cost item, which should influence SNF facility location [42]. The ingredient cost came from the optimized RUTF, RUSF, and SC+ formulae for each country (explained in Section 3.2). Fixed costs (e.g. factory startup, purchasing and upgrading machinery) were distributed over a five year period. Required equipment include cleaning/preparation machines and the mixing/bagging

machines. Fixed costs were estimated based on literature, supplier catalogues of machinery, and personal communication with SNF manufacturers [7,43].

Transportation costs were estimated between each sub-Saharan African capital. On-road distances were calculated using the python Google Maps API. This distance was multiplied by estimated regional costs, to obtain the cost of transporting one tonne of packaged product between each capital [44,45]. Import and export costs per tonne of product were then added to the trucking cost, as document and border compliance costs in sub-Saharan are extremely high [17]. Import tariffs on non-local ingredients were also added to the model, and were estimated based on published tariff data [46]. Shipping costs on internationally-produced SNF were calculated to each major port in sub-Saharan Africa, and were based on published shipping costs [2].

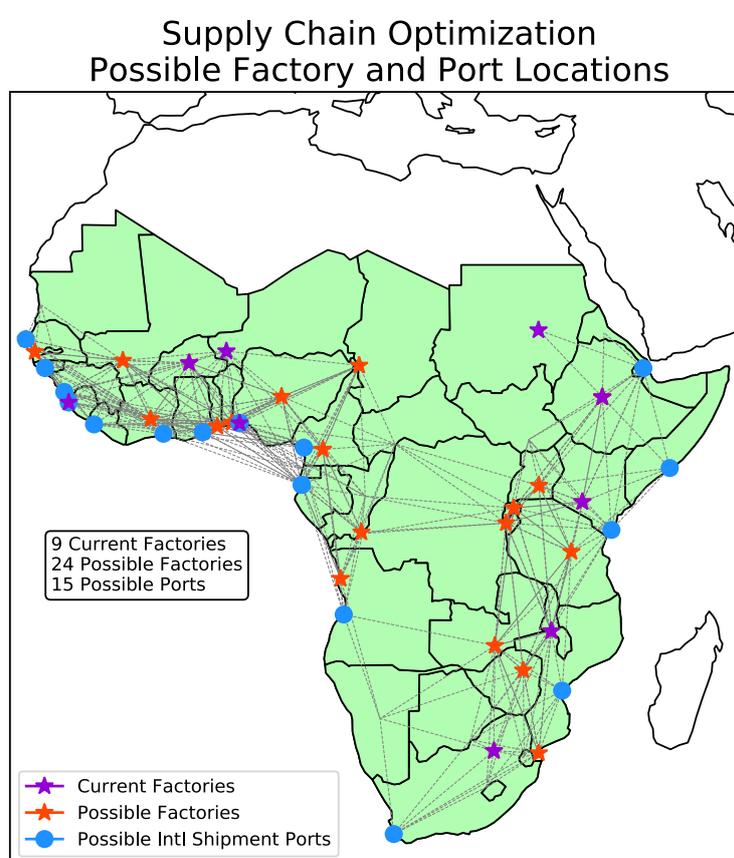


Figure 7. A skeleton of the supply chain optimizer. Every purple star is a current factory location, every red star is a possible factory location (in every country with sufficient price data), and every blue circle is a possible port for international shipments.

3.3.3. Supply Chain Model

A skeleton of the facility model can be seen in Figure 7. The model starts with the current factories and can build new ones. The cost of local production in African cities (stars) include factory start-up (for new factories), machinery, and running costs, plus the cost of producing the SNF in that country (includes tariffs on non-local ingredients). When treatment is shipped from that factory to neighboring

capitals, the costs incurred include trucking, import, and export costs. Off-continent production costs include manufacturing internationally-optimized SNF, shipping into one of the major African ports (blue dots in Figure 7), and trucking to the final destination with import/export costs across borders.

In the end, the optimizer outputs include the factory and port locations and capacities, as well as the type, quantity, and destination of SNF from each factory and port, in order to either treat all malnourished children on the lowest budget or treat the highest number of children on a set budget.

3.3.4. Scenarios Using the Logistic Model

The different supply chain possibilities were ran to reflect the possible use of this supply chain tool. The model can be set to optimize the cost while meeting a demand or to maximize the cases treated on a set budget. By modelling to treat the full caseload, the model is able to identify the optimal supplier of treatment for each country, while maximizing cases treated allows users to see where interventions are least costly. The user can also set whether the model can build new factories or just uses the current ones. Within the scenarios, three SNF recipe levels were used to determine the cost reduction that could be achieved through optimization of recipes (Current recipe, local recipes optimized while international remains current, and all optimized recipes, Section 3.2). The total scenarios are:

1. Current factories and prices; cost is optimized to treat the full caseload;
 - Current prices used [47]
2. New factories can be established; cost is optimized to treat the full caseload;
 - Current recipes
 - Local Optimized
 - All Optimized
3. New factories can be established; number of cases is optimized on a budget
 - Current recipes
 - Local Optimized
 - All Optimized

High-quality local ingredients are sometimes not available locally, and must be imported. For this reason, estimated tariffs were added to the cost of imported ingredients.

3.3.5. Parameter Study

To identify barriers in increasing access to treatment, as well as to account for the inaccuracy and lack of data in sub-Saharan Africa, we ran a parameter study using the logistic model. The logistic model was run for approximately 700 cases, varying each of the following parameters independently: trucking costs, sea shipping costs, border and document compliance costs, factory start-up costs, tariffs, and budget (for the budget scenario).

4. Results

We built a tool to optimize treatment of acute malnutrition across sub-Saharan Africa. In this section we present the results from each of the three components, including forecasting SNF demand

(Section 4.1); optimizing SNF recipes (Section 4.2), and optimizing the supply chain of SNF distribution (Section 4.3).

4.1. Forecasting SNF Demand

The model predicted malnutrition prevalence with minimal error. The random forest regressor was trained on 80% of the data for each year, 2000 through 2015. It then predicted the 20% of the data it was not trained on. Figure 8a shows the training data for 2015 and Figure 8b shows the same data with the boxes filled in with predictions. The predicted and actual malnutrition prevalence across 2000–2015 had a correlation of 0.95 with an average difference of 0.83% prevalence (Figure 8c).

After validating the model, we predicted malnutrition prevalence from 2016 through 2021 (Figure 8d). It was found that the geospatial prevalence of malnutrition will likely remain the same over the following years, with the highest prevalence remaining in Ethiopia, South Sudan, Niger, and Kenya. Malnutrition prevalence forecasts show a decreasing trend in the future. Malnutrition prevalence has been decreasing since 2000, and our predictions suggest that this trend will continue. However, because of the growing population in sub-Saharan Africa, caseload of malnutrition and demand for SNF is expected to grow in the upcoming years.

Through the random forest regressor, we also find which training variables have the greatest influence on malnutrition. Ordered by importance, the top ten features are: (1) female education; (2) mean annual precipitation; (3) forest cover; (4) percent of school-aged children in school; (5) political stability and absence of violence; (6) crop yield; (7) crop production per capita; (8) access to electricity; (9) distance to coasts; (10) elevation.

Lack of education, especially female education, is highly correlated to malnutrition. Education has been shown to lead to better nutritional habits and food diversity, increased development, and decreased poverty. From this result, we can conclude that aid organizations should have a focus on education to improve long-term measures of living standards.

The environment also affects malnutrition prevalence. Forested areas are less vulnerable to acute malnutrition than semi-arid regions or deserts, likely because dryer regions are more susceptible to seasonal variability and droughts. Proximity to coastlines may temper malnutrition by providing easier access to fisheries and overseas markets for alternative food sources.

Malnutrition prevalence is also dependent on crop yield and crop production per capita, which indicate both the level of agricultural technology and amount of food available. As the population of Africa is projected to grow throughout this century, aid organizations should focus on supplying local farmers with better agricultural technology to increase food availability. Political instability and violence also may obstruct markets to limit access to food, thus increasing malnutrition.

4.2. Optimizing SNF Recipes

The linear programming tool successfully optimized SNF formulae and generated cost-effective RUTF, RUSF, and SC+ suitable for international and local production in sub-Saharan Africa. Compared

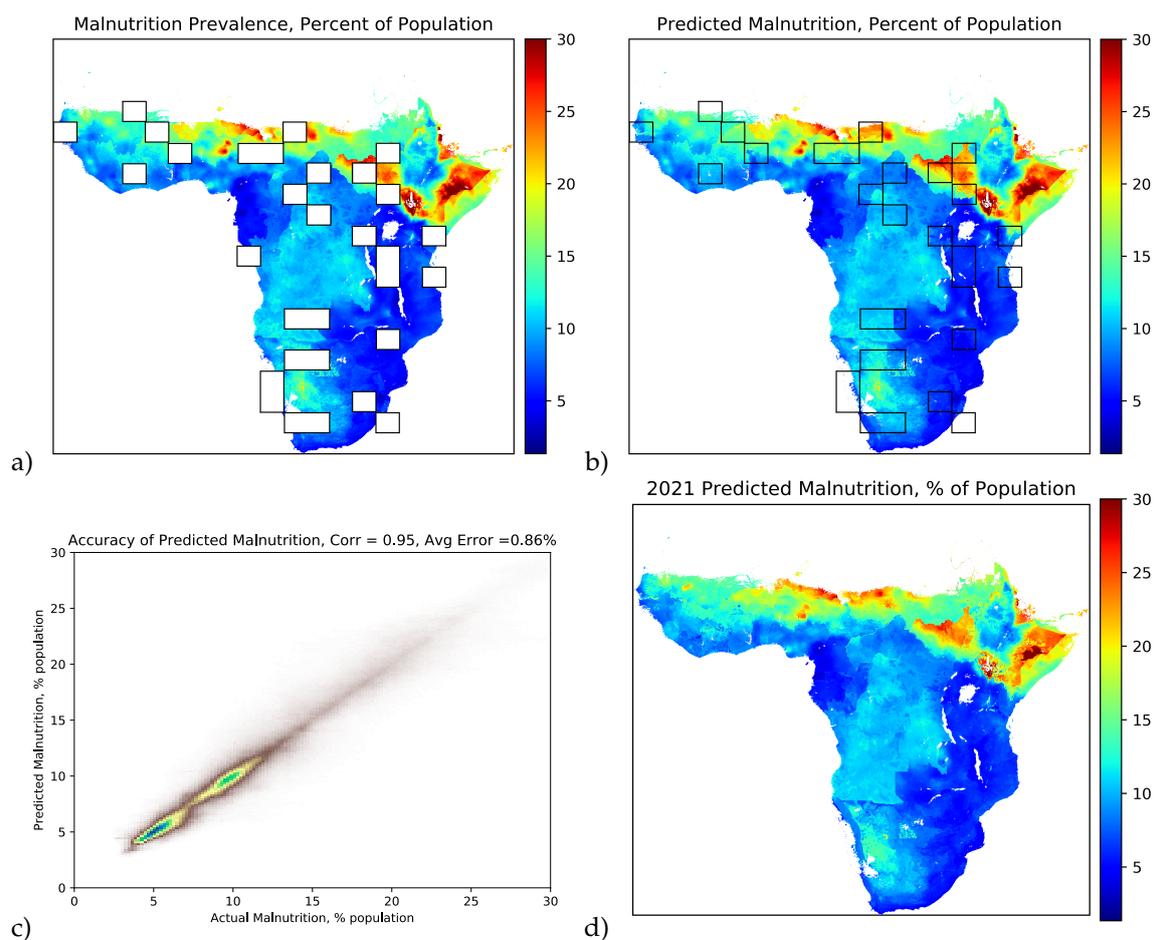


Figure 8. The malnutrition prevalence training data (a), and the same data with the boxes filled in with predictions (b), shown for 2015. The difference between the actual and predicted malnutrition prevalence (c) has a correlation of 0.95 with an average difference of 0.86% prevalence. After the model was trained, malnutrition prevalence was predicted to 2021 (d).

to current products, the optimized formulae have significantly lower ingredient cost and water footprint. The primary driver in ingredient cost was low-cost protein.

4.2.1. Linear Programming Tool

The LP tool generated cost-effective RUTF, RUSF, SC+ formulae meeting nutritional and technical specifications. The tool was found to accurately calculate recipe nutrition values: when we calculated the nutritional values of current recipes (peanut and dairy free RUTF), caloric value and PDCAAS were within 1% of actual values and percent energy from protein was within 5%. The amino acid profile (non digestibility corrected) was within 5% of the actual value, and within 1% for the limiting amino acid.

4.2.2. Optimizing SNF for International Production

Here we present the optimized SNF based on international commodity prices. The optimized RUTF, RUSF, and SC+ all include corn flour, palm or soy oil, soybean, sugar, and small quantities of leucine and lysine supplements (amino acids). For each formula, the tool calculated a complementary micronutrient premix and included the cost of the premix in ingredient cost. Following prototyping, adjustments may be made to the micronutrient premix to ensure goal nutrition regardless of losses during processing and storage.

As shown in Table 2, the proposed optimized formulae within nutritional requirements, lower ingredient cost, and lower water footprints compared to the current practice [48,49]. The LP tool created formulae with PDCAAS equivalent to current recipes by automatically balancing proteins with complementary quantities of the essential amino acids.

Table 2. Comparisons between current and optimized RUTF, RUSF, and SC+.

| Formulae | Ready-to-Use Therapeutic Food | | Ready-to-Use Supplemental Food | | Super Cereal Plus (SC+) | |
|---|--|---|--|--|--|--|
| | Current (Plumpy'nut) | Optimized | Current (Plumpy'nut) | Optimized | Current | Optimized |
| Composition (g) *[descriptors below] | peanut sugar milk powder palm oil micronutrient premix | corn flour 28.8 palm oil 24.6 soybean 22.5 sugar 15 sorghum 7 leucine 0.199 lysine 0.170 valine 0.058 adjusted premix | peanut sugar milk powder palm oil micronutrient premix | corn flour 34.2 soybean 19.4 palm oil 23.5 sugar 15 sorghum 7 adjusted premix | corn flour soybeans milk powder Sugar soy oil micronutrient premix | corn flour 62.8 soybean 26 sugar 9 palm oil 1.8 leucine 0.0722 lysine 0.0353 adjusted premix |
| Total weight (g) | 100 | 100 | 100 | 100 | 100 | 100 |
| Total calories | 520–560 | 520 | 510–560 | 510 | 410–430 | 410 |
| Total protein (g) | 13–16 | 13 | 11–16 | 11 | ≥16 | 16 |
| Total fat (g) | 26–36 | 30.85 | 26–36 | 29.39 | ≥9 | 10.10 |
| Fiber (g) | <5 | 4.53 | <5 | 4.7 | 6 | 2 |
| Omega 6 (g) | 2.03–6.75 | 5.29 | 2.03–6.75 | 6.10 | 4.73 | 4.04 |
| PDCAAS | required: ≥95 actual: 106 | 106 | required: ≥70 actual: 79 | 86 | required: ≥70 actual: 87 | 87 |
| Total int'l ingredient cost (USD/100g) | 0.126 | 0.04956 | 0.118 | 0.0422 | 0.04487 | 0.03116 |
| Water footprint (gallons/mt) | 11,551 | 2,762 | 10,123 | 2,675 | 4,444 | 1,717 |

Notes: Prices calculated using FAO, UNCTAD and GEM data. Water footprint calculated using UNESCO-IHE report [34].

* Descriptors for USDA nutrient database: soybean: (16111, soybeans, mature, dry roasted. roasted); palm oil: (04055, oil, palm); maize: (20017, corn flour, enriched, white).

The optimized SNF have a lower total international ingredient cost than current SNF. Optimized RUTF and RUSF reduce ingredient cost by 60% and optimized SC+ by 30%. Water footprint was also reduced significantly in the optimized formulae, largely due to the removal of milk powder.

Table 3. Results from the laboratory analysis of the prototyped SC+. The calculated values were very similar to the actual values, and all parameters were in compliance with WFP standards.

| Parameter | Calculated Value | Actual Value | Compliance with WFP Standards? |
|--------------------|------------------|--------------|--------------------------------|
| Moisture (%) | 3.1 | 2.75 | Yes |
| Fat (%) | 10.1 | 11.17 | Yes |
| Fiber (%) | 2 | 1.98 | Yes |
| Protein (%) | 15.99 | 16.57 | Yes |
| Ash (%) | 3 | 2.72 | Yes |
| Energy (kcal/100g) | 410 | 424 | Yes |

4.2.3. Prototyping and Testing Optimized SNF

The optimized international SC+ recipe was prototyped and analyzed in Kenya in collaboration with Valid Nutrition. The analysis confirmed recipe compliance with WFP standards for nutrition, composition, and texture. This analysis confirmed the LP tool's suitability for developing new recipes. Table 3 shows results from the laboratory analysis.

4.2.4. Optimizing SNF for Local Production

After validating the accuracy of the international recipe, we optimized RUTF, RUSF, and SC+ in 24 sub-Saharan African countries with local commodity prices while meeting WFP standards. All optimized recipes are more cost-effective than the current recipe for RUTF and RUSF (Figure 9).

For both RUTF and RUSF, the local optimized recipes ranged between about 60% and 3% ingredient cost reduction in South Africa and DR Congo, respectively, compared to the current international recipe (Figure 9). Locally optimized formulae likely reach the lowest ingredient cost in South Africa due to the lower cost of vegetable oils there.

For SC+, only two-thirds of countries had lower ingredient cost for locally optimized SC+ compared to the current international formula. This reflects the relatively low cost of current SC+. Again, the greatest cost reduction (25%) was achieved in South Africa. Due to high daily dose of 200g for SC+, the relative cost of one serving (92g) of optimized RUSF is lower than that for one serving (200g) of optimized SC+, making optimized RUSF the more cost-effective treatment for MAM.

Cost reduction is mainly facilitated by cost-effective protein quality. The compositions of each RUTF and RUSF recipe can be found in Figure 10.

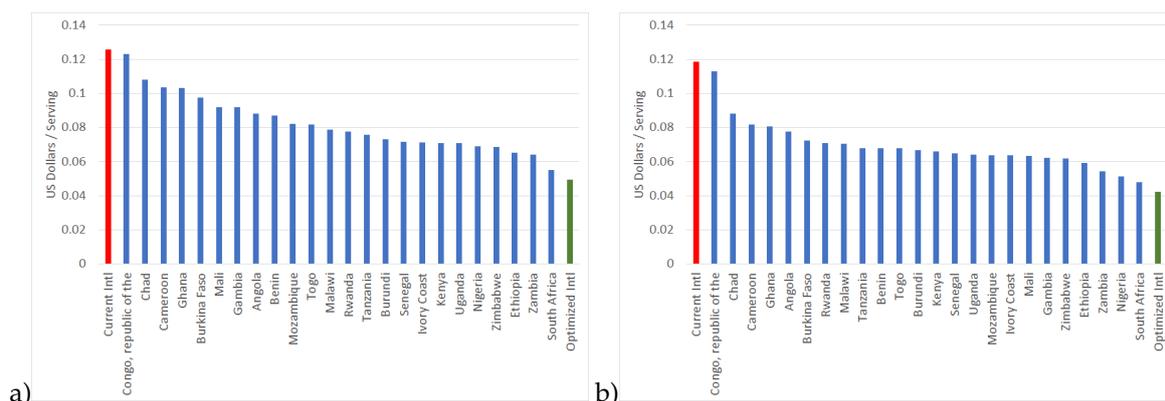


Figure 9. Ingredient cost comparison for optimized RUTF (a) and RUSF (b) arranged by cost.

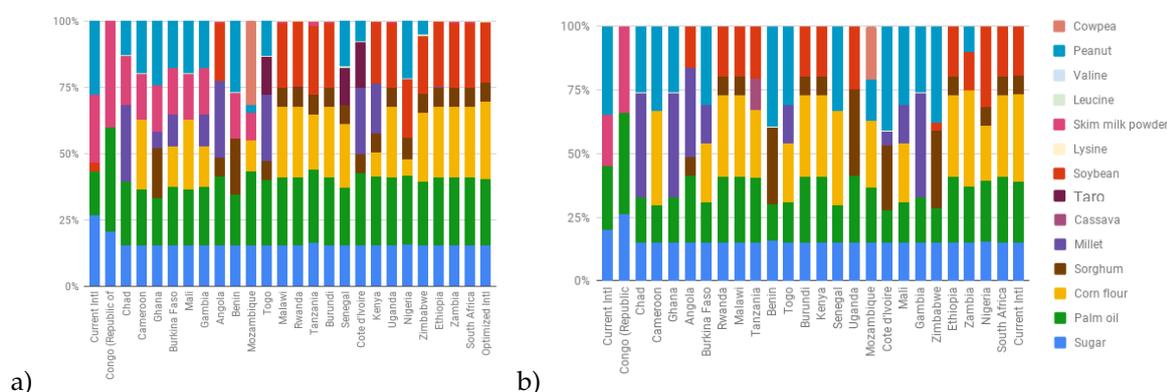


Figure 10. Ingredient composition comparison for local optimized RUTF (a) and RUSF (b) by cost.

4.3. Optimizing SNF Supply Logistics

The supply chain model effectively distributed acute malnutrition treatment through an optimized network of factories and ports. Here we will present the optimal supply logistics using only current factories to meet all demand (Section 4.3.1), building new factories to meet all demand (Section 4.3.2), and building new factories to treat the highest number of children on a budget (Section 4.3.3).

4.3.1. Optimizing Cost while Meeting All Demand: Current Factories

The supply chain model used existing factories in Africa and internationally in order to meet the current caseload. The model supplied 40% of the total demand from current factories in South Africa, Kenya, and Sudan, relying on international import to meet the remaining demand for treatment (Figure 11).

Although the model had the potential to use 9 factories without any startup cost, it only procured treatment from two because of the current high cost of local SNF. Local SNF remains more expensive because the countries must import expensive ingredients, such as milk powder or peanuts, and pay high tariffs on them. Because South Africa has relatively low tariffs and can produce some of these ingredients locally, it is much cheaper than the other local producers and supplies several countries with treatment.

To evaluate how the supply chain would change if local production became cheaper, we reduced the cost of local SNF (excluding South Africa) by 5% intervals. Even at a 5% reduction, 56% of SNF is procured locally with 5 local factories (Figure 11). This exhibits how using local ingredients or reducing tariffs could make local production economically competitive with international procurement.

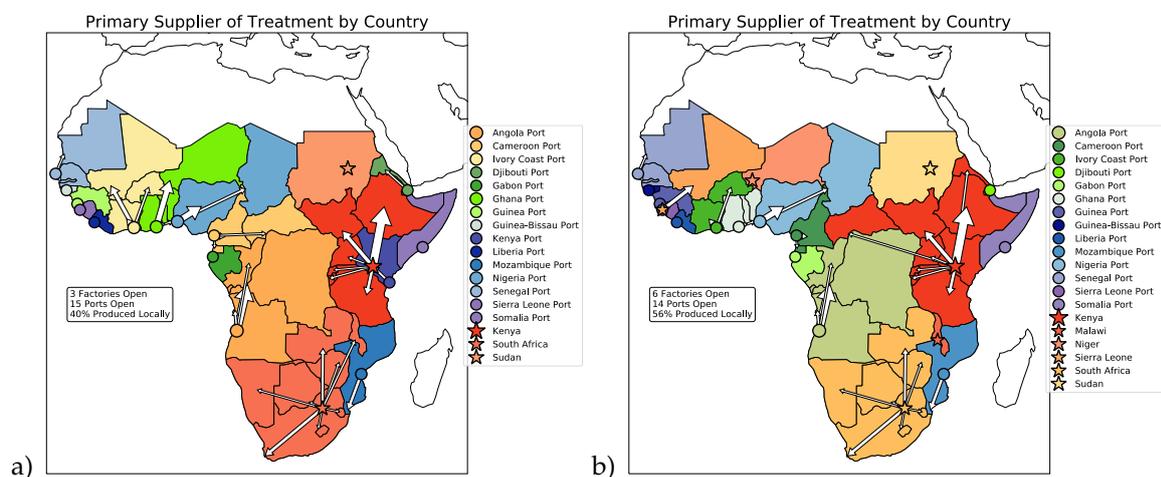


Figure 11. The primary supplier of SAM and MAM treatment using only current factories at the current cost (a) and when local costs are reduced by 5%. The arrows are scaled by the amount of treatment being shipped.

4.3.2. Optimizing Cost while Meeting All Demand: Building New Factories

The supply chain model effectively met the demand for acute malnutrition treatment through an optimized network of factories and ports. We optimized the supply chain for three scenarios: current recipe, local optimized recipes, and all optimized recipes. First, we present the results from the all optimized scenario.

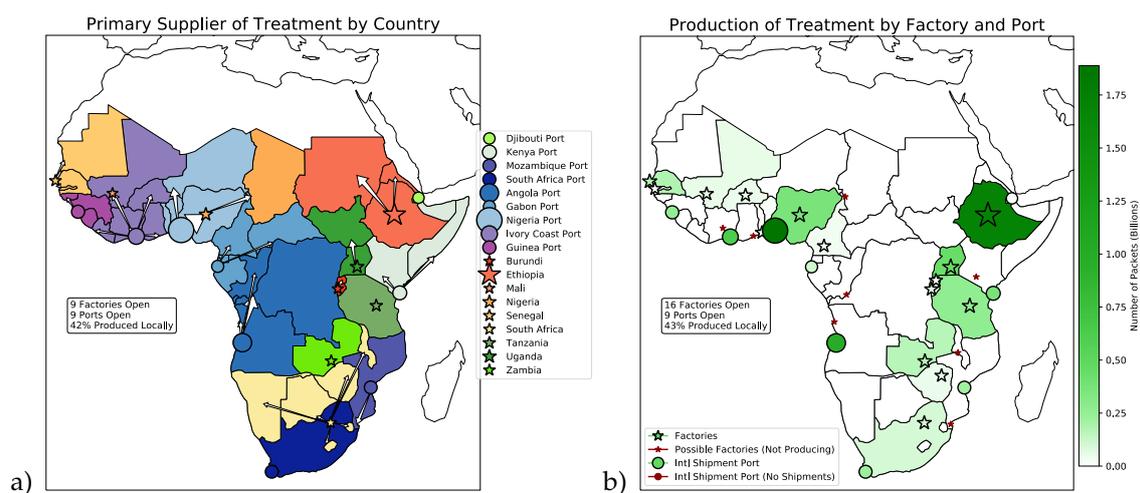


Figure 12. The primary supplier of SAM and MAM treatment (a) when all recipes are optimized. The arrows are scaled by the amount of treatment being shipped. About half of both products are produced locally. (b) shows the number of packets of RUSF produced or imported per year.

Figure 12a shows the primary supplier of RUF by country when all recipes are optimized, and Figure 12b shows the relative amount produced at each factory or port. About half of the treatment is produced locally, with nine factories and nine ports open. Factories and ports typically supply between one and three countries, and supply the greatest quantity of treatments domestically.

It is notable that most of the local producers of both SAM and MAM treatment are placed in inland countries. Because of the high startup and factory running costs in sub-Saharan Africa, it is more cost-effective to import treatment on the coasts. However, the high trucking cost causes local production to be more cost-effective farther inland.

To examine how each variable cost affects the supply chain, we ran a parameter study in which we changed the relative costs of startup, import/export, trucking, shipping, and tariffs. For example, we altered the startup costs from 20% to 200% of today's prices, with 20% intervals. Figure 13 shows the optimal amounts of treatment provided by each port and factory as startup cost varies. At today's startup prices (100%), about half of the treatment for both SAM and MAM is produced locally (shown in warm colors). As startup cost increases, the local production cannot compete with the cheaper international product. If countries are able to lower their high startup costs, local production would become more economically viable.

The parameter study may also help identify countries suitable for long-term investment, despite possible changes in variable costs. For example, when examining Figure 13, we can see that Ethiopia remains a major producer of SNF even with extremely high startup costs. The amount of SNF produced in Ethiopia remains fairly constant regardless of changes in any of the 5 parameters, suggesting it to be an optimal supplier regardless of exogenous changes.

Next, the logistical model optimized SNF supply chains for all three recipe scenarios: current, only local optimized, and all optimized recipes. From these calculations, we may evaluate the effect of optimized SNF formulae. Optimized SNF reduces the total modelled cost by 25% compared to the current recipe (Figure 14), reinforcing the importance of low ingredient cost. Interestingly, the total modelled cost is similar between only local optimized and all optimized recipes, suggesting the feasibility of local manufacturing.

The total cost of procurement varies when changing the parameters of startup, import/export, trucking, and shipping cost (Figure 13). Even as the parameters change, the current recipes remain the most expensive while the optimized recipes are much cheaper.

Import and export costs are exceptionally high in sub-Saharan Africa, with the cost of trucking across a border averaging at \$1700 per 15 tons of material, or about half a truck of SNF. As seen in Figure 14b, prices drop dramatically as import and export costs decrease. Thus, improving cross-border transportation should become a priority for sustainable development.

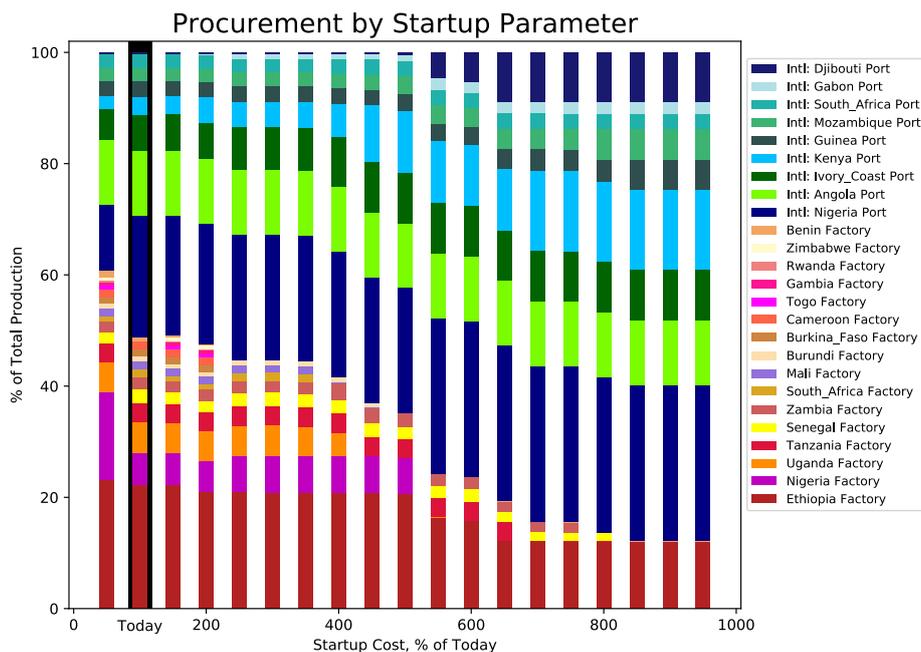


Figure 13. Optimal suppliers of acute malnutrition treatment, according to the changing startup parameter. Warm colors correspond to local production and cool colors correspond to international production.

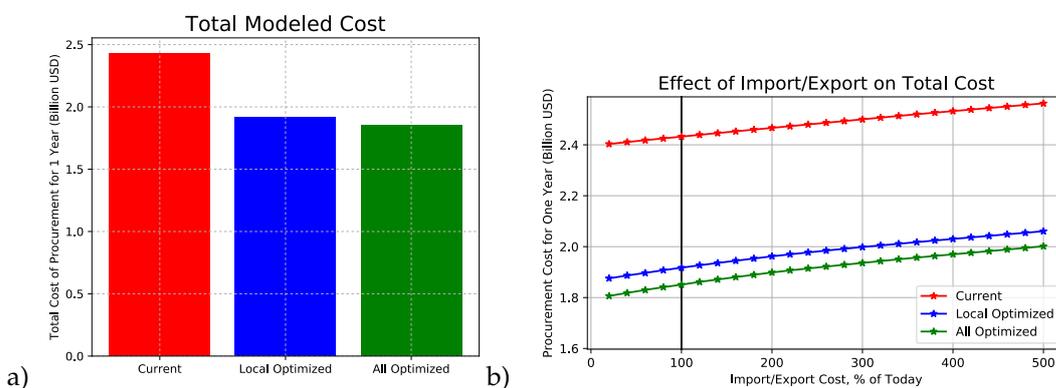


Figure 14. The cost of procurement for one year when all parameters are standard (a) and when varying the import/export parameter (b). The cost is shown for current SNF recipes (red), local optimized recipes (blue), and all optimized recipes (green).

4.3.3. Optimizing Cases Treated on a Budget: Building New Factories

The supply chain model optimized the number of children treated on a set budget. We optimized the treatment logistics of SAM and MAM separately, reflecting their current budgets. Based on UNICEF and WFP reports of current procurement of treatment, we estimated the budget of SAM and MAM treatment associated with costs included in our model to be 54 million each. As in Section 4.3.2, the supply chain was optimized using all optimized recipes, local optimized recipes, and the current recipes. First we discuss the scenario of all optimized recipes.

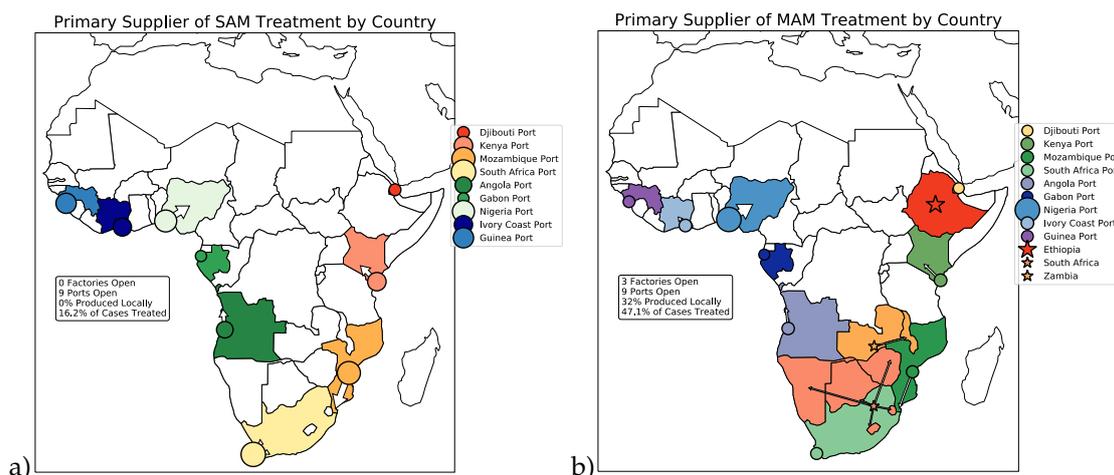


Figure 15. The primary supplier of SAM (a) and MAM (b) treatment under the current budget when all recipes are optimized.

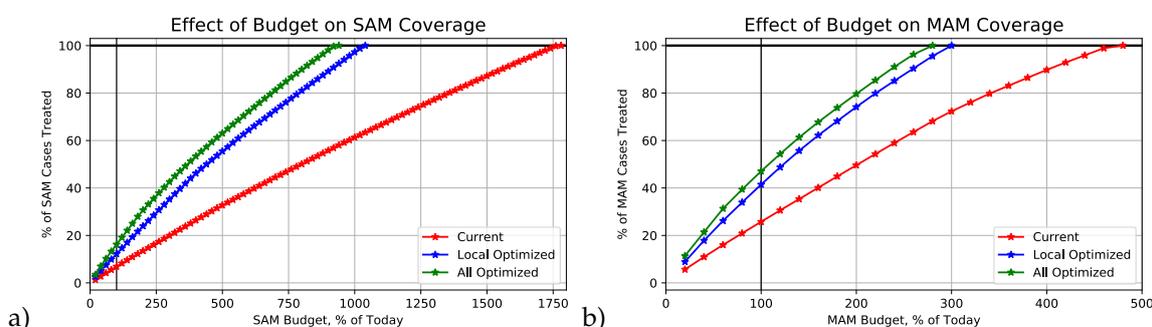


Figure 16. The percent of cases treated as the budgets for SAM (a) and MAM (b) treatment increases. To treat all cases using all optimized recipes, the MAM budget must be increased by 3 fold, while the SAM budget must be increased by 10 fold, due to the current small SAM budget.

Figure 15 shows the primary supplier of treatment for SAM (a) and MAM (b) treatment under the current budget. It is notable that most of the countries that are supplied with treatment are on the coasts, suggesting that international procurement is cheaper for coastal countries.

Under the current budgets, about 15% of SAM cases and 45% of MAM cases could be treated using all optimized recipes, and about half of that when using the current recipes. The numbers for current treatment compare well with the number of cases actually treated: UNICEF estimates that 3 million children suffering from SAM in sub-Saharan Africa receive treatment [2], and our model treats 2.9 million SAM cases when set to use current budget and treatment. To meet the full caseload with optimized recipes, the SAM budget would need to be increased by 10 fold, due to the low current budget, while the MAM budget would need to be increased by 3 fold (Figure 16). However, when considering only the current recipe, both budgets would need to be increased much more.

5. Conclusions

Due to difficulty in forecasting demand, the high cost of current treatments, and costly supply chains, acute malnutrition treatment reaches only a small fraction of children in need. Here we develop

a tool to inform nutrition interventions in sub-Saharan Africa. This tool can forecast the caseload of acute malnutrition, optimize SNF recipes, and inform on cost-effective production and distribution of SNF.

Our analysis suggests that current treatment of acute malnutrition is inefficient and unnecessarily expensive. The development of SNF leveraging amino acid supplements complementary to local crops could play a major role in reducing costs. Furthermore, the supply chain model identified countries with the optimal combination of low production costs and proximity to demand to support cost-effective local production. This proposed model can help assess relative location suitability for SNF production; compare local, regional, and international supply chains; identify barriers to low-cost treatment; and better inform policy makers or donor organizations on cost-effective nutrition intervention. Improved forecasting of acute malnutrition can enable timely procurement, shipping, and distribution of treatment, thereby lowering logistical costs.

We are currently applying the supply chain model to further scenarios. For example, to reflect actual procurement more closely, we are running the model according to newly provided data of UNICEF's actual RUTF procurement per country. We are also applying the model to compare between the current SNF, previously proposed soy-based SNF recipes based on literature, and an optimized SNF recipe.

Bringing low-cost SNF into use has the potential to greatly reduce the total cost of acute malnutrition treatment, and thereby reach more patients within the current budgetary constraints. All parts of the tool developed here can be adjusted by the user to include up-to-date information on ingredient costs, variable costs, and political situations, thus allowing aid organizations using this tool this tool to adjust distribution networks according to real-time information.

Used in conjunction, the forecasted demand, optimized recipes, and optimized supply chain model could allow more children to receive life-saving treatment within existing budgets while supporting sustainable agriculture and future food security in developing countries.

6. Acknowledgements

We would like to thank Valid Nutrition for conducting the prototyping of optimized SC+, Dr. Steve Collins for feedback and guidance on current practices, and Peter Akomo of Valid Nutrition for data on local production. We also thank Matt Cooper of University of Maryland College Park for help assembling training datasets for prediction of acute malnutrition.

7. References

1. Trehan, I.; Manary, M.J. Management of complicated severe acute malnutrition in children in resource-limited countries **2018**.
2. UNICEF. Malnutrition in Children, 2018.
3. UNICEF. Evaluation of Community Management of Acute Malnutrition (CMAM): Global Synthesis Report. Technical report, New York, 2013.

4. Collins, S.; Dent, N.; Binns, P.; Bahwere, P.; Sadler, K.; Hallam, A. Management of severe acute malnutrition in children. *The Lancet* **2006**, *368*, 1992–2000. doi:10.1016/S0140-6736(06)69443-9.
5. Collins, S.; Sadler, K.; Dent, N.; Khara, T.; Guerrero, S.; Myatt, M.; Saboya, M.; Walsh, A. Key Issues in the Success of Community-Based Management of Severe Malnutrition. *Food and Nutrition Bulletin* **2006**, *27*, S49–S82. doi:10.1177/156482650602735304.
6. Wagh, V.D.; Deore, B.R. Ready to Use Therapeutic Food (RUTF): An Overview. *Adv. Life Science and Health* **2015**, *2*, 15.
7. Manary, M.J. Local Production and Provision of Ready-To-Use Therapeutic Food (Rutf) Spread for the Treatment of Severe Childhood Malnutrition. *Food and Nutrition Bulletin* **2006**, *27*, S83–S89. doi:10.1177/156482650602735305.
8. Black, R.E.; Allen, L.H.; Bhutta, Z.A.; Caulfield, L.E.; de Onis, M.; Ezzati, M.; Mathers, C.; Rivera, J. Maternal and child undernutrition: global and regional exposures and health consequences. *The Lancet* **2008**, *371*, 243–260. doi:10.1016/S0140-6736(07)61690-0.
9. UNICEF. A supply chain analysis of ready-to-use therapeutic food for the horn of Africa. A study commissioned by the United Nations Children's Fund. Technical report, 2009.
10. Ahmed, T.; Choudhury, N.; Hossain, M.I.; Tangsuphoom, N.; Islam, M.M.; de Pee, S.; Steiger, G.; Fuli, R.; Sarker, S.A.M.; Parveen, M.; West, K.P.; Christian, P. Development and acceptability testing of ready-to-use supplementary food made from locally available food ingredients in Bangladesh. *BMC Pediatrics* **2014**, *14*, 164. doi:10.1186/1471-2431-14-164.
11. Park, J.H.; Kazaz, B.; Webster, S. Surface vs. Air Shipment of Humanitarian Goods under Demand Uncertainty. *Production and Operations Management* **2018**, *27*, 928–948. doi:10.1111/poms.12849.
12. UNICEF. Ready-to-Use Therapeutic Food: Current Outlook. UNICEF Supply Division, 2017.
13. Segrè, J.; Liu, G.; Komrska, J. Local versus offshore production of ready-to-use therapeutic foods and small quantity lipid-based nutrient supplements. *Maternal & Child Nutrition* **2017**, *13*, e12376. doi:10.1111/mcn.12376.
14. United Nations. First 1,000 Days, 2017.
15. Osgood-Zimmerman, A.; Millea, A.I.; Stubbs, R.W.; Shields, C.; Pickering, B.V.; Earl, L.; et al.. Mapping child growth failure in Africa between 2000 and 2015. *Nature* **2018**, *555*, 41–47. doi:10.1038/nature25760.
16. Graetz, N.; Friedman, J.; Osgood-Zimmerman, A.; Burstein, R.; Biehl, M.H.; Shields, C.; Mosser, J.F.; Casey, D.C.; Deshpande, A.; Earl, L.; Reiner, R.C.; Ray, S.E.; Fullman, N.; Levine, A.J.; Stubbs, R.W.; Mayala, B.K.; Longbottom, J.; Browne, A.J.; Bhatt, S.; Weiss, D.J.; Gething, P.W.; Mokdad, A.H.; Lim, S.S.; Murray, C.J.L.; Gakidou, E.; Hay, S.I. Mapping local variation in educational attainment across Africa. *Nature* **2018**, *555*, 48.
17. World Bank. World Bank Development Indicators, 2018.
18. Kumm, M.; Taka, M.; Guillaume, J.H.A. Data from: Gridded global datasets for Gross Domestic Product and Human Development Index over 1990-2015 **2018**. [itemType: dataset], doi:10.5061/dryad.dk1j0.
19. Tatem, A.J. WorldPop, open data for spatial demography. *Scientific Data* **2017**, *4*, 170004. doi:10.1038/sdata.2017.4.
20. Weiss, D.J.; Nelson, A.; Gibson, H.S.; Temperley, W.; Peedell, S.; Lieber, A.; Hancher, M.; Poyart, E.; Belchior, S.; Fullman, N.; Mappin, B.; Dalrymple, U.; Rozier, J.; Lucas, T.C.D.; Howes, R.E.; Tusting, L.S.; Kang, S.Y.; Cameron, E.; Bisanzio, D.; Battle, K.E.; Bhatt, S.; Gething, P.W. A global map of travel time to cities to assess inequalities in accessibility in 2015. *Nature* **2018**, *553*, 333–336. doi:10.1038/nature25181.
21. Uchida, H.; Nelson, A. Agglomeration Index: Towards a New Measure of Urban Concentration. *Urbanization and Development* **2010**.
22. Pesaresi, M.; Ehrlich, D.; Ferri, S.; Florczyk, A.; Freire, S.; Halkia, M.; Julea, A.; Kemper, T.; Soille, P.; Syrris, V. Global Human Settlement built-up grid, derived from Landsat, multitemporal (1975, 1990, 2000, 2014) - ecodp.common.ckan.site_title. *Joint Research Centre Technical Reports* **2016**. doi:10.2788/656115.
23. Kaufmann, D.; Kraay, A.; Mastruzzi, M. The Worldwide Governance Indicators: Methodology and Analytical Issues1. *Hague Journal on the Rule of Law* **2011**, *3*, 220–246. doi:10.1017/S1876404511200046.
24. Raleigh, C.; Linke, A.; Hegre, H.; Karlsen, J. Introducing ACLED: An Armed Conflict Location and Event Dataset: Special Data Feature. *Journal of Peace Research* **2010**, *47*, 651–660. doi:10.1177/0022343310378914.
25. Song, X.P.; Hansen, M.C.; Stehman, S.V.; Potapov, P.V.; Tyukavina, A.; Vermote, E.F.; Townshend, J.R. Global land change from 1982 to 2016. *Nature* **2018**, *560*, 639–643. doi:10.1038/s41586-018-0411-9.

26. Wessel, P.; Smith, W. A global, self-consistent, hierarchical, high-resolution shoreline databas, 2017.
27. Survey, U.G. GTOPO30, 1996.
28. Funk, C.; Peterson, P.; Landsfeld, M.; Pedreros, D.; Verdin, J.; Shukla, S.; Husak, G.; Rowland, J.; Harrison, L.; Hoell, A.; Michaelsen, J. The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. *Scientific Data* **2015**, *2*, 150066. doi:10.1038/sdata.2015.66.
29. FAO. FAOSTAT: Food and Agricultural Organization of the UN, 2019.
30. Herrero, M.; Thornton, P.K.; Power, B.; Bogard, J.R.; Remans, R.; Fritz, S.; Gerber, J.S.; Nelson, G.; See, L.; Waha, K.; Watson, R.A.; West, P.C.; Samberg, L.H.; van de Steeg, J.; Stephenson, E.; van Wijk, M.; Havlík, P. Farming and the geography of nutrient production for human use: a transdisciplinary analysis. *The Lancet Planetary Health* **2017**, *1*, e33–e42. doi:10.1016/S2542-5196(17)30007-4.
31. Siebert, S.; Henrich, V.; Frenken, K.; Burke, B. Update of the digital global map of irrigation areas to version 5., 2013.
32. CIESIN (Center for International Earth Science Information Network). Gridded Population of the World, Version 4 (GPWv4): National Identifier Grid, Revision 10 **2017**. doi:https://doi.org/10.7927/H4T72FDB.
33. Brixi, G. Innovative optimization of ready to use food for treatment of acute malnutrition. *Maternal & Child Nutrition* **2018**, *14*, e12599. doi:10.1111/mcn.12599.
34. Mekonnen, M.M.; Hoekstra, A.Y. The green, blue and grey water footprint of crops and derived crop products. *Hydrology and Earth System Sciences* **2011**, *15*, 1577–1600. doi:10.5194/hess-15-1577-2011.
35. Bahwere, P.; Akomo, P.; Mwale, M.; Murakami, H.; Banda, C.; Kathumba, S.; Banda, C.; Jere, S.; Sadler, K.; Collins, S. Soya, maize, and sorghum-based ready-to-use therapeutic food with amino acid is as efficacious as the standard milk and peanut paste-based formulation for the treatment of severe acute malnutrition in children: a noninferiority individually randomized controlled efficacy clinical trial in Malawi. *The American Journal of Clinical Nutrition* **2017**, *106*, 1100–1112. doi:10.3945/ajcn.117.156653.
36. Bahwere, P.; Balaluka, B.; Wells, J.C.K.; Mbiribindi, C.N.; Sadler, K.; Akomo, P.; Dramaix-Wilmet, M.; Collins, S. Cereals and pulse-based ready-to-use therapeutic food as an alternative to the standard milk-and peanut paste-based formulation for treating severe acute malnutrition: a noninferiority, individually randomized controlled efficacy clinical trial. *The American Journal of Clinical Nutrition* **2016**, *103*, 1145–1161. doi:10.3945/ajcn.115.119537.
37. Bulti, A.; Briend, A.; Dale, N.M.; De Wagt, A.; Chiwile, F.; Chitekwe, S.; Isokpunwu, C.; Myatt, M. Improving estimates of the burden of severe acute malnutrition and predictions of caseload for programs treating severe acute malnutrition: experiences from Nigeria. *Archives of Public Health* **2017**, *75*, 66. doi:10.1186/s13690-017-0234-4.
38. Isanaka, S.; Boundy, E.O.; Grais, R.F.; Myatt, M.; Briend, A. Improving Estimates of Numbers of Children With Severe Acute Malnutrition Using Cohort and Survey Data. *American Journal of Epidemiology* **2016**, *184*, 861–869. doi:10.1093/aje/kww129.
39. Isanaka, S.; Grais, R.F.; Briend, A.; Checchi, F. Estimates of the Duration of Untreated Acute Malnutrition in Children From Niger. *American Journal of Epidemiology* **2011**, *173*, 932–940. doi:10.1093/aje/kwq436.
40. World Food Programme. Specialized nutritious food.
41. World Food Programme. Nutrition at the World Food Programme - Programming for Nutrition-Specific Interventions. Technical report, 2012.
42. Akomo, P. Local Production of Specialized Nutritious Foods, 2018.
43. Engineers, W. RUTF Plant | Ready to Use Therapeutic Food(RUTF), 2018.
44. Teravaninthorn, S.; Raballand, G. *Transport Prices and Costs in Africa: A Review of the Main International Corridors*; World Bank Publications, 2009. Google-Books-ID: rS5r64EFStwC.
45. Eberhard-Ruiz, A.; Calabrese, L. Trade facilitation, transport costs and the price of trucking services in East Africa, 2018.
46. World Health Organization. Tariff Download Facility: WTO tariff data base, 2018.
47. UNICEF. Ready-to-Use Therapeutic Food Price Data, 2019.
48. Jelensperger, C. Technical Specifications for Ready-to-Use Supplementary Food. Technical report, 2016.
49. Nguyen, V.H. Technical Specifications for the Manufacture of: Super Cereal Plus, Corn Soya Blend. Technical report, 2014.

Tsunami Forecasting and Risk Assessment

New Mexico

Supercomputing Challenge

Final Report

April 3, 2019

Team Number 97

Los Alamos High School

Team Members:

Robert R. Strauss

Logan B. Dare

Teachers:

Mark Petersen

Project Mentor:

Charlie E. M. Strauss

Table of Contents

| | |
|--|-----------|
| 1 Executive Summary | 3 |
| 2 Introduction | 4 |
| Palu..... | 4 |
| What was different this time in Palu? | 4 |
| Krakatoa | 5 |
| Why Krakatoa?..... | 6 |
| Scaling the computational model | 6 |
| Roadmap of this report | 6 |
| 3 Methods | 7 |
| The physics..... | 7 |
| Finite element grid..... | 8 |
| Time step propagation | 8 |
| Numerical stability resolved..... | 8 |
| Data sets and management..... | 9 |
| From single threaded to massively parallel vector computation..... | 9 |
| Refinement & Verification | 9 |
| 4 Results | 11 |
| Palu | 11 |
| Krakatoa | 14 |
| Computer Model | 14 |
| 5 Discussion | 18 |
| Palu | 18 |
| Did the harbor shape or bathymetry have an effect? | 18 |
| Did the tsunami's initial location have an effect? | 18 |
| Did the properties of the wave have an effect? | 18 |
| Krakatoa | 19 |
| 6 Conclusion | 21 |
| 7 Acknowledgements | 21 |
| 8 References | 22 |

1 Executive Summary

Tsunamis are tragic events, killing thousands of people at a time, and to make matters worse, they often come unexpectedly because they affect places far from the point of origin. Two tsunamis have struck in Indonesia alone in the past six months, together killing over a thousand people. Can the damage from a tsunami be forecasted? Tsunamis can be simulated on supercomputers, but this is clearly not effective for real-time warning; if it were, the death count in each of these events would be much lower. I created a faster than real-time, highly accurate, and very affordable computer-based tsunami forecasting system, that has the potential to be part of a life-saving early warning system.

I built my model by creating a finite-element implementation of a set of differential equations. I numerically solved the non-linear coupled partial shallow water equations, describing the interaction between sea surface height and water velocity. I constructed a computer program including differential functions, time-step integrators, unit tests, and data and results visualization in thousands of lines of code, all from scratch. My model was my own creation; I did not implement a pre-existing ocean simulation engine. This meant I could write my code in a way that allowed for parallelization to run on a computer graphics card. This both cuts down the price dramatically from that of a supercomputer, while accelerating the simulation speed of my model to 28 times faster than real time. This makes it affordable to agencies of developing countries or even small villages.

I validated my forecasting system on the outcomes of two real tsunamis that happened last year: a tsunami that devastated Palu City, Indonesia, and one triggered by the eruption of volcano Anak Krakatoa. The destruction to Palu was surprising because Palu seemed to be protected from previous tsunamis by its harbor. To determine the factors contributing to the severity of this case, I asked three questions: First, “did the harbor shape or ocean floor topology have an effect”, second, “did the initial location of the wave have an effect”, and third, “did the properties of the wave have an effect”. I simulated several tsunami events with my model to answer these. I answered the questions I asked, and in doing so I isolated the key factors in the severity of the Palu case. For the Krakatoa tsunami, I tested the accuracy of my model at predicting the distribution of damage from a tsunami. I simulated the Krakatoa tsunami, and compared the forecasted damage from my model to the real damage. I found that my model accurately predicted not only hazard areas but safe zones. My tool has a user interface, simulates significantly faster than real time, and is connected to a database allowing it to be applied to any area on Earth.

2 Introduction

Seismic events such as landslides and earthquakes near water create tsunamis, whose impact can be devastating and deadly. Tsunamis can deliver immense damage at places far from their origin and often arrive without warning. Risk analysis of coastal areas can be aided by computational modeling of tsunamis. In a computer model, the initial conditions and factors can be altered, whereas there is only one data point from real life. Thus a computer model is especially useful for risk assessment. This project produced a computer model with an easily adjusted user interface that also presents results with graphs, heat maps, and animations.

Simulation also allows multiple types of assessment. One type is to start hypothetical tsunamis at likely points of origin, such as an earthquake fault, then predict the resulting distribution of damage to surrounding coasts. Another type is to investigate why, that is what special features, put certain coastal regions at higher risk than others. Here, both types of assessment are demonstrated in the context of the 2018 Palu earthquake-tsunami and the 2019 Anak Krakatoa volcano landslide-tsunami.

Palu

In 2018 a tsunami devastated Palu city, Indonesia.

- More than 800 people were killed.
- Over 500 were severely injured.
- 48,000 were made homeless

The Palu region is shown in figure 1. Before and after images of a bridge in Palu to give a sense of the extreme power of the impact. (Figure 2 and figure 3)

What was different this time in Palu?

Palu is deep inside a protective harbor. Historically, it had been a rather safe area from tsunamis, so it was especially surprising this tsunami devastated Palu to the extent that it did. Media speculations included the shape of the harbor guiding the wave or the special location of the tsunami or the direction or type of the earthquake fault making for enhanced severity. (4,5)

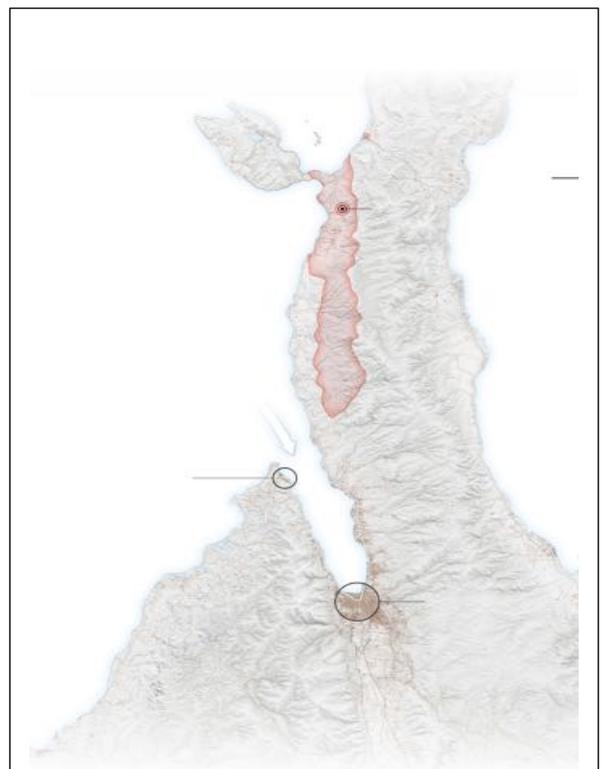


Figure 1: (NY Times) Map around Palu, Indonesia, showing earthquake epicenter (red dot), area of most intense earthquake shaking (red area), and damage to Palu (shaded in brown). Up is North.



Figure 2: (NY Times) Bridge in Palu before tsunami. In good condition, undamaged.



Figure 3: (NY Times) The same bridge after the tsunami completely destroyed

All of these factors are plausible, but what contributes most to the severity? I asked three questions: “did the harbor shape or ocean floorTo answer this, the effect of the tsunami at Palu is observed as I change its initial parameters including its point of origin, type of wave, and angle. Furthermore, I looked at a time series of the tsunami’s evolution in the harbor channel to see how the bathymetry profile (ocean topology/depth) contributed to the severity.

I found that the location of the tsunami was critical to admit wave energy into the harbor. But once in the harbor, the location of the damage was determined by the bathymetry more than the point of origin. Additionally, the shape and direction of the wave had only minor effect. While the guesses in the media were obvious ones to make, not all of them were correct and the results depended on the unfortunate combination of factors. *For example, if Palu had been located at a different point along the channel it might have evaded much destruction.*

Interestingly, I also discovered that a second completely unexpected mode for strong energy coupling into Palu harbor was possible, greatly enlarging the origin hazard zone. This was from a Kelvin wave being guided along the shore up North to the mouth of Palu bay (“Kelvin Waves”). A Kelvin wave is a phenomenon that occurs when the Coriolis force – a fictitious force from the spinning of the Earth – pushes a wave against a bathymetric feature, and between the two the wave is guided in a direction. I observe in both the wave patterns and maximum height predictions that it can, surprisingly, round the corner and enter into the harbor, despite the point of origin appearing to be blocked by the harbor orientation.

Krakatoa

The second case (figures 4, 5) that was investigated was the collapse of volcano Anak Krakatoa in Indonesia. Anak Krakatoa is in the ocean, so when it collapsed it caused a tsunami. The nearby islands of Java and Sumatra suffered major devastation from this tsunami. In the Krakatoa tsunami: (figures 4,5)

- Over 500 people were killed.
- Over 1400 people were injured.



Figure 4: (The Telegraph) Scene of destruction caused by the Krakatoa tsunami

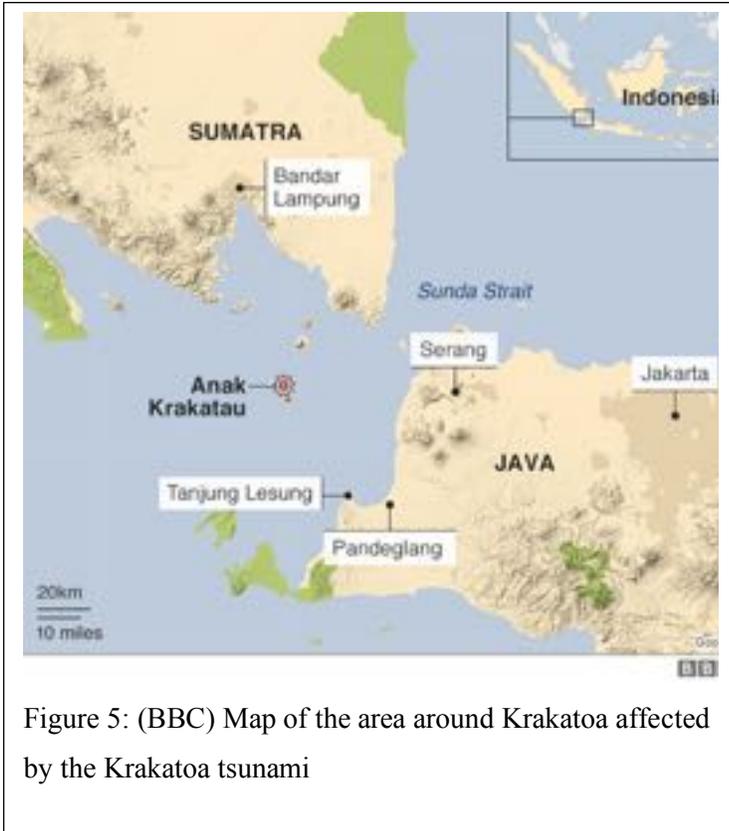


Figure 5: (BBC) Map of the area around Krakatoa affected by the Krakatoa tsunami

Why Krakatoa?

When this project was started, a list of likely tsunami origins that could be simulated to forecast risk regions was compiled. High among these was the infamous Krakatoa volcano tsunami from 1883, the largest tsunami in recorded history, killing tens of thousands of people in diverse and distant areas. Coincidentally, during this project a cinder cone on the caldera of Krakatoa erupted, creating a landslide, which in turn created a tsunami. This is the event that will be forecast in this report. This widened the scope of the project from one of demonstrating a forecast to analyzing the agreement of my model to the actual event.

From the perspective of additional validation, the agreement of the model to the real outcome shows the power of this for risk assessment. No parameters of the simulation were artificially altered to increase agreement, meaning this model was run in a forecasting mode, not a model-fitting mode.

Scaling the computational model

To model larger forecast-scale regions and with higher accuracy, the simulation system was scaled up to use the memory more efficiently as well as speed up the calculations, cutting simulation time down from days to hours. Supercomputing advisors suggested using a large cluster. Instead I initially scaled to eight parallel CPU cores each with four-wide SIMD vector processing. Then I scaled to tens of thousands of threads on thousands of processors with 32-wide SIMD vector processing via a GPU.

Roadmap of this report

This paper is organized as follows: how the simulation was developed is discussed within the methods section. The physics is modeled by Non-linear shallow water differential equations. I discuss how they were implemented computationally, finite element and finite time step discretization, how instabilities were avoided, where bathymetry data was acquired, and more. The implementation is verified. Applications to understanding Palu and forecasting Krakatoa are shown in the results section. The discussion analyzes the data, showing that the forecasts are validated.

Code for this project can be found here: <http://github.com/robertstrauss/shallowwater>

3 Methods

The physics

The approximated physics of the model is entirely contained in a set of coupled partial differential equations, known as the “non-linear shallow water equations” (“Thacker, William Carlisle”), shown in figure 6. There are no adjustable parameters – any coefficients are set constants such as gravity – meaning there is no model fitting aspect in this project. Instead, the experimental parameters of the simulation are event location, coastal boundary condition, and wave shape.

The shallow water equations are an approximation of the behavior of water. In a shallow water equations system, a body of water is simplified to many individual columns of water of varying height and velocity. The equations incorporate terms for advection, a sort of momentum for propagating waves, attenuation, a friction term, and the Coriolis force, a fictitious force arising from the rotation of the Earth.

$$\frac{\partial \eta}{\partial t} = -\frac{\partial}{\partial x}((\eta + h)u) - \frac{\partial}{\partial y}((\eta + h)v)$$

$$\frac{\partial u}{\partial t} = \text{Coriolis} + \text{Advection} + \text{Gravity} + \text{Attenuation}$$

$$= +fv + (\kappa \nabla^2 u - (u, v) \cdot \vec{\nabla} u) - g \frac{\partial \eta}{\partial x} - \frac{1}{\rho(h + \eta)} \mu u \sqrt{u^2 + v^2}$$

$$= +fv + \left(\kappa \frac{\partial^2 u}{\partial x^2} + \kappa \frac{\partial^2 u}{\partial y^2} - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} \right) - g \frac{\partial \eta}{\partial x} - \frac{1}{\rho(h + \eta)} \mu u \sqrt{u^2 + v^2}$$

$$\frac{\partial v}{\partial t} = -fu + (\kappa \nabla^2 v - (u, v) \cdot \vec{\nabla} v) - g \frac{\partial \eta}{\partial y} - \frac{1}{\rho(h + \eta)} \mu v \sqrt{u^2 + v^2}$$

$$= -fu + \left(\kappa \frac{\partial^2 v}{\partial x^2} + \kappa \frac{\partial^2 v}{\partial y^2} - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y} \right) - g \frac{\partial \eta}{\partial y} - \frac{1}{\rho(h + \eta)} \mu v \sqrt{u^2 + v^2}$$

h - bathymetry (depth)

f - Coriolis force $\propto \sin(\text{latitude})$

η - surface height deviation

κ - viscous damping coefficient

u - X speed (East)

μ - friction coefficient

v - Y speed (North)

g - gravity

Figure 6: The shallow water equations. A set of three differential equations, representing the rate of change of the sea surface height and velocities.

To implement these equations computationally, discretization was performed and a time-step integrator for these equations was constructed from scratch.

Finite element grid

A computer model was created to simulate the ocean. To do this, a system for storing the state of a body of water as well as a finite element model of the shallow water equations (via discretization) had to be engineered. This meant creating a discrete system, in which time and space come in discrete chunks rather than continuously. The size of the time step was determined from the size of spatial elements and the depth. The shallow water equations can be approximated to a simple linear wave equations producing a depth-dependent wave speed, equal to the square root of the product of the depth and the gravitational acceleration ($c = \sqrt{gh}$). The time step ' Δt ' must be shorter than the time it takes for a wave to move across the width of an element, ' Δx '. Thus, Δt and Δx are related by $\Delta t < \Delta x/c$. This is known as the CFL condition (Caminha, Guilherme)

Time step propagation

The shallow water equations give the rate of change of the water height and velocity at each time step. The most obvious approach would be to simply add the rate of change times the timestep on to the values. However, this can be unstable. Several other methods of higher order integration were investigated.

The time-step integrator had multiple time-stepping methods: a simple forward Euler method, a forward-backward method, a forward-backward method with a corrector step (forward-backward feedback), and a generalized forward-backward method. There was a time-memory tradeoff between these time-stepping methods; The generalized forward-backward time-step, for example, was faster than the forward-backward feedback time-step, but took up more memory at a time.

Numerical stability resolved

Finite element simulation of coupled partial differential equations can have stability issues. There are three ways to deal with these. First, a higher order approximation integrator can be used as noted above. Second, the temporal and spatial step sizes can be reduced using the CFL condition. And third, damping can be introduced.

The damping term appears in the equations due to the friction with the ocean floor. However, because the shallow water equations integrate out the entire vertical water column the friction coefficient is affecting the entire finite element water speed. Since setting the coefficient on the damping term seems to be in a way subjective (private communication with Bill Lucy Knight, University of Alaska), it was aimed to set this to be as small as possible. To determine this value, the damping coefficient was slowly increased until the differential equations became stable. In the literature of the Non-linear shallow water equations, the viscosity term appears to be usually ignored because it's a higher order approximation, so

chi (κ) was set to zero. (private communication with Bill Lucy Knight, University of Alaska) This is useful for a second reason: the viscosity term can be numerically unstable since it is a higher order derivative whereas the damping term is stabilizing.

Data sets and management

Bathymetry (ocean depth) data was acquired from The CoNED project (“Coastal National Elevation Database (CoNED) Project - Topobathymetric Digital Elevation Model (TBDEM) | The Long Term Archive”), and packages such as pandas and netCDF were used to query these large data within the program. This allows the program to consider any region of Earth just by inputting the coordinates.

From single threaded to massively parallel vector computation

Computational methods were investigated for optimizing the speed of the code’s execution. The code was implemented with pure python, the slowest method; the numpy python library, allowing for array calculations; the numba library, which compiled the python code; and the cuda library, allowing for the code to be run on the graphics card in a massively parallel way. On a small test case of array size 200 by 200 the time the code took to complete a simulation was recorded: pure python was the slowest method, with a time of 39.9 seconds; numpy was much faster, with a time of 1.0 second; numba was even faster, with a time of 0.4 seconds; and cuda was by far the fastest, with a time of 0.05 seconds, about 800 times faster than pure python.

Scaling up to 200 times to 4000x2000 point arrays and a long 550 second simulation, I observed the cuda ran in 19 seconds and the Numba in 1260 seconds, showing the graphics card gains more as the array size increases (from 8 times faster to 66 times faster).

Refinement & Verification

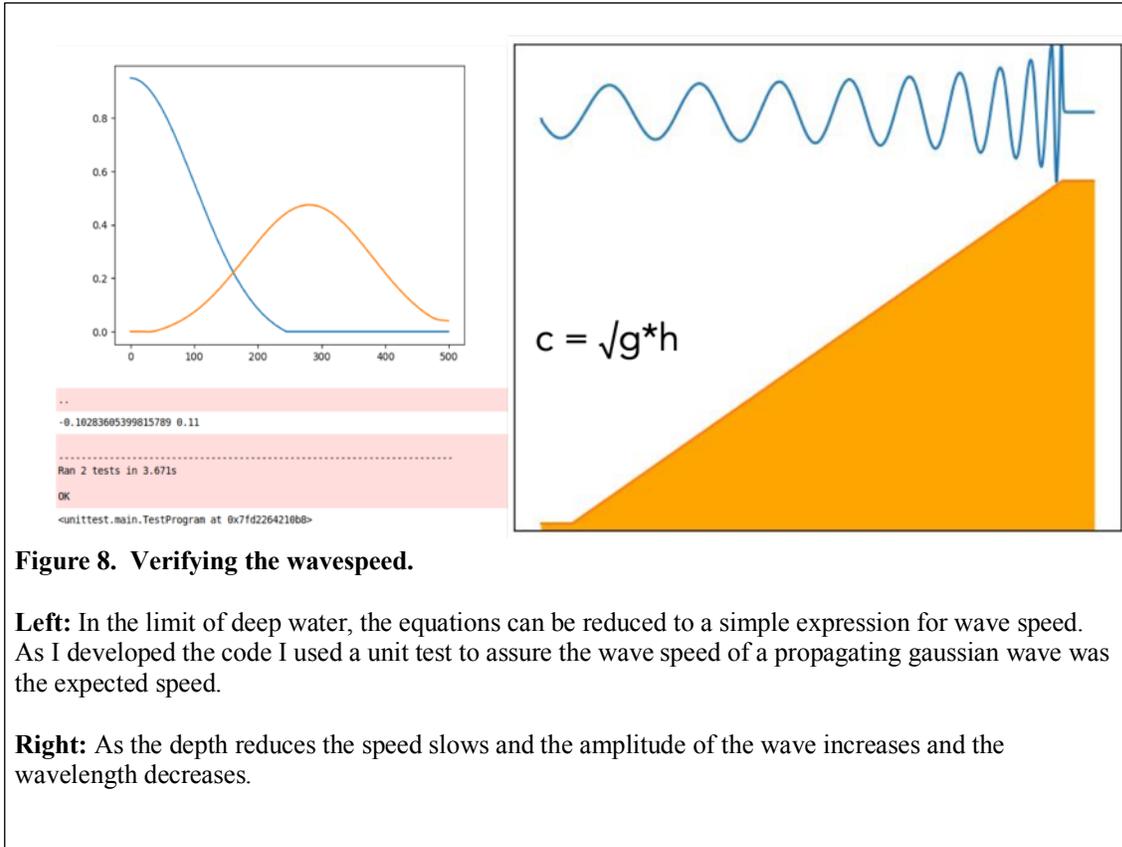
Most of the time in this project was spent developing and refining the model. I began with the forward Euler time-step, but doing larger and longer simulations caused instabilities with that technique, so I switched to a forward-backward predictor-corrector time-step. I also created restrictions in my model to make sure the CFL condition was met. To make sure my model was working correctly, I verified it with unit tests.

Verification is checking that a model is working as expected and not breaking down, while validation is checking if the model is accomplishing the overall goal. The shallow water equations are a well-known and valid model. However, because I created my own new implementation of the shallow water equations, I wrote the program myself, verification is required. I verified my model with unit tests and an animation of a wave propagating.

In the shallow water model, waves propagate at approximately the square root of the product of gravity and depth. I let a wave propagate out for a fixed amount of time in simplified/controlled

conditions. I then compared the distance it went divided by the time it took with the expected approximate square root gravity times depth value. See figure 8.

I also verified my model by showing a simple initial disturbance propagate outwards in all directions within simplified/controlled conditions. See figure 7. This shows the model is working as expected and produces what one would expect to see in these conditions.



4 Results

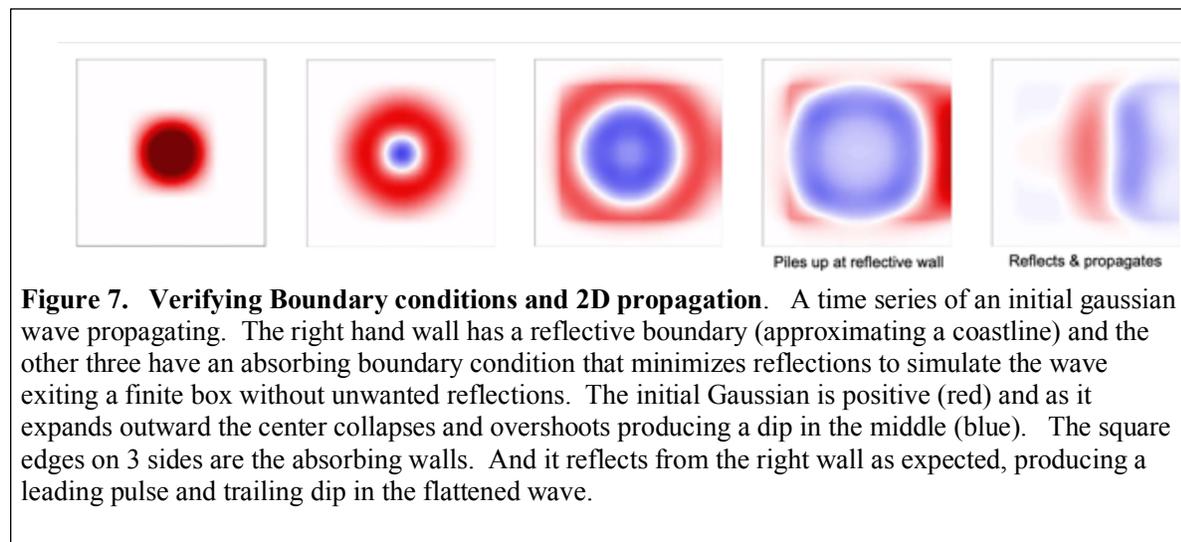
Palu

A Time series of the wave propagating is shown in figure 11. The wave starts as a simple initial disturbance and then propagates out into a ring. The wave flattens against the coast. A part of the wave enters Palu bay, and propagates down to the end. The wave on the heat map reddens upon reaching the end of the bay, indicating it is increasing in height as it strikes Palu city.

I ran many simulations varying the origin location of the tsunami, the height of the tsunami, and three different tsunami wave shapes in various orientations. I compared the maximum wave heights in Palu bay resulting from each one. A subset of these many runs are shown in figure 12abc. I have only shown 3 types of Tsunami wave shapes here because overall the results are similar in their forecast of where the wave will strike, meaning that the initial tsunami wave shape is not a major factor in the outcome at a distant location.

In each plot the initial condition of each event is displayed, beside the resulting maximum water height in Palu. The maximum water height contains the largest height a point reached during the whole simulation, for each point. This is displayed as an image. The impact of the tsunami at a given location is expected to correlate to the maximum water heights near that location, and so maximum height is used as a measure of impact. (Kristina, W)

The location of the origin, and bathymetry visibly affect the Maximum Wave Heights in figure 12abc. And the bathymetry dramatically affects the wave height as seen in the time series Figure 11.



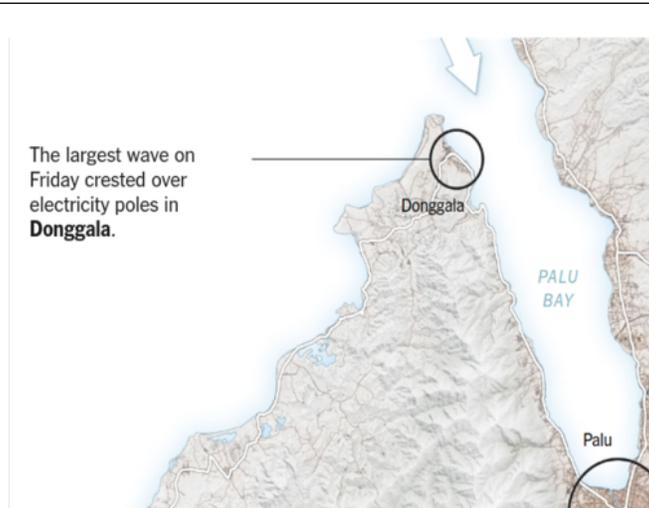


Figure 9: (NY Times) image of Palu Bay. The largest waves were recorded on the left side of the entrance to the bay. Palu City was the next highest wave height observed.

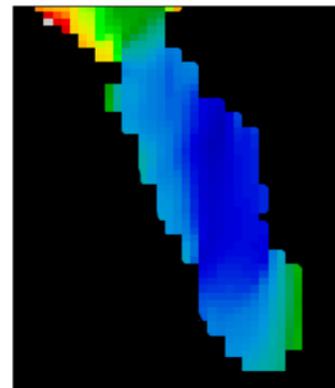


Figure 10: Palu bay maximum wave heights from simulation of actual event. Heatmap image shows highest waves were at the entrance to Palu bay. The particular simulation shows the results for a elliptical gaussian. Tsunami shape. Results for other Tsunami shape models are similar and show in figures below.

Animation of a Tsunami simulation starting North of Palu bay

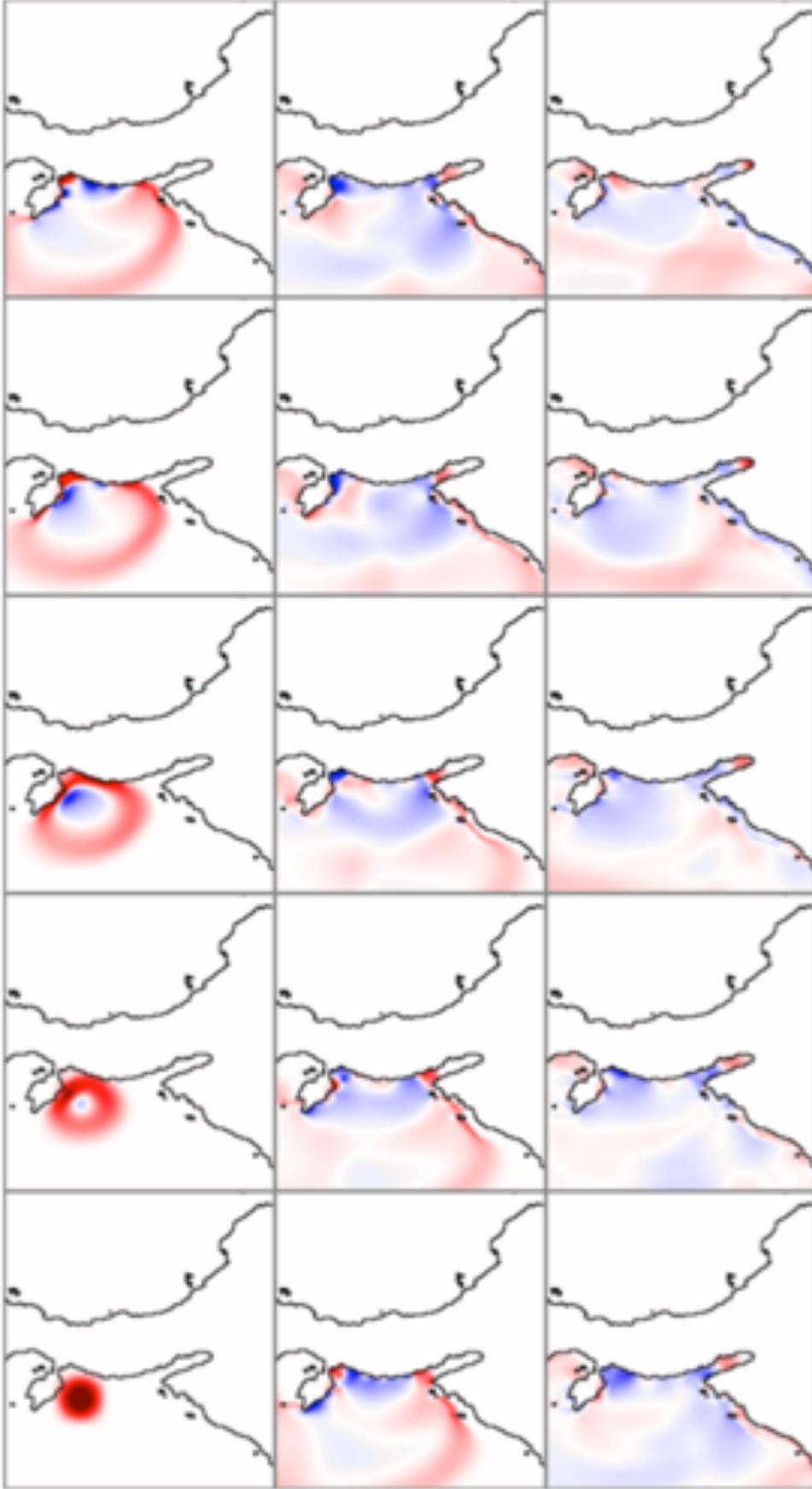


Figure 11. Time series of the Tsunami wave height at 100 second intervals. RED: positive wave height, BLUE negative, BLACK coast. The wave noticeably bends as it nears the shallow water near the coast. Palu bay is the notch in the lower middle of the image. Notice the Red brightens as it enters the harbor. This corresponds to the observed high wave in reality. The wave speed increases in the deep harbor water. Then the Red brightens again as it reaches the shallow slope at the end of the Palu bay where the city is located

Krakatoa

The maximum heights from the simulation of the Krakatoa case are seen in figure 14. A map showing where damage was inflicted on the surrounding coast in real life is seen in figure 13. These are compared as a measure of the accuracy of the simulation to real life. If the simulation produces larger maximum heights at areas that suffered a lot of damage in real life and produces smaller maximum heights at areas that suffered little damage in real life, then the model is accurate to real life. As it is compared, the model is found to be very similar to the real life damage map, and so very accurate.

Computer Model

The computer model was continuously produced during this project. It was over a thousand lines of code in total. Much more than just a shallow water computer model was created. In this project, big data was handled, an interactive graphical user interface was created, and code was written in multiple different ways, allowing for vector operations and running the code on a massive scale with thousands of parallel threads. Bathymetry data was acquired for all over the globe, meaning the model can easily be applied to anywhere on Earth. Additionally, the model simulates many times faster than real time, meaning given an initial tsunami, the model can accurately predict what it will do before the real tsunami does it.

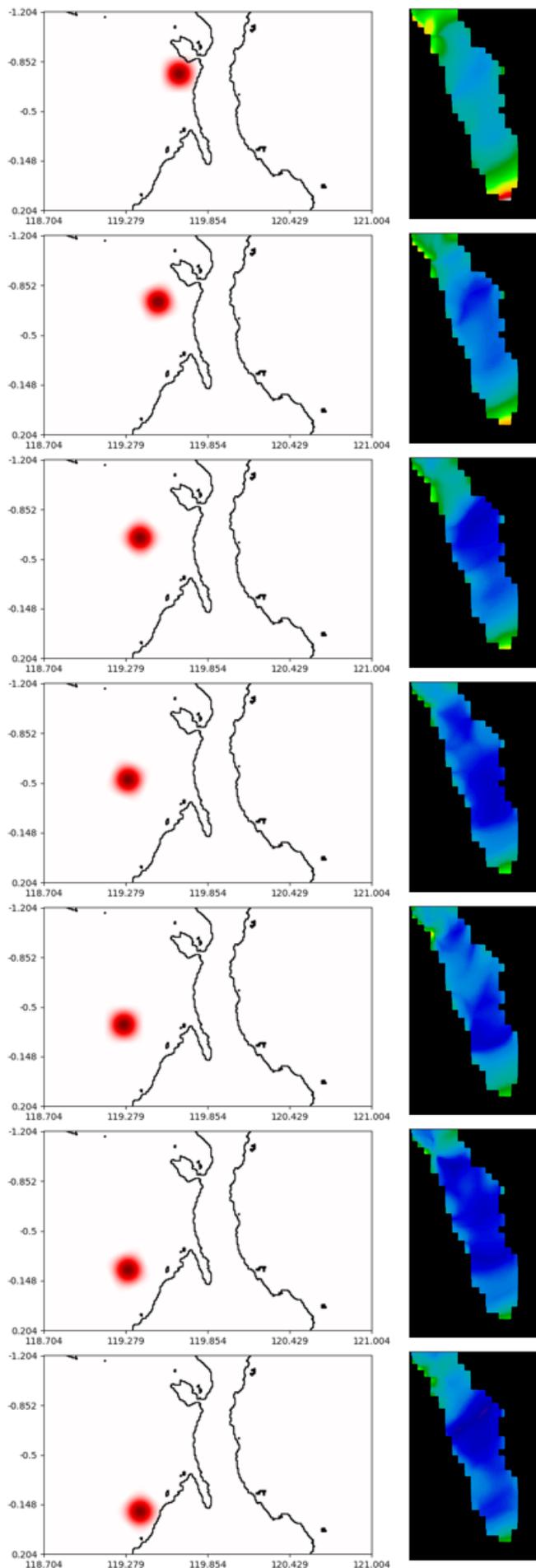


Figure 12a: a 5 meter tall round gaussian wave.

Simulations were run using different wave shapes, orientations, and heights. A sample of three of these are shown in Figure 7.

Figure 7a shows a 5 meter tall round gaussian wave.

LEFT: the initial condition,

RIGHT: the resulting maximum water height in Palu bay.

Observations:

1. **Alignment:** Only when the northern most events must align with the bay axis in a very narrow angle are high waves produced in Palu.
2. **Bay shape:** The largest waves occur at the base of the cove and at the very entrance.

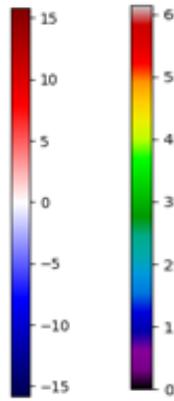
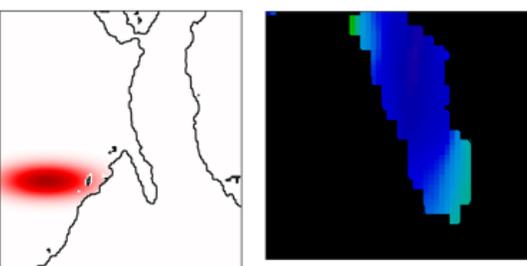
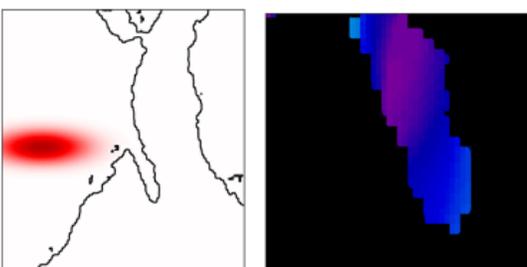
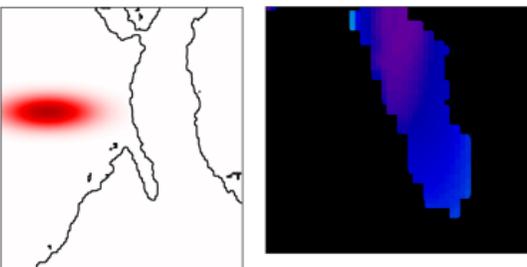
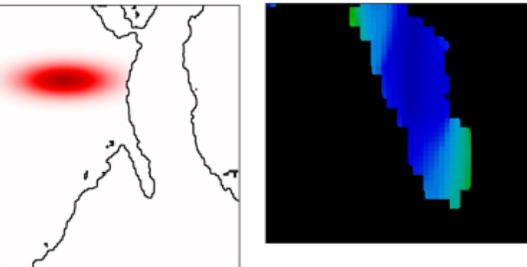
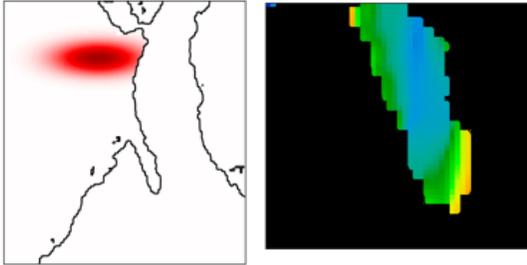
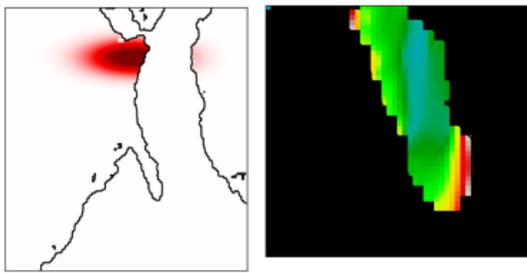


Figure 12b: a 15 meter tall, wide, elliptical gaussian in a particular orientation relative to Earth

Simulations were run using different wave shapes, orientations, and heights. Only three of these are shown in this report as they were generally similar.

This series uses a more intense elongated ellipse. The greater intensity appears to be responsible for the added observation of the kelvin wave.

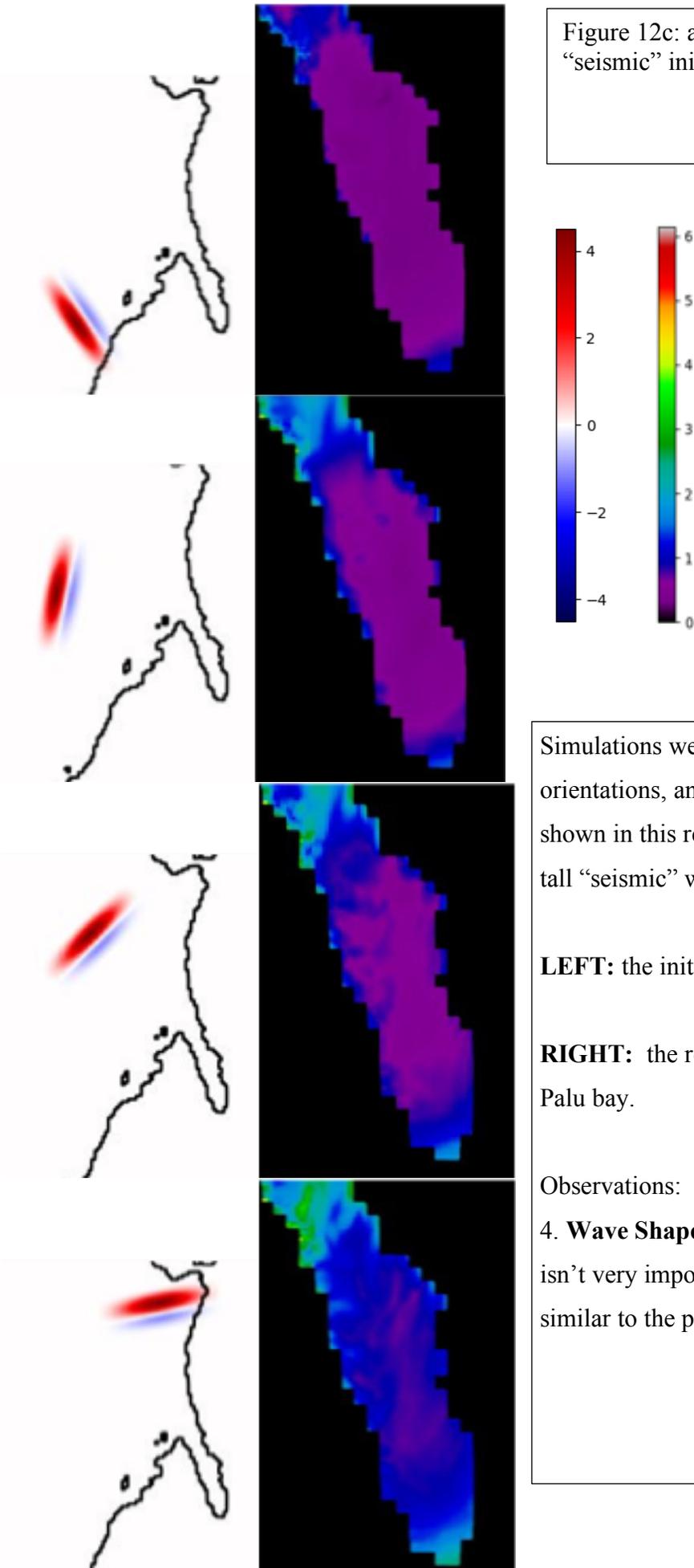
LEFT: the initial condition,

RIGHT: the resulting maximum water height in Palu bay.

Additional Observations:

3. **Kelvin Wave:** While overall, less aligned events couple less energy into the bay, surprisingly, the southern most event (bottom) shows an increase in the amount of wave height coupling into the bay compared to more northern waves

Figure 12c: a 15 meter tall, wide, “seismic” initial condition



Simulations were run using different wave shapes, orientations, and heights Only three of these are shown in this report Figure 12c shows a 5 meter tall “seismic” wave.

LEFT: the initial condition,

RIGHT: the resulting maximum water height in Palu bay.

Observations:

4. **Wave Shape:** Evidently the initial wave shape isn't very important to the outcome as this is similar to the previous two.

5 Discussion

Palu

Figure 12 displays data from multiple type of initial condition. The first is a circular gaussian, the most simple of initial conditions. The next is an elliptical gaussian, which may make more sense because an earthquake fault will be long in one dimension. Then, there is the “seismic” condition, which is much more realistic for an earthquake or slip fault source event. This initial condition is composed of two elliptical gaussians, one positive and one negative. The following is true for all of these.

Did the harbor shape or bathymetry have an effect?

In figure 11, deeper red represents higher waves, and deeper blue represents more negative waves. As the wave propagates out from its starting point, it is most red at the South side, which is the piece of the wave that will enter Palu Bay. The wave is very red as it enters Palu bay, but then much fainter as it goes down Palu bay. It then gets much redder at the end of Palu Bay, right at Palu city. This is because the end of the bay is shallower, so the wave is pushed up higher. Yes, the harbor shape and bathymetry had an effect.

Did the tsunami’s initial location have an effect?

Multiple simulations are represented in each section of figure 12 (12a, 12b, 12c). On the left, the initial condition is shown, and on the right, the resulting maximum height in Palu Bay is shown. More warm colors such as red and yellow represent higher waves, while cooler colors such as blue and purple represent lower waves. The image with the most “warm” colors in it is the maximum height plot corresponding to the initial condition in which the tsunami starts most North. Initial conditions South of the angle of Palu Bay have much “cooler” maximum height graphs. The intensity appears to drop off significantly as the initial tsunamis passes the angle of Palu bay. Tsunamis starting South of a line extended from Palu through the mouth of Palu bay do much less damage. Yes, the tsunami’s initial location had an effect.

Did the properties of the wave have an effect?

All the initial conditions or wave types tested displayed similar results, with one small exception. Only true of the elliptical initial condition, the intensity increases again in the final data point. This is likely because of a Kelvin wave (“Kelvin Waves.”), a wave that is guided along the coast. This tsunami starts close to the coast, so a Kelvin wave is likely, and may guide the wave up to the mouth of Palu bay. Because the Coriolis force is asymmetric this not seen in waves reflected off the north wall in the animation. For the most part, no, the properties of the wave did not have an effect.

Even larger maximum wave heights were observed at the mouth of Palu bay. This agrees with the real case, in which the highest waves were seen exactly. The model forecasted relatively lower wave heights along the inlet only becoming high again as it approached Palu, agreeing with reports. This shows that the forecast not only predicts where the waves will be high waves but it predicts where they are low

too. See figures 9 and 10. Because these observations agree, this point serves as further validation for my model.

For forecasting purposes some factors suggested by experts did not matter. The height of the tsunami, the shape of the tsunami, and the orientation of the tsunami did not strongly affect where the highest waves occurred. The kelvin wave effect is seen more clearly in the larger tsunamis but is visible in both the high and low wave height examples shown in the figures. Thus, Palu's unfortunate devastation was a matter of the bathymetry profile and the very unlucky location of the event being in exactly the worst possible place.

Krakatoa

Between figures 13 and 14, showing the maximum water heights produced by my simulation, and figure 13, a map displaying what areas were affected most in the real event, many similarities can be seen. Darker shades of red on the maximum water height images are located similarly to patches of yellow in the affected areas map and lighter shades of red located near the blank patches on the map, meaning the simulation produced similar results to that seen in real life. No factors were controlled to make the model fit real life better, meaning this model is run in a forecasting mode, rather than a model fitting mode. This means the model could be used as a danger forecaster to predict what regions of an area will be most severely damaged by a tsunami, with a high degree of accuracy.

Again, the forecast of both high and low threat areas is shown to agree with the news reports of damage. Notice in figures 13 and 14 that the Pesawaran and Sunda strait near Serang and Pagelaran coasts escaped inundation, as the predictions also indicate. This is despite appearing to be in line of sight with the tsunami origin, thus safe zones are not necessarily obvious ahead of time. Likewise, damage is seen wrapping around to behind the coast, so obvious safe zones may not be safe. It is important that both safe zones and hazard areas were predicted, because people need to know where to flee to, and where not to flee from.

Volcano-triggered tsunami hits Indonesia

The powerful tsunami struck at night and without warning, sweeping over popular beaches and inundating tourist hotels and coastal settlements

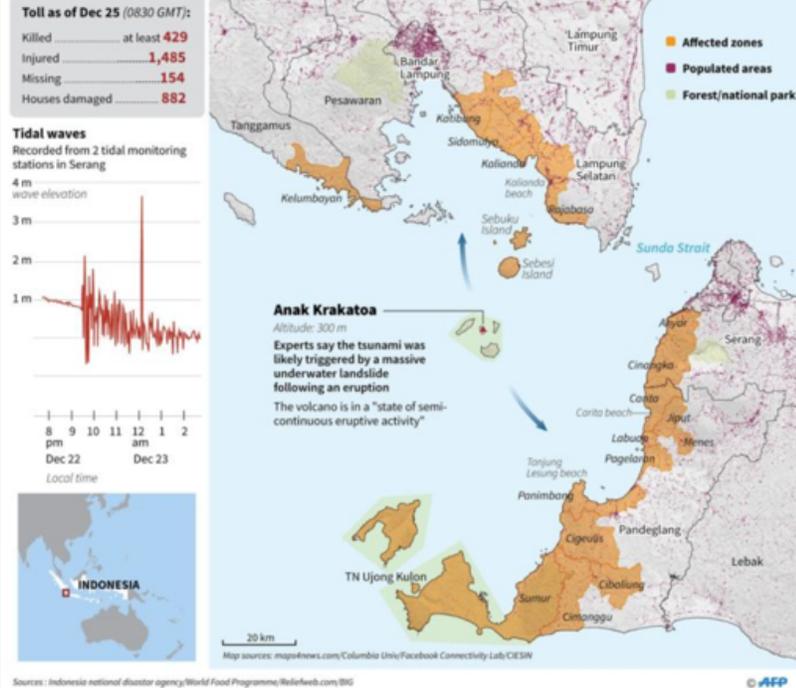


Figure 13: (Singapore news) Map of effected areas from Krakatoa tsunami. Orange colored regions represent flooded areas, and blank areas are less effected. Notice that the Pesawaran and Sunda strait near Serang and Pagelaran coasts escaped inundation, as the predictions also indicate.

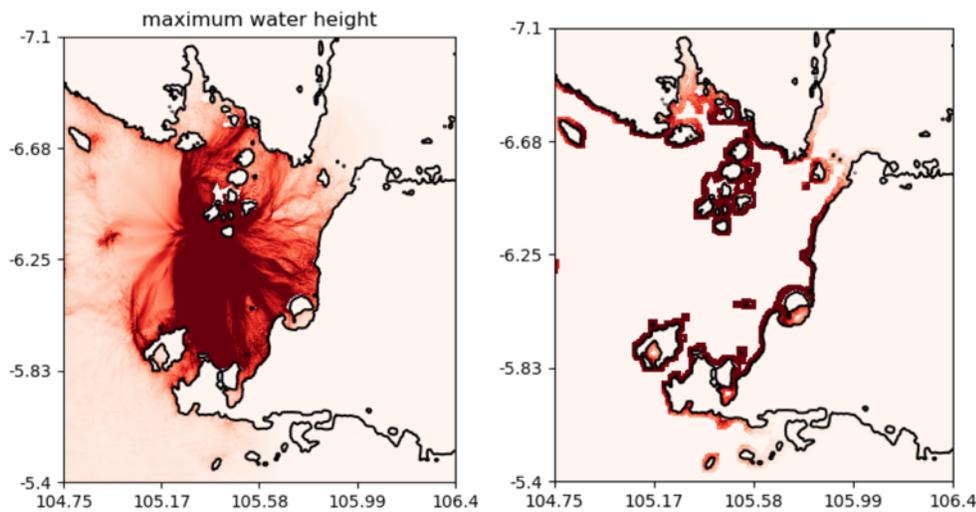


Figure 14: The Krakatoa Forecast: maximum water heights produced by the computer simulation. Some areas are much more effected, visualized by being a deeper red. The distribution of high waves along the coast varies considerably. For better visibility near the coast, the color bar is clipped at the high end **Left plot:** shows only the intensity near shore. Axes are Latitude and Longitude.

6 Conclusion

It was found that the shape of Palu bay did indeed worsen the Palu tsunami. I successfully created my own finite element implementation of the shallow water equations. My findings for the Palu case show: First, as one might guess, narrow inlets are protective in that they limit the directions from which a tsunami can easily enter. Second, the inlet's tapering ocean floor will also guide and amplify waves that do enter as they reach the end of the bay. And third, most surprisingly, special coastal features outside of the inlet can unexpectedly expand the hazard zone, via a Kelvin wave traveling along the southern side of the bay.

A pleasantly surprisingly high degree of accuracy was found for risk assessment of the Krakatoa case. Not only does it predict all the observed high water inundations but it also predicts where they did not happen. This shows the model would be excellent for risk forecasting, something that could save lives by predicting tsunami danger zones. This model could be used in the future to create a danger level map of a whole region, which could determine what regions are at especially high risk, and so what regions should take precaution.

The constructed computer model was able to simulate much faster than real time, and can be applied for any location on Earth. In the future it could form the basis for an integrated seismic event warning system if it were coupled with real time seismometer event locations. Because it runs faster than real-time on an inexpensive GPU it could be affordable to businesses and villages in seismically active regions.

The code is documented and available on Git Hub at this link:
<http://github.com/robertstrauss/shallowwater>.

7 Acknowledgements

LD created the on-line web site report. RRS wrote the code, designed and performed the experiments, and wrote the report and poster. Mentor MP suggested the shallow water equations, provided literature and tutorials on integration of the equations, and made suggestions for organizing the poster. Mentor CEMS taught python suggested key libraries like pandas, numba and numpy, how to code and reviewed the code for errors and bugs. Bill Lucy Knight, University of Alaska provided some hints on the shallow water equations by e-mail.

8 References

1. Caminha, Guilherme. “The CFL Condition and How to Choose Ymy Timestep Size.” *SimScale*, SimScale, 21 Mar. 2019, www.simscale.com/blog/2017/08/cfl-condition/
2. Center, National Geophysical Data. Tsunami Events Full Search, Sort by Date, Country. https://www.ngdc.noaa.gov/nndc/struts/results?bt_0=&st_0=&type_8=EXACT&query_8=None+Selected&op_14=eq&v_14=&st_1=&bt_2=&st_2=&bt_1=&bt_10=&st_10=&ge_9=&le_9=&bt_3=&st_3=&type_19=EXACT&query_19=74&op_17=eq&v_17=&bt_20=&st_20=&bt_13=&st_13=&bt_16=&st_16=&bt_6=&st_6=&ge_21=&le_21=&bt_11=&st_11=&ge_22=&le_22=&d=7&t=101650&s=70. Accessed 16 Dec. 2018.
3. Coastal National Elevation Database (CoNED) Project - Topobathymetric Digital Elevation Model (TBDEM) | The Long Term Archive. https://lta.cr.usgs.gov/coned_tbdem. Accessed 6 Dec. 2018.
4. “Indonesia Tsunami Kills Hundreds after Krakatau Eruption.” *BBC News*, BBC, 23 Dec. 2018, www.bbc.com/news/world-asia-46663158.
5. Indonesia Tsunami Worsened by Shape of Palu Bay: Scientists. <https://www.yahoo.com/news/indonesia-tsunami-worsened-shape-palu-bay-scientists-025002225.html>. Accessed 16 Jan. 2019.
6. Kelvin Waves. www.oc.nps.edu/webmodules/ENSO/kelvin.html.
7. Kristina, W. Effective Coastal Boundary Conditions for Tsunami Simulations. 2014. ---. Effective Coastal Boundary Conditions for Tsunami Simulations. 2014.
8. Randall, David A. “The Shallow Water Equations.” Selected Papers, p. 11. Shallow.Pdf. <http://kestrel.nmt.edu/~raymond/classes/ph589/notes/shallow/shallow.pdf> . Accessed 6 Dec. 2018.
9. Synolakis, C. E., et al. “Validation and Verification of Tsunami Numerical Models.” *Pure and Applied Geophysics*, vol. 165, no. 11–12, Dec. 2008, pp. 2197–228. Crossref, doi:10.1007/s00024-004-0427-y. ---. “Validation and Verification of Tsunami Numerical Models.” *Tsunami Science Fmy Years after the 2004 Indian Ocean Tsunami: Part I: Modelling and Hazard Assessment*, edited by Phil R. Cummins et al., Birkhäuser Basel, 2009, pp. 2197–228. Springer Link, doi:10.1007/978-3-0346-0057-6_11.
10. Thacker, William Carlisle. “Some Exact Solutions to the Nonlinear Shallow-Water Wave Equations.” *Journal of Fluid Mechanics*, vol. 107, June 1981, pp. 499–508. Cambridge Core, doi:10.1017/S0022112081001882.
11. Time-Stepping Schemes Review - WikiROMS. https://www.myroms.org/wiki/Time-stepping_Schemes_Review#Forward-Backward_Feedback_.28RK2-FB.29. Accessed 6 Dec. 2018.
12. Tinti, S., and R. Tonini. *The UBO-TSUF D Tsunami Inundation Model: Validation and Application to a Tsunami Case Study Focused on the City of Catania, Italy*. 2013
13. Giachetti, T. ,Karim R., Budianto, K. Tsunami hazard related to a flank collapse of Anak Krakatau Volcano, Sunda Strait, Indonesia, January 2012 Geological Society London
14. Ontowirjo Smith, Nicola. “Indonesia Tsunami: At Least 222 Dead and 843 Injured after Anak Krakatau Volcano Erupts .” *The Telegraph*, Telegraph Media Group, 22 Dec. 2018, www.telegraph.co.uk/news/2018/12/22/indonesia-tsunami-least-20-dead-165-injured-waves-hit-beaches/.
15. Ali, Hani, et al. “APPLICATION OF THE SHALLOW WATER EQUATIONS TO REAL FLOODING CASE.” *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)*, Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece, 2016, pp. 735–49. Crossref, doi:10.7712/100016.1849.10225.

Traffic Model

New Mexico Supercomputing Challenge

Final Report

April 3, 2019

Rishi Tikare Yang

Team Number: 4

School Name: Albuquerque High School

Teacher: Joan Newsom

Executive Summary

Traffic flow in cities affects nearly everyone's lives as our society depends on automobiles for transportation. Making traffic more efficient could have effects ranging from increased productivity of the city's inhabitants, to reducing CO₂ emissions from cars. For my project, I have modeled traffic focusing on the questions: What variables have the greatest impact on traffic flow? How can we make traffic flow more efficient?

My model, developed in NetLogo¹, consists of roads on a rectangular grid. Each road is two lanes, with one going in each direction and stop lights at intersections. My model can control the following variables: stoplight cycle duration, the timing of lights relative to each other, car follow distance, car spawn rate and car speed. My model tracks the percentage of cars waiting at a given time and the average trip duration of the cars in the model. In this report, I will describe my model, show results and discuss the effects and significance of each variable.

The only variable that could approach zero wait time was stoplights cycle duration. The stoplight timing and cycle length could be optimized by timing lights according to the speed and distance between intersection. With this timing, no cars waited after they entered the road system. Theoretically this timing can be applied to any grid-like road system. Car follow distance was only significant after a threshold which was dependant on car density.

My model can be applied to roads in our city to assist in optimizing the road system. My model also can predict how advances in technology, such as self driving cars that can drive in a precise manner, could increase or change efficiency of our roads, and how roads should be designed to most effectively work with self-driving cars.

¹ Wilensky, U. 1999. NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Problem Statement

Almost everyone has to deal with traffic on a daily basis. Slow traffic and badly timed lights are frustrating and waste time from everyone's day. Not only does inefficient traffic waste time, it increases pollution released from vehicles. A typical passenger vehicle emits about 4.6 metric tons of carbon dioxide per year², and it is predicted there will be 281.3 million registered vehicles in the U.S. this year.³ Optimizing roads can help reduce the amount of fuels burned, reduce wasted time of transportation, boost productivity, and save drivers money.

Solution Method

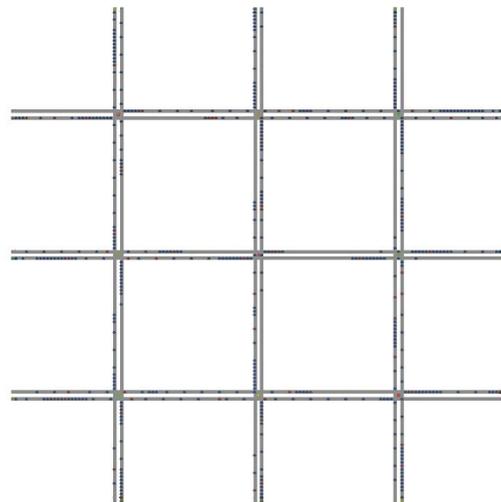
I have developed a model to test which variables affect traffic flow efficiency and to what extent. I have created a NetLogo model that simulates cars driving through a road system. My model is able to control variables of the cars, road, and stop lights, and tracks data on the efficiency of vehicles moving through the system. By systematically changing my variables and analyzing the results, I was able to determine how variables influence efficiency and to what extent.. I created simulations that would collect data on the model, and gradually increase or decrease a variable to chart the change in efficiency. This method proved effective and allowed me to visualize and find trends of a changing variable.

Model Description

² <https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle>

³ <https://hedgescompany.com/automotive-market-research-statistics/auto-mailing-lists-and-marketing/>

My model consists of the a road system (see figure below), stoplights, and cars. Car position and stoplight cycles are updated every tick, which is the unit of time. Each lane of a roads has a set direction that cars follow. At every tick, cars move ahead a set distance which simulates *car speed*. Cars will only move if there are no cars within a given distance in front, which I call *car follow distance*. If a car is stopped at an intersection, it reports it is *waiting*, and if a car approaches from behind a waiting car, it will stop right behind it, and then is also *waiting*. The *percent of cars waiting* is collected and plotted. At every tick, cars have a probability of spawning at the beginning of each road, which I named *car spawn probability*. When a car reaches the end of a road, it is despawned and reports the number of ticks it was alive, or it's *trip duration*. The average trip duration is plotted. Stoplights are set to allow traffic horizontally or vertically for a certain number of ticks, which I call *stoplight cycle duration*. All stoplights in a simulation have the same stoplight cycle duration because all intersections are equidistant from each other. The timing of stoplights switching directions relative to each other can be controlled, and is called *relative stoplight timing*.



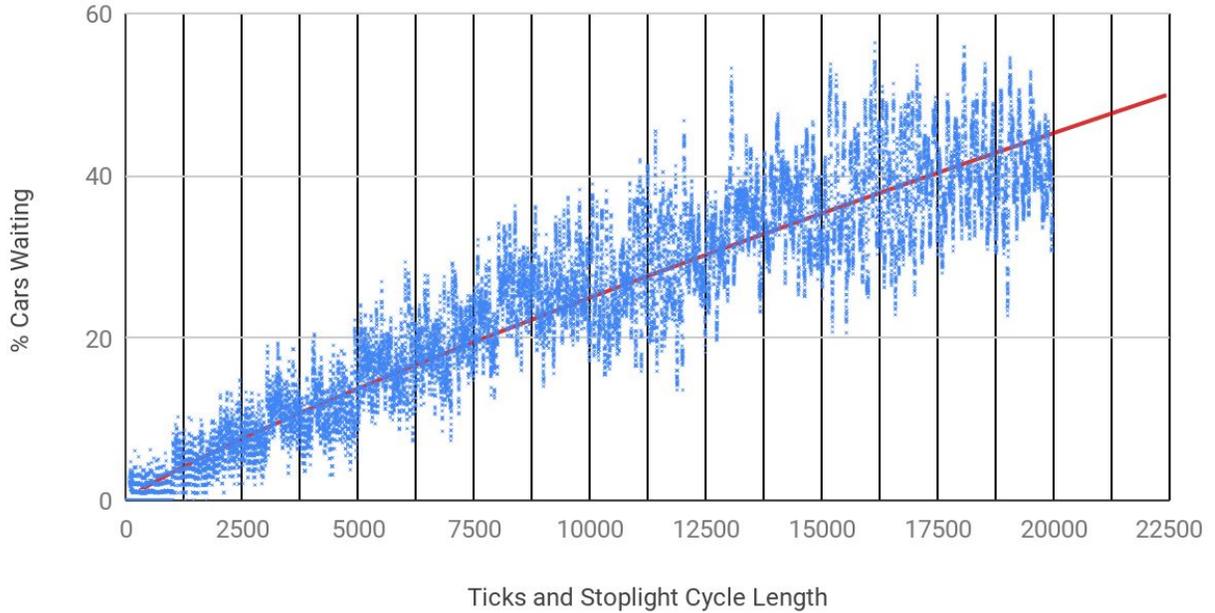
Validation

Cars and drivers follow specific patterns and rules of traffic, and by using agent-based modeling to program agents to follows these same rules, my model can prove accurate and useful in finding trends in traffic flow as a function of model variables.. An example is having cars follow a set distance behind the car ahead, which

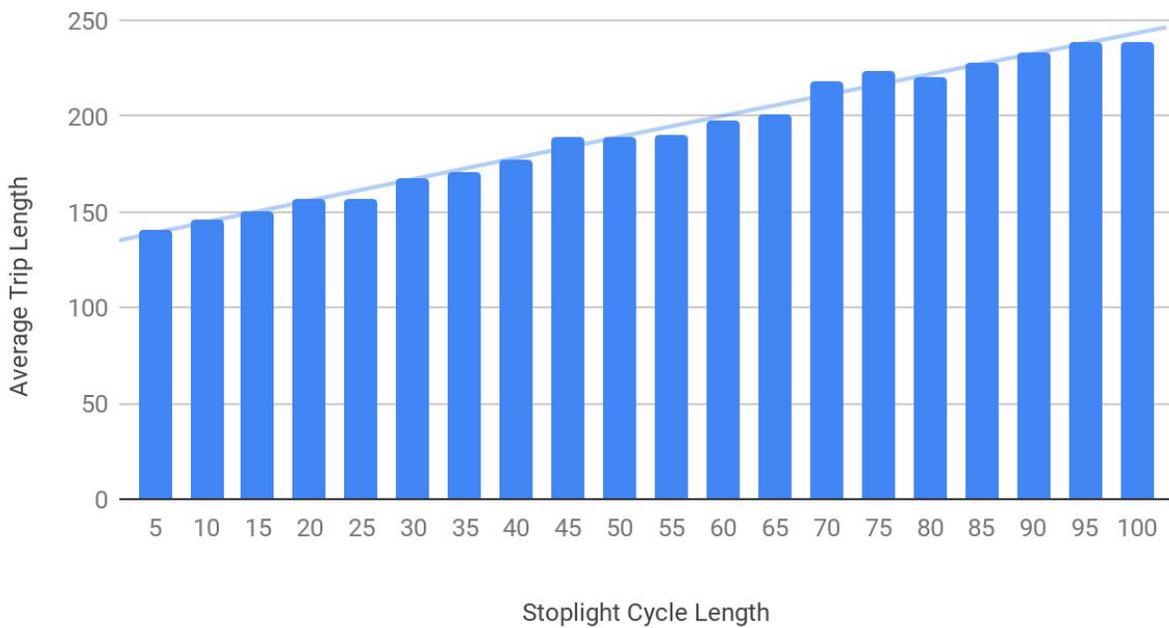
more accurately simulates cars accelerating out of an intersection. I have also considered limitations in my model and tried to account for them while running simulations, as well as taken care to completely isolate variables while testing. An example is during my simulations, the calculated average trip duration of car in my model only averaged cars that completed their trip in the last half of each interval, so that any adjustment period cause by a changed variable does not influence my data. My model has limitations; it only has two lane roads, and does not consider human behavior and variation in driving. A specific limitation is that NetLogo moves each car one at a time, so cars have instantaneous reactions to cars ahead of them and to stoplights. This allows my model to operate without stoplights altogether. This clearly is not how people drive, but perhaps can be applied to self driving cars, where ideally every car in the system can communicate and have accurate predictions of the positions of other cars, and have instantaneous reactions.

Results

Percent Cars Waiting with Varying Stoplight Cycle Duration



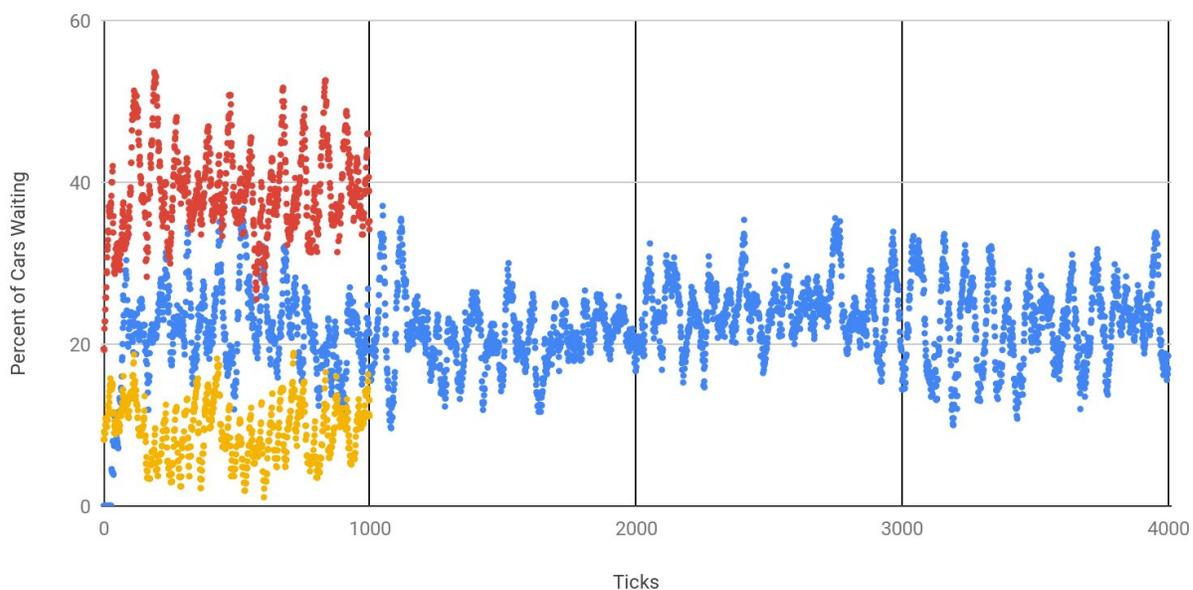
Average Trip Length vs Stoplight Cycle Duration



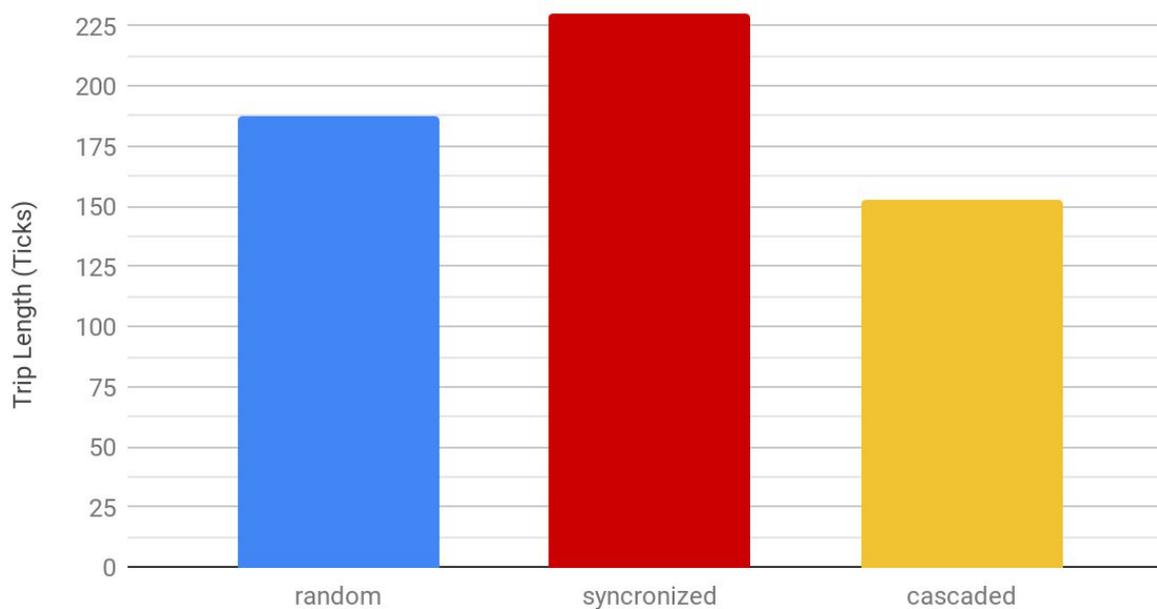
To collect data from my simulation, I used the plot feature in NetLogo that creates a graph and collects data that I specify on intervals I specify. I then exported the data into the spreadsheet software Google Sheets for analysis and visualization. For this simulation, I varied **stoplight cycle duration**. I initialized my model starting the stoplight cycle length to 5 ticks, randomized the relative light timing, and all other variables were held constant. Every 1000 ticks, the stoplight cycle length increased by 5 ticks and the relative light timing was re-randomized. The percent of cars waiting was collected every tick and the average trip duration was found from the cars in the last half of each interval. The data in the graph was an average of a couple simulation runs. The first graph plots the percentage of cars waiting at every tick as blue dots. The red line is a trend line fit to all these data points. The percentage of cars waiting and the variation in cars waiting per tick increases with stoplight cycle duration. The second graph shows the average trip duration correspondingly increased with stoplight cycle. The data points show wait times increased almost linearly with stoplight cycle length, with a slight concavity down. This simulation ignores the fact that in real traffic, with shorter stoplight cycles drivers must more accurately predict when to accelerate and stop. My model assumes all drivers are always paying attention and react instantaneously. This means my model is not an accurate predictor of human drivers at the very short stoplight cycles, but my model might be able to predict how self-driving cars would behave under these conditions. The trend that shorter stoplight cycles reduces wait times is the obvious conclusion.

Percent of Cars Waiting vs. Ticks Comparing Relative Stoplight Timing

Blue(random) Red(Synchronized) Yellow(Cascaded)



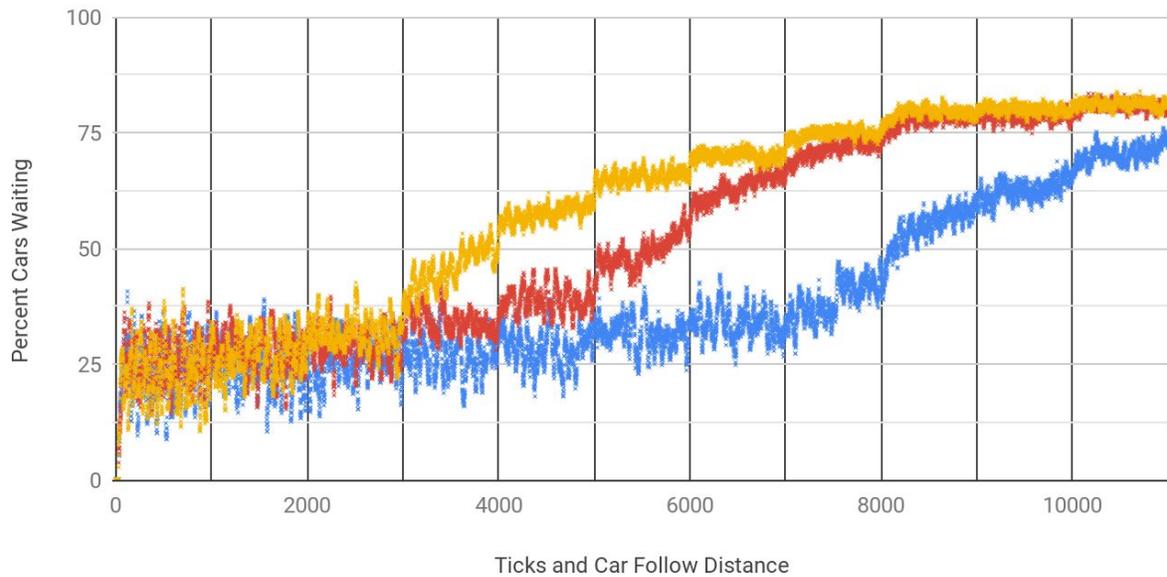
Average Trip Length vs. Relative Stoplight Timing



For the next simulation, I varied the relative **stoplight cycle timing**. The stoplight cycle duration was held constant at 40 ticks for the entire simulation, because this is the most efficient cycle length for cascaded lights. During this simulation, I ran the model for 1000 tick intervals of randomized, synchronized, or cascaded lights. I ran 4 intervals of different random stoplight timings, and found the average trip length over all 4. The randomizing code initialized the stoplights to randomly select between vertical or horizontal traffic, and randomly set the number of ticks -between 0 and the stoplight cycle duration- until it changes directions. This means all the lights cycle with the same cycle duration, but are randomly offset in direction and timing. The synchronizing code initialized the lights to all be in horizontal traffic, and all change direction on the same cycle. The code to cascade the lights initialized the stoplights to allow vertical traffic, and then were timed so that adjacent lights switched 40 ticks later, which is exactly the time it took for a car to travel from one intersection to the other. Because the cycle duration was equal to the travel time between stoplights, by cascading in to the right and down directions, the system it also is cascaded left and up. This simulation shows that with predictable travel times between stop lights, roads can be optimized to have the same stoplight cycle duration, and can be cascaded in all directions. The limitation of my model is again that it ignores human reaction time. At short stoplight cycle durations, human reaction and prediction would most likely be too imprecise and inconsistent to function at maximum efficiency, and human drivers do not have perfectly predictable travel times between stoplights. The most significant conclusion of this test it that it is possible to time lights on a grid pattern of roads to allow drivers to never stop at intersection once they hit a green light. The limitation to this it that cycle required might be too low to be efficient with human drivers, or with very dense traffic.

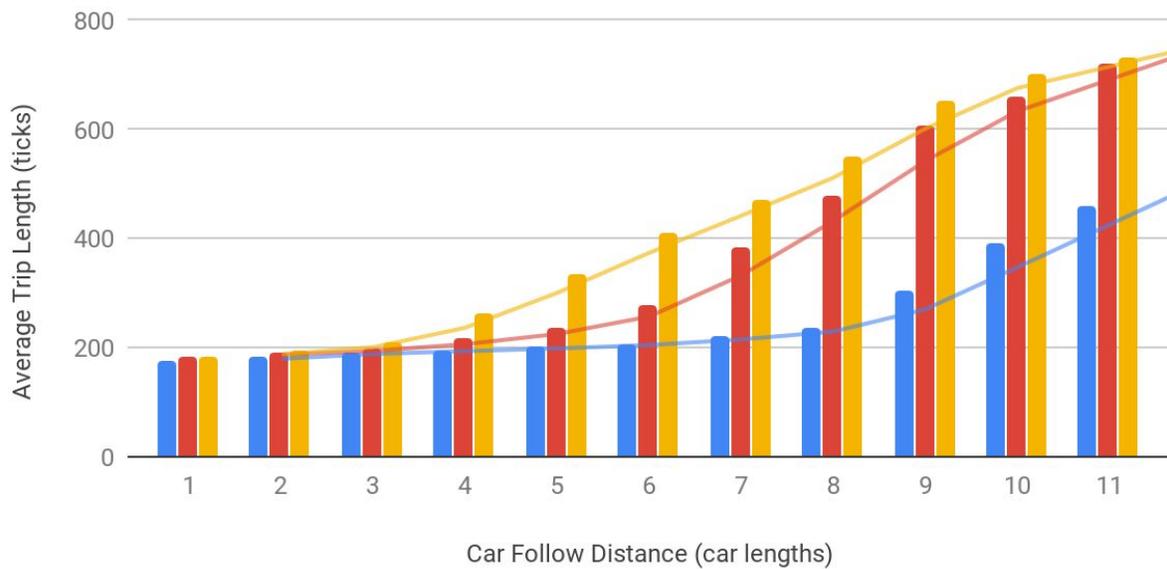
Percent of Cars Waiting with Varying Car Follow Distance

Spawn Probability: Blue (6%) Red (9%) Yellow (12%)



Average Trip Length vs. Car Follow Distance

Spawn Probability: Blue (6%) Red (9%) Yellow (12%)



In the next simulation I varied the **car follow distance**. I initialized the model to a stoplight cycle duration of 50 ticks, initially randomized relative light timing, and set the car spawn probability to 6%. Then, starting at 1, increased the car follow distance every 1000 ticks. Again, I collected the percent of cars waiting per ticks, and the average trip duration of every interval. To ensure the average was only influence by the current variables, I found the average trip duration of cars in the last half of each interval, to give the model time to cycle through the transition periods between varying car follow distances. I ran this simulation at 3 different car spawn probabilities, shown in blue, red, and yellow, with car spawn percentages of 6%, 9%, and 12% respectively. The data suggests that there is a threshold, that depends on the amount of cars on the road, where below the threshold the car follow distance is not very significant, but over the threshold wait times increase significantly with car follow distance. During the blue test (6% car spawn probability), that threshold is around 8 car length, in the red test (9% car spawn probability) the threshold is around 5 car lengths, and in the yellow test (12% car spawn probability) the threshold is around 3 car lengths. There also appears to be a upper limit to how significant car follow distance can be, but I believe this is because my model will only spawn cars as fast as they can be spawned without backing up behind the spawn points. I believe this upper threshold is caused by this limitation of my model. But in real traffic, there is also no reason to drive such a large distance behind the car ahead of someone in the conditions my model simulates.

Conclusions

My model established trends of traffic efficiency for a changing variable, including stoplight cycle duration, relative stoplight light timing, and car follow distance.

Stoplight cycle duration increased wait times almost linearly, with a slight concavity downwards. Stoplight cycle duration was the only variable that had a lower limit of 0 percent of cars waiting, as the stoplight cycle duration approached 0. The range of variation in percent of cars waiting also increased with cycle duration, showing the larger and longer cycles caused by the stoplights cycle. The shortest stoplight cycle durations are not valid for human drivers. My model moves each car one at a time after checking the spaces ahead is not occupied. This means the model operates so that cars react instantaneously and can drive in a precise manner, which humans cannot do reliably. My model may still be valid for self-driving cars, and can be used to optimize stoplights for computer controlled cars that can both react instantaneously and predict other car movement with precision.

Relative stoplight timing had a significant effect on the efficiency of traffic. The three types of light timing my model implements are randomized, synchronized, and cascaded. Randomized timing was fairly consistent, and in simulations where multiple random light timings were tested, their average trip durations had little variation. The percent of cars waiting had varying cycles and ranges, but had similar averages. Synchronized lights were less efficient than randomized lights, and only were more efficient at specific stoplight cycle durations where it mimicked cascaded light timing. Cascaded lights were the most efficient timing of the three at specific stoplight cycle durations. Cascaded lights only required cars that were entering the system to wait for the first green light. When creating the algorithm to cascade lights, I learned that in a grid pattern, it is possible to cascade all roads in both directions if the stoplight cycle duration was equal to the travel time between intersections, or half the travel time.

An interesting insight I reached through creating this algorithm is that for any road system where only two roads meet at any intersection, there is a possible cascading light timing that where no car will wait at any stoplight (except the initial light). This would required the stoplight cycle duration to be equal to the lowest common factor of all travel times between all intersections, and cars to travel between intersections in precise times. This is not always possible for human drivers in city road systems because it would require, most likely, a very short stoplight cycle duration and precision in driving time.

Car follow distance had a distinct threshold; below this threshold the follow distance had little effect on efficiency, but above this threshold follow distance increasingly reduced efficiency. This threshold depended on the number of cars on a road, or the car density. The threshold at which follow distance became significant was lower with higher car density. In my simulation I achieved higher car density by increasing the car spawn rate. In the graph of average trip duration versus follow distance, there is a very shallow slope before this threshold, and after a steep slope. My model shows an upper limit where longer car follow distances would not make traffic more inefficient, and this limit was consistent regardless of car density. This conclusion is not valid because of a limitation of my model, which would stop spawning cars if traffic backed up to spawn points.

Achievements

I created my first numerical model, and used it to draw meaningful conclusions that could be applied to our world. I learned how model-based simulations allow one to gain insights into complex physical systems.

Acknowledgements

I would like to thank my teacher Ms. Newsom who introduced me and guided through the Supercomputing Challenge. I would like to acknowledge the organizers of the Supercomputing Challenge who provide a wonderful service to our community. I would like to thank my parents for helping me proofread my work and general guidance.

Appendix

Code:

```
#main code
```

```
breed[cars car]
```

```
cars-own[go-dist maxspeed acceleration tic tics-alive follow-dist waiting?]
```

```
breed[intersections intersection]
```

```
intersections-own[up? down? right? left? ]
```

```
directed-link-breed[redlinks redlink]
```

```
patches-own[direction direction? intersection?]
```

```
breed[stoplights stoplight]
```

```
stoplights-own[vert? switch maxswitch]
```

```
globals[cars-waiting total-trip-length num-of-cars-despawnd selected-stoplight
```

```
num-of-sums CWS]
```

```
to setup
```

```
  clear-all
```

```
  reset-ticks
```

```
  setup-colors
```

```
  ;set CWS 0 set num-of-sums 0
```

```
  set total-trip-length 0 set num-of-cars-despawnd 0
```

```
  make-roads
```

```
  setup-roads
```

```
initialize-stoplights  
end
```

```
to go  
  if spawn-cars? [spawn-cars]  
  move-cars  
  work-stoplights  
  despawn-cars  
  monitor-cars-waiting  
  ask cars [set tics-alive (tics-alive + 1)]  
  if display-timing? [display-timing]  
  tick  
end
```

```
to setup-colors
```

```
  ask patches [set pcolor white]  
  ask intersections [set color yellow set shape "flag"]
```

```
end
```

```
to make-roads
```

```
  ask patches [set intersection? false]
```

```
  make-intersections 70 1 -70 1  
  make-intersections -70 -1 70 -1
```

```
  make-intersections 70 41 -70 41  
  make-intersections -70 39 70 39
```

```
  make-intersections 70 -39 -70 -39  
  make-intersections -70 -41 70 -41
```

```
  make-intersections 1 -70 1 70  
  make-intersections -1 70 -1 -70
```

```
  make-intersections 41 -70 41 70  
  make-intersections 39 70 39 -70
```

```
  make-intersections -39 -70 -39 70  
  make-intersections -41 70 -41 -70
```

```
end
```

```
to initialize-stoplights
```

```
make-stoplight 40 40
make-stoplight 40 0
make-stoplight 40 -40
make-stoplight 0 40
make-stoplight 0 0
make-stoplight 0 -40
make-stoplight -40 40
make-stoplight -40 0
make-stoplight -40 -40
```

```
ask stoplights
[
  set vert? false
  set maxswitch stoplight-cycle-length
  set switch random maxswitch
]
randomize-lights
end
```

```
to make-stoplight [x y]
  ask patch x y [sprout-stoplights 1 ]
  ask patches with [pxcor > (x - 2) and pxcor < (x + 2) and pycor > (y - 2) and pycor < (y + 2)]
  [set direction "fd"]
end
```

end

```
to make-intersections [x y xx yy]
```

```
  make-intersection x y
  make-intersection xx yy
  connect-intersections x y xx yy
```

end

```
to connect-intersections [x1 y1 x2 y2]
  ask patch x1 y1
  [
    ask intersections-here
    [
      create-redlink-to one-of intersections-on patch x2 y2
    ]
  ]
end
```

```

to make-intersection[x y]
  ask patch x y
  [
    set intersection? true
    set pcolor green
    sprout-intersections 1
  ]
end

```

```

to setup-roads
  ask patches [set direction? false]
  ask intersections
  [
    let x pxcor
    let y pycor
    ask out-link-neighbors
    [
      draw-road x y pxcor pycor
    ]
  ]
end

```

```

to draw-road [x1 y1 x2 y2];[here to there] or from [patch to patch]
  let numer (y2 - y1)
  let denom (x2 - x1)
  let g 1
  set g (GCD numer denom)
  set numer (numer / g)
  set denom (denom / g)

  if denom != 0 or numer != 0;if intersection is not on the same square
  [
    let ver-dir ""
    let hor-dir ""

    ifelse numer > 0 [set ver-dir "up"][set ver-dir "dn"]
    ifelse denom > 0 [set hor-dir "rt"][set hor-dir "lt"]

    let lastx x1
    let lasty y1

    while [lastx != x2 or lasty != y2]
    [

```

```
ask patches with [pxcor <= max list lastx (lastx + denom) and pxcor >= min list lastx (lastx +
denom) and pycor = lasty ];coordinates in the range i want, which is the last patch to the patch
distance away horzo r vert
```

```
[
  set direction hor-dir
  set direction? true
  set pcolor grey
```

```
]
set lastx (lastx + denom);remove form loop
```

```
ask patches with [pycor <= max list lasty (lasty + numer) and pycor >= min list lasty (lasty +
numer) and pxcor = lastx ];coordinates in the range i want, which is the last patch to the patch
distance away horzo r vert
```

```
[
  set direction ver-dir
  set direction? true
  set pcolor grey
```

```
]
set lasty (lasty + numer)
```

```
]
```

```
ask patches with[pxcor = x1 and pycor = y1][set pcolor green]
ask patches with[pxcor = x2 and pycor = y2][set direction? false]
]
end
```

```
to-report GCD [n d]
if n = 0 or d = 0 [report 1]
let num1 (abs n)
loop
[
  if(n mod num1 = 0) and (d mod num1 = 0)
  [
    report num1
  ]
  set num1 (num1 - 1)
]
report 1
end
```

```
;-----End of Setup COmmands Only Run Commands Now-----  
;stoplights:
```

```
to work-intersections;i think this can be removed
```

```
ask intersections  
[  
  let r false  
  let l false  
  let u false  
  let d false  
  ask patch-at 0 1  
  [  
    if direction? = true [ set u true]  
  ]  
  ask patch-at 0 -1  
  [  
    if direction? = true [set d true]  
  ]  
  ask patch-at 1 0  
  [  
    if direction? = true [set r true]  
  ]  
  ask patch-at 0 -1  
  [  
    if direction? = true [set l true]  
  ]  
  set up? u  
  set down? d  
  set right? r  
  set left? l  
]
```

```
end
```

```
to work-stoplights;switch from vert to horizontal
```

```
ask stoplights  
[  
  set maxswitch stoplight-cycle-length  
  if switch <= 0  
  [  
    ifelse vert? = true [switch-horizontal xcor ycor set switch maxswitch]  
  ]  
]
```

```

    [switch-vertical xcor ycor set switch maxswitch]
  ]

  set switch (switch - 1)
]

end
to switch-stoplight [sl]
  ask sl [
    ifelse vert? = true [switch-horizontal xcor ycor]
      [switch-vertical xcor ycor]
  ]
end

to switch-horizontal [x y]
  ask patch (x - 2) (y - 1) [set direction "rt"]
  ask patch (x + 2) (y + 1) [set direction "lt"]
  ask patch (x - 1) (y + 2) [set direction "st"]
  ask patch (x + 1) (y - 2) [set direction "st"]
  ask patch x y [set pcolor green]
  ask stoplights-on patch x y [set vert? false]
end
to switch-vertical [x y]
  ask patch (x - 2) (y - 1) [set direction "st"]
  ask patch (x + 2) (y + 1) [set direction "st"]
  ask patch (x - 1) (y + 2) [set direction "dn"]
  ask patch (x + 1) (y - 2) [set direction "up"]
  ask patch x y [set pcolor red]
  ask stoplights-on patch x y [set vert? true]
end

to make-stoplights [h w trx try];height, width, top right x cor, top right y cor **not in use
  ask patch trx try [sprout-stoplights 1]
  ask patch trx (try - (h - 1)) [sprout-stoplights 1]
  ask patch (trx - (w - 1)) try [sprout-stoplights 1]
  ask patch (trx - (w - 1)) (try - (h - 1)) [sprout-stoplights 1]

  ask stoplights[ask patch-at 0 0 [set direction? true set direction "fd"] ]

end

;-----end of s tolights

```

;move, spawn, kill and cars stuff

to spawn-cars

```
ask patches with[intersection? and pcolor = green]
[
  if random 100 < spawn-percentage and not any? cars-on patch pxcor pycor
  [
    let dir direction
    sprout-cars 1
    [
      set shape "car"
      set color blue
      set go-dist 0
      ifelse random 100 < fast-car-spawn-rate [set maxspeed 2 set acceleration 1 set color red
]] [set maxspeed 1 set acceleration 1]
      set follow-dist car-follow-distance
      set waiting? false
      set tics-alive 0

      if dir = "rt" [facexy (xcor + 1) ycor]
      if dir = "lt" [facexy (xcor - 1) ycor ]
      if dir = "up" [facexy xcor (ycor + 1)]
      if dir = "dn" [facexy xcor (ycor - 1) ]
    ]
  ]
]
```

end

to move-cars

```
let dir ""
ask cars
[

  ;let i 0
  ask patch-at 0 0 [set dir direction]
  if dir = "st" [set waiting? true set go-dist 0]

  if not waiting?
  [ let w? false
  ask cars-on patch-ahead 1 [if waiting?[set w? waiting?]]
  set waiting? w? ]
  ;while [i < spots and dir != "st"];stop
  ;[
```

```

if go-dist < maxspeed [set go-dist (go-dist + acceleration)];used if cars accelerate
if dir = "rt" and check-ahead follow-dist who and not any? cars-on patch-ahead 1;right
[facexy (xcor + 1) ycor fd (go-dist * car-speed-multiplier) set waiting? false]
if dir = "lt" and check-ahead follow-dist who and not any? cars-on patch-ahead 1;left
[facexy (xcor - 1) ycor fd (go-dist * car-speed-multiplier) set waiting? false]
if dir = "up" and check-ahead follow-dist who and not any? cars-on patch-ahead 1;up
[facexy xcor (ycor + 1) fd (go-dist * car-speed-multiplier) set waiting? false]
if dir = "dn" and check-ahead follow-dist who and not any? cars-on patch-ahead 1;down
[facexy xcor (ycor - 1) fd (go-dist * car-speed-multiplier) set waiting? false]

```

```

if dir = "fd" [fd go-dist * car-speed-multiplier] ;forward
;set i i + 1
:]

```

```

]
end

```

```

to turn-cars
ask intersections
[
ask patch-at -1 -1 [ask cars-here []]
]

```

```

end

```

```

to-report check-ahead [x w];[how far ahead to check, the who # of the car]
let move true
let d x
ask car w
[
while [d > 1]
[
if any? cars-on patch-ahead d [ask cars-on patch-ahead d [set move waiting?]]
set d (d - 1)
]
]
]
report move

```

```

end

```

```

to-report check-right [x w]

```

```

end

```

```
to-report check-left [x w]
```

```
end
```

```
to despawn-cars
```

```
ask cars
```

```
[
```

```
ask patch-at 0 0
```

```
[
```

```
if not direction?
```

```
[
```

```
ask cars-here [
```

```
if collect-trip-length? [ set total-trip-length (total-trip-length + tics-alive ) ]
```

```
die]
```

```
if collect-trip-length? [ set num-of-cars-despawnd (num-of-cars-despawnd + 1) ]
```

```
]
```

```
]
```

```
]
```

```
end
```

```
;;;monitor stuff and adjusting variable
```

```
to select-stoplight
```

```
if mouse-down?
```

```
[
```

```
let xmouse mouse-xcor
```

```
let ymouse mouse-ycor
```

```
if any? stoplights-on patch xmouse ymouse
```

```
[
```

```
if selected-stoplight != 0[unlabel-current]
```

```
set selected-stoplight one-of stoplights-on patch xmouse ymouse ;change selsted stolight
```

```
label-current
```

```
]
```

```
]
```

```
end
```

```
;; label the current light
```

```
to label-current
```

```
ask selected-stoplight
```

```
[
```

```
ask patch-at -2 5
```

```
[
```

```
set plabel-color black
```

```
set plabel "current"
```

```

]
]
end
;; unlabel the current light (because we've chosen a new one)
to unlabel-current
  ask selected-stoplight
  [
    ask patch-at -2 5
    [
      set plabel ""
    ]
  ]
end
;-----
to display-timing
  ask stoplights
  [
    let sw switch
    ask patch-at -2 3
    [
      set plabel-color black
      set plabel (word sw)
    ]
  ]
end
to change-timing
  ask selected-stoplight [
    set switch input-switch
    let sw switch
    ask patch-at -2 3
    [
      set plabel-color black
      set plabel (word sw)
    ]
  ]
end

to monitor-cars-waiting

  let summ 0
  ask cars[ if waiting?[set summ (summ + 1)]]
  set cars-waiting summ

;if ticks mod 1000 = 0 [set CWS 0 set num-of-sums 0]
;set CWS (CWS + cars-waiting) set num-of-sums (num-of-sums + 1)

```

```

end
to reset-follow-distance
  ask cars[set follow-dist car-follow-distance]
end

to synchronize-lights
  ask stoplights
  [
    switch-horizontal xcor ycor
    set maxswitch stoplight-cycle-length
    set switch maxswitch
    display-timing
  ]
end
to randomize-lights
  ask stoplights
  [
    ifelse random 2 > 0[switch-horizontal xcor ycor][switch-vertical xcor ycor]
    set maxswitch stoplight-cycle-length
    set switch (random maxswitch)
    display-timing
  ]
end
to cascade-lights;only works if maxswitch is >=40, prioritize down and right traffic
;initialize it
  ask stoplights [switch-vertical xcor ycor]
  ask stoplights-on patch -40 40 [set switch 0]
  ask stoplights-on patch -40 0 [set switch 40]
  ask stoplights-on patch -40 -40 [set switch 80]

  ask stoplights-on patch 0 40 [set switch 40]
  ask stoplights-on patch 0 0 [set switch 80]
  ask stoplights-on patch 0 -40 [set switch 120]

  ask stoplights-on patch 40 40 [set switch 80]
  ask stoplights-on patch 40 0 [set switch 120]
  ask stoplights-on patch 40 -40 [set switch 160]

  display-timing
end

```


Data-Based Approach to Estimating Ice-Shelf Melt Rates

New Mexico Supercomputing Challenge

Final Report

April, 2019

LAHS30

Los Alamos High School

Christie Djidjev

Mentor: Mark Petersen

Contents

| | | |
|----------|--|-----------|
| 1 | Executive summary | 5 |
| 2 | Introduction | 6 |
| 3 | Methods | 9 |
| 3.1 | The data set | 9 |
| 3.2 | Processing the Data | 12 |
| 3.2.1 | Cleaning of faulty data | 12 |
| 3.3 | Geometric aspects of the data | 14 |
| 3.4 | Computing the flux | 17 |
| 3.5 | Deriving and applying the meltrate formula | 18 |
| 3.6 | Coding environment | 19 |
| 4 | Results | 20 |
| 4.1 | Annual melt and flux estimates | 20 |
| 4.2 | Temporal analysis of the results | 20 |
| 4.3 | Comparison with previous results | 22 |
| 5 | Conclusion | 23 |
| 6 | Acknowledgements | 24 |

List of Figures

| | | |
|---|--|---|
| 1 | Ice shelves are sheets of ice stretched above the ocean's surface formed by the flow of glacier down and over the ocean surface. | 6 |
| 2 | Comparison between ice melting in the south and north polar regions. | 8 |

3 Diagram of ice melting caused by sea water under an ice shelf. Downloaded from <http://www.antarcticglaciers.org/2014/05/west-antarctic-ice-sheet-collapsing/>. 9

4 Satellite map of Amery ice shelf. Downloaded from https://www.researchgate.net/figure/Sketch-map-of-the-characterized-Amery-Ice-Shelf-overlaid-on-MOA-image-The-green_fig1_262571908. 10

5 Data is collected by placing mooring at the front of the ice shelf. 11

6 The structure of the data files. 12

7 Error due to replacement of faulty values. (a) Temperature data; (b) Volume data. The error displayed is the absolute value of the difference between the two measurements divided by the absolute value of the original measurement. 14

8 Voronoi diagram example. Each region is the set of points closest to one of the sites, shown as black points. Image downloaded from <http://blog.alexbeutel.com/voronoi/basic-vor.png> in February 2019. 15

9 Voronoi diagrams illustrating average volumes (a) and temperatures (b) per region. Instrument positions are shown as blue dots. y axis gives the depth and the x axis gives the distance from the ice-shelf leftmost point, in meters. 17

10 Voronoi diagrams illustrating smoothed average volumes (a) and temperatures (b) per region using multiple original values for computing approximations. 18

11 Parameters used in the heat-to-melt exchange formula. 20

12 Plot showing melt rate during 2/2001-2/2002. 21

13 Plot showing average temperatures during 2/2001-2/2002. 22

14 This table shows our melt rate compared to previous studies results. . . 23

1 Executive summary

Climate change and rising temperatures are causing increased melting of ice in the polar regions. But the melt rate of the ice shelves of Antarctica is difficult to directly measure as most of it happens not on the surface, but under the ice by the interaction of ice and sea water. The problem addressed in this project is to apply a data-based approach for predicting the ice melt rate from an Antarctic ice shelf based on ocean water measurements made at the front of the ice shelf. Specifically, data recording instruments placed at the ice-shelf front take measurements of the temperature, velocity, and direction of the sea water flow. With the information they obtain, a melt rate can be calculated by comparing the heat of the water going in and out of the ice shelf. The heat difference, which by the law of preservation of energy has to be converted into a different type of energy, goes for converting ice into melt water. We use data for the Amery Ice Shelf derived from the Amisor project of the Australian Antarctic Data Centre. The challenges we face are dealing with defective and incorrect measurements as well as with the low spacial resolution (having a small number of instruments placed on an ice shelf front stretching over hundreds of miles). We develop and analyze several methods for dealing with such challenges. We replace missing or incorrect values by existing values that are closest in time and space and we use Voronoi diagrams to assign each point of the ice shelf front to the closest point with valid measurements. Our prediction for the annual melt is 46.85 gigatons and is consistent with the estimates of other studies that use alternative method and/or data. Our method allows estimating daily values and seasonal variations of the ice melt amounts.



Figure 1: Ice shelves are sheets of ice stretched above the ocean’s surface formed by the flow of glacier down and over the ocean surface.

2 Introduction

Containing global warming and climate change and predicting their consequences is a major challenge of the 21st century. One of the risks is that increased temperatures will melt much of the snow and ice that cover the south and north polar regions. This will cause the global sea level to rise and create floods in major cities across the globe. Oceans are rising 3.2 millimeters each year [7], and the rate is accelerating. Accurately estimating the amount of melt rates at different locations is important, but challenging problem. Antarctica contains the largest body of ice on the Earth, and its ice is melting constantly. If all of the ice in Antarctica melts, the ocean level could rise an extra 60 meters [1].

The continent is surrounded by *ice shelves* (Figure 1), which are bodies of ice that stretch out over the water and most of the melting happens on the bottom surface of these shelves, directly into the ocean. Ice shelves in Antarctica cover an area of 1.561 million km², comparable in size to the Greenland [8]. But estimating the rate of the

melting of the ice shelves in Antarctica is difficult to directly measure as it happens under the ice by the interaction of ice and sea water (Figure 2a), where it is hard to place data collecting instruments. The air temperatures in Antarctica are too low for significant amounts of ice and snow to melt on the surface. In contrast, in the Arctic and in Greenland, Figure 2b, the air temperature can become high enough so that ice and snow melt happens mostly on the surface.

The problem addressed in this project is to model and predict the ice melt rate from an Antarctic ice shelf based on ocean water measurements made at the front of the ice shelf. Specifically, data recording instruments need to be placed at the ice-shelf front that could take measurements of the temperature, velocity, and direction of the sea water flow. With the information they obtain, a melt rate can be calculated by comparing the heat of the water going in and out of the ice shelf, Figure 3. The heat difference, which by the law of preservation of energy has to be converted into a different type of energy, goes for converting ice into melt water. The idea is that if we are able to estimate that difference, we could evaluate the amount of ice melted.

Previous approaches to estimating ice-shelf melting rates include glaciological studies [5, 8], using numerical models [4], and using satellite and radar measurements [2]. Closest to our approach is the study of Herraiz-Borreguero et al. [3], but they complement sea water measurements with previous analysis and knowledge about the water currents under and around the ice shelves. Specifically, they consider modified Circumpolar Deep Water (mCDW), Dense Shelf Water (DSW), and Ice Shelf Water (ISW) currents information.

In contrast, our approach relies only on analysis of the sea water velocity, direction, and temperature at the ice shelf front. It also allows temporal resolution, i.e., to estimate the daily melt rates, while [3] produced a single annual total melt prediction



(a) Antarctic ice shelf extending over the ocean. Downloaded from <https://www.cnn.com/2015/05/16/us/antarctica-larsen-b-ice-shelf-to-disappear/index.html>.



(b) Stream of melted ice on Greenland downloaded from <https://www.businessinsider.com/greenland-approaching-threshold-of-irreversible-melting-2019-1>.

Figure 2: Comparison between ice melting in the south and north polar regions.

only.

Our goal is to make estimations of the ice melted for each day of the year and the total ice melted as well as to analyze seasonal variations. The challenges we face are to find data that contains measurements over an entire shelf front and to deal with defective and incorrect measurements as well as with the low spacial resolution (having a small number of instruments placed on an ice shelf front stretching over hundreds of miles). The specific ice shelf we study is the Amery Ice Shelf, the third largest ice shelf in Antarctica.

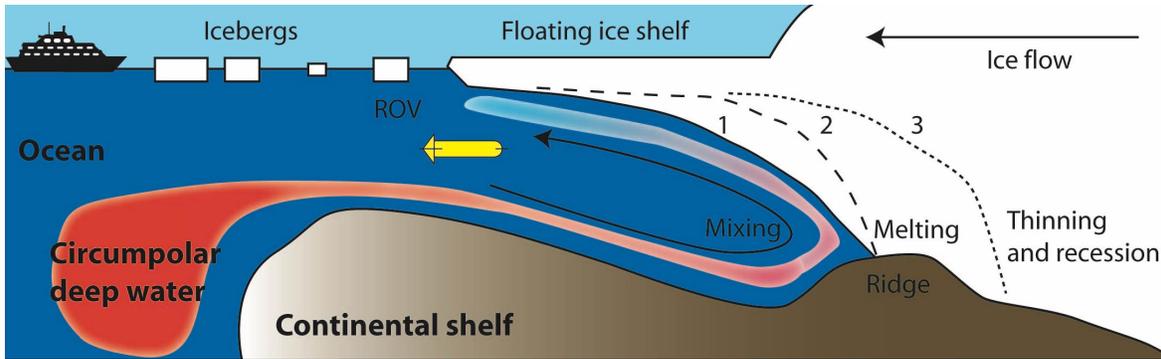


Figure 3: Diagram of ice melting caused by sea water under an ice shelf. Downloaded from <http://www.antarcticglaciers.org/2014/05/west-antarctic-ice-sheet-collapsing/>.

3 Methods

3.1 The data set

The first step is to find the right data. It takes some effort to find suitable measurements, since they should be from an ice-shelf front, from a time period of at least a year of continuous measurements, and with sufficient spacial density. For instance, many measurements are taken from the ice shelf interior, rather than from its front. Some studies collect data from instruments attached to seals, but seals cannot be limited to stay at the ice shelf front. Finally, the data set should include measurements of the velocity, direction, and temperature of the sea water, all three taken at the same times.

We eventually came across the Australian Antarctic Data Centre (AADC) collection, which has data files collected by their moorings at the Amery Ice Shelf front in Antarctica. Amery Ice Shelf is suitable for our approach because of its long and narrow shape, see Figure 4. This helps the model to be more accurate, because the shelf's front is relatively short (about 230 km) compared to its area (roughly 62,000 km²), making the placement of instruments closer together and permitting more accurate estimates. Amery is the third largest ice shelf in Antarctica and the largest in East Antarctica.

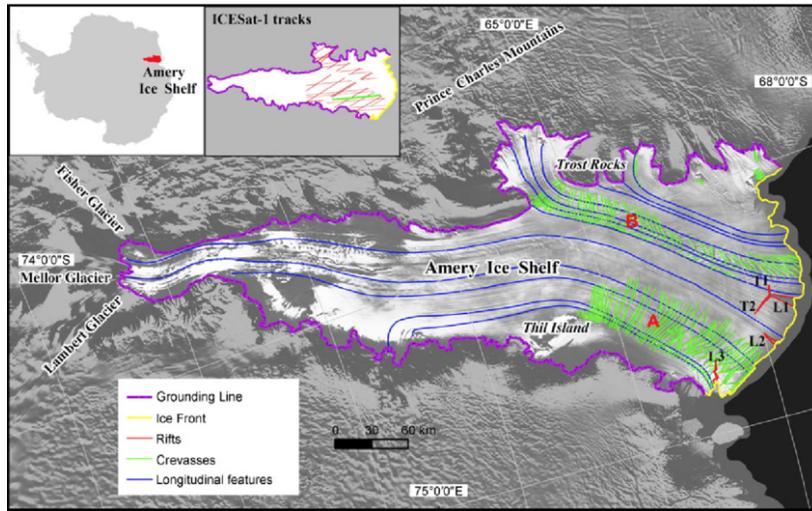
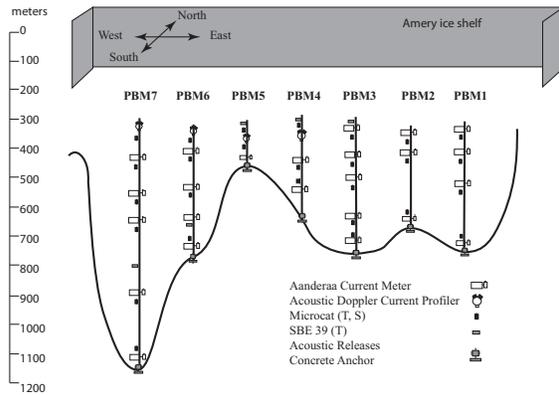


Figure 4: Satellite map of Amery ice shelf. Downloaded from https://www.researchgate.net/figure/Sketch-map-of-the-characterized-Amery-Ice-Shelf-overlaid-on-MOA-image-The-green_fig1_262571908.

The instruments are attached to seven moorings, whose positions are illustrated on Figure 5a. Moorings are long wires that connect the ocean’s floor to the ocean’s surface and which have data instruments attached to them that collect different measurements like temperature, velocity, and salinity. Moorings are dropped by ships, as one can see on Figure 5. The anchors to the moorings must be drilled securely into the ocean floor with precision so that they do not shift.

The moorings were deployed by a ship in February 2001 and collected by another ship a year later, in February 2002. It is relatively expensive to send ships to South Antarctica and there is a very small window of time during the austral summer when the area is free of the thick ice and accessible, which explains why there are only few such data sets available none of them very recent one.

The moorings held two types of devices that recorded the measurements we needed. The first is the RCM type instrument, which records depth, direction, position and



(a) Diagram of seven moorings at ice shelf front [3].



(b) A ship dropping moorings at an ice shelf front. Downloaded from <https://pbs.twimg.com/media/Deen9BEU8AAUr8q.jpg>.

Figure 5: Data is collected by placing mooring at the front of the ice shelf.

velocity, plus other parameters that we don't use (Figure 6a), and which allows us to estimate the amount of water going pass the moorings. Their data files contain about 200,000 lines in total, corresponding to a frequency one record per every hour.

The second instrument type, the microCAT type, measures temperature and salinity (Figure 6b), which will be used later to estimate the amount of heat going in and out of the shelf and the melt rate. The microCAT files had more than 2.5 million lines, much more than the RCM type, because each instrument was set to record measurements every 5 minutes.

```

MOORING : amisor1
INSTRUMENT : Aanderaa rcm 7837x
POSITION : 69 22.014 S 074 38.153 E
INSTRUMENT DEPTH : 735 (m)
INSTRUMENT PRESSURE : 744.0 (dbar)
BOTTOM DEPTH : 750 (m)
MAGNETIC DECLINATION APPLIED: -76.46 (deg)
COMMENT : temperature data bad for whole record; pressure data suspicious
COMMENT :
COMMENT :
COMMENT :
MOORING : amisor7
INSTRUMENT : microcat 1119
POSITION : 68 28.659 S 070 23.118 E
INSTRUMENT DEPTH : 695 (m)
INSTRUMENT PRESSURE : 703.5 (dbar)
BOTTOM DEPTH : 1135 (m)
COMMENT :
COMMENT :
COMMENT :
COMMENT :
decimal_time date and time(UTC) speed dir E comp N comp press temp cond
      yyyy mm dd hh mm (cm/s) (deg.true) (cm/s) (cm/s) (dbar) (deg.C) (mS/cm)
31.02083333 2001 02 01 00 30 -999.00 -999 -999.00 -999.00 -999.00 -999.00 -999.000
31.06250067 2001 02 01 01 30 -999.00 -999 -999.00 -999.00 -999.00 -999.00 -999.000
31.10416800 2001 02 01 02 30 -999.00 -999 -999.00 -999.00 -999.00 -999.00 -999.000
31.14583534 2001 02 01 03 30 -999.00 -999 -999.00 -999.00 -999.00 -999.00 -999.000
31.18750267 2001 02 01 04 30 -999.00 -999 -999.00 -999.00 -999.00 -999.00 -999.000
. . .
46.60441694 2001 02 16 14 30 1.10 341 -0.36 1.04 670.76 -999.00 -999.000
46.64608428 2001 02 16 15 30 1.10 352 -0.15 1.09 670.76 -999.00 -999.000
46.68775161 2001 02 16 16 30 1.10 5 0.10 1.10 663.77 -999.00 -999.000
46.72941895 2001 02 16 17 30 2.54 17 0.74 2.43 670.76 -999.00 -999.000
46.77108628 2001 02 16 18 30 2.13 7 0.26 2.12 663.77 -999.00 -999.000
46.81275362 2001 02 16 19 30 1.10 357 -0.06 1.10 670.76 -999.00 -999.000
46.85442096 2001 02 16 20 30 1.10 356 -0.08 1.10 663.77 -999.00 -999.000
46.89608829 2001 02 16 21 30 1.10 344 -0.30 1.06 663.77 -999.00 -999.000
46.93775563 2001 02 16 22 30 1.10 345 -0.28 1.06 663.77 -999.00 -999.000
46.97942296 2001 02 16 23 30 1.10 352 -0.15 1.09 670.76 -999.00 -999.000
47.02109030 2001 02 17 00 30 1.20 40 0.77 0.92 663.77 -999.00 -999.000
47.06275763 2001 02 17 01 30 3.99 52 3.14 2.46 663.77 -999.00 -999.000
47.10442497 2001 02 17 02 30 4.82 55 3.94 2.76 663.77 -999.00 -999.000
47.14609231 2001 02 17 03 30 4.71 62 4.16 2.21 670.76 -999.00 -999.000
47.18775964 2001 02 17 04 30 3.68 52 2.90 2.27 663.77 -999.00 -999.000
47.22942698 2001 02 17 05 30 1.10 54 0.89 0.65 663.77 -999.00 -999.000
47.27109431 2001 02 17 06 30 1.10 50 0.84 0.71 663.77 -999.00 -999.000
. . .
49.75766421 -1.8846 27.4196 34.5364 19 02 2001 18 11 02
49.76113641 -999.0000 -999.0000 -999.0000 19 02 2001 18 16 02
49.76460861 -1.8849 27.4193 34.5363 19 02 2001 18 21 02
49.76808080 -1.8850 27.4193 34.5364 19 02 2001 18 26 02
49.77155300 -1.8853 27.4191 34.5365 19 02 2001 18 31 02
49.77502520 -1.8857 27.4190 34.5368 19 02 2001 18 36 02
49.77849740 -1.8858 27.4187 34.5365 19 02 2001 18 41 02
49.78196960 -1.8858 27.4183 34.5359 19 02 2001 18 46 02
49.78544180 -1.8860 27.4182 34.5360 19 02 2001 18 51 02
49.78891400 -1.8853 27.4182 34.5352 19 02 2001 18 56 02
49.79238620 -1.8843 27.4186 34.5349 19 02 2001 19 01 02
49.79585840 -1.8840 27.4193 34.5352 19 02 2001 19 06 02
49.79933060 -1.8841 27.4194 34.5355 19 02 2001 19 11 02
49.80280280 -999.0000 -999.0000 -999.0000 19 02 2001 19 16 02
49.80627500 -1.8841 27.4194 34.5355 19 02 2001 19 21 02

```

(a) Example of an RCM data file.

(b) Example of a microCAT data file.

Figure 6: The structure of the data files.

3.2 Processing the Data

3.2.1 Cleaning of faulty data

We need to make sure that every valid data point is included, while the faulty ones are removed. Since the instruments sometimes malfunction and record invalid data, such data elements are replaced by ‘-999.0’ in the dataset. As seen in Figures 6a and 6b, the first several hundred lines record defective values. This must be because as the moorings get dropped, it takes time for the instruments to get to function properly. Later on, most of the record are valid, but occasionally ‘-999’ may appear.

To make use of such occasional defective items of data, we replace the missing elements with values likely to be close to the real ones. An easy to implement solution is to replace each faulty value by the average of the values recorded by that instrument. For this, we used the python function SimpleImputer from the sklearn

library. However, due to seasonal variations, such strategy approximates inaccurately measurements from the summer and the winter. Hence, we implemented our own imputer function, which replaces each missing data with the closest previous value from the same instrument that is a good one. This is justified by the fact that temperature, velocity, and direction are likely to change gradually in time and there is not much difference between measurements taken at roughly the same time.

In order to check the validity of our hypothesis, we analyzed the data to estimate the error of such replacement. We go about this by taking into account the difference of measurements taken at certain time intervals and averaging them. Specifically, we loop over skip values s between 1 and 1000, and for each s we average the absolute value of the difference $t(i) - t(i - s)$ for all valid temperatures t at times i and $i - s$. The results are shown on Figure 7. One can see that the average errors are small, and they increase slowly with the value of s , which means that our hypothesis is correct for the temperature data.

In the second test, we measure in a similar way the error in the approximation of the volume, which depends on both the direction and the velocity. (Computation of the volume is discussed in the next subsection.) We see here a different behavior. The error is relatively high (tens of times higher than in the temperature case), and its growth in time is very slow. This shows that replacing volume data leads to much higher error than replacing temperature data. Unfortunately, faults in volume data are also much more frequent, with about 24432 invalid elements, vs only 3480 for the temperature.

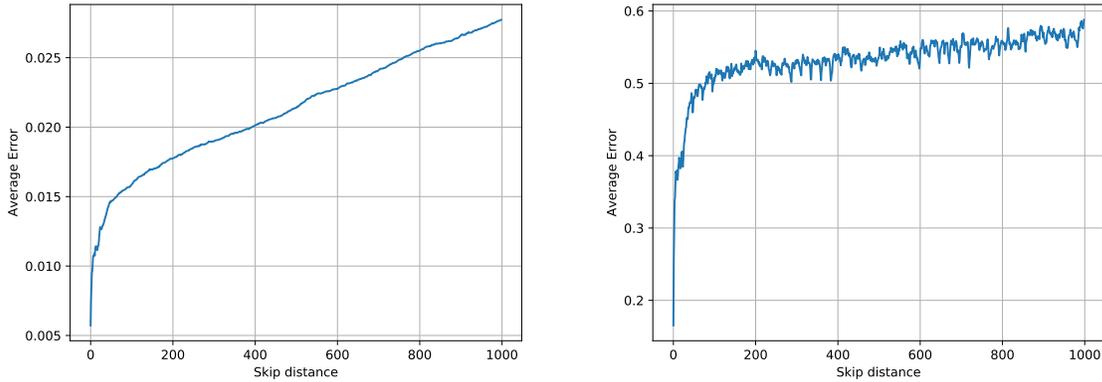


Figure 7: Error due to replacement of faulty values. (a) Temperature data; (b) Volume data. The error displayed is the absolute value of the difference between the two measurements divided by the absolute value of the original measurement.

3.3 Geometric aspects of the data

To use the formulas for the melt rate, we need values for the velocity and direction of sea water movement as well as the temperature at each point of the shelf front. However, we have such values only at several points (where the instruments are). For each of the remaining points, we use the value of its closest instrument location.

This means that we have to compute for each instrument a data structure, called a *Voronoi region*, which gives the set of points closest to the instrument's location. The set of all Voronoi regions makes the *Voronoi diagram* (Figure 8) of the set of points representing the instrument locations. Since the Voronoi diagram covers the points of the entire plane, the regions on the periphery are infinite. So, for the ice shelf front, we add some additional points on the periphery (which we ignore in our analysis) to remove those infinite regions.

In our Voronoi diagram, the x coordinate correspond to the distance to the be-

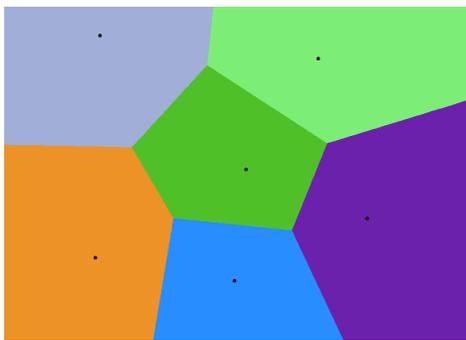


Figure 8: Voronoi diagram example. Each region is the set of points closest to one of the sites, shown as black points. Image downloaded from <http://blog.alexbeutel.com/voronoi/basic-vor.png> in February 2019.

ginning of the front and the y coordinate is the depth of the instrument. In order to find the x coordinates of the moorings, we had to find the coordinate of the beginning of the ice shelf, for which we used Google Maps. The longitude and latitude of the two moorings on either end of the front were shown visually on Google Maps and then clicked the point beside it where the ocean starts and ice ends. These specific locations, we named 'amisorA' and 'amisorB', and their locations were $(-69.394711, 76.007650)$ and $(-68.429035, 70.140795)$. Then, the distance to each mooring was calculated using the Haversine formula. This formula gives the distance between two points on a sphere given their longitudes and latitudes.

As can be observed on the Voronoi diagrams shown in Figures 9, there are both colored and uncolored areas, and the uncolored ones represent the infinite regions that are actually not part of the shelf's front. If compared with Figure 5a, one can recognize the shape of the ocean floor at the bottom of the diagrams. Figure 9 (a) illustrates the average volumes of water going through the corresponding regions, with red colors meaning the water is going in the ice shelf, and blue means it is going out. Figure 9 (b) shows the average temperatures with red for warmer and blue for colder water.

Analyzing the diagrams, we observe that the top regions are much bigger than the other ones due to the fact that the top instruments are positioned relatively deep, at depth of about 500 m. There is a relatively big difference between values in adjacent regions. Those may potentially result in inaccuracy in the model. One way to deal with these issues is to readjust the areas of the Voronoi regions based on the balance of incoming and outgoing volumes of water. Although melting increases the amount of outgoing water, such increase is small fraction of the total sea water. We computed that the disbalance is about 15%, which is not a huge amount, but still shows some inaccuracy.

We first tried a machine learning model, the `LinearRegression` function of the package `sklearn`, to find optimal adjustments of the areas that will minimize the disbalance. However, this didn't work out as many of the optimal scaling coefficients (and areas) turned out to be negative. We then tried the function `Lasso`, which allows to constrain the solution to be nonnegative. This option did produce nonnegative coefficients and achieved a quite small disbalance of only 1.5%, but came with another shortcoming, more than half of the coefficients were zero, meaning that we have to ignore the majority of the data. In fact, with such modification of the areas, the melt rate predicted was three times higher than the expected range, so we had to abandon this idea.

Next we tried get a better approximation of the values for points that don't have instruments at them. The problem with the straightforward Voronoi-regions approach is that we use only the value at single instrument location to assign a value for each such point. This results in big difference between the values close to the boundaries of some of the Voronoi regions, as seen on Figure 9. In the refined approach, we first assign a value at each Voronoi vertex, which is a corner of a Voronoi region, equal to the average of the values of all Voronoi regions it belongs to. Then we triangulate

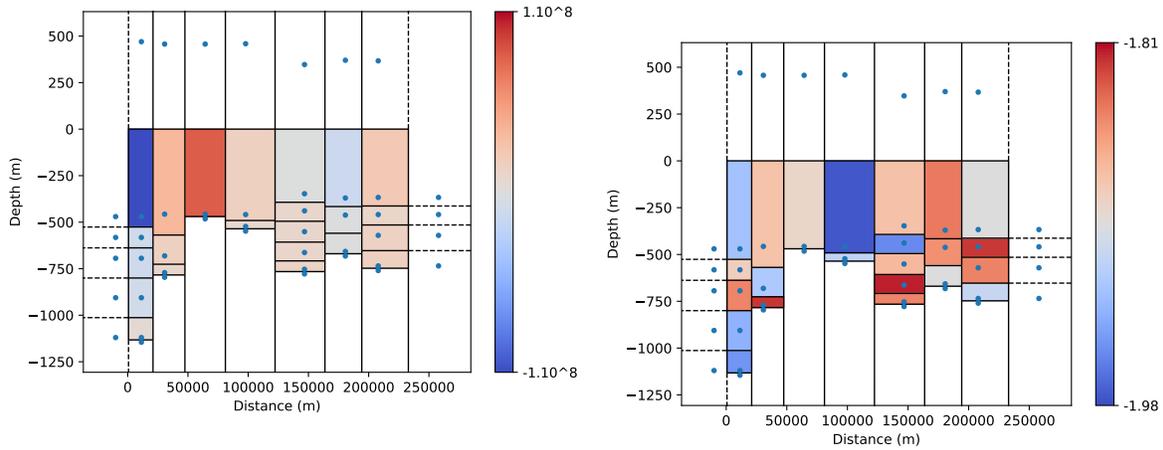


Figure 9: Voronoi diagrams illustrating average volumes (a) and temperatures (b) per region. Instrument positions are shown as blue dots. y axis gives the depth and the x axis gives the distance from the ice-shelf leftmost point, in meters.

each Voronoi region and assign value to that region equal to the average value of its vertices. The result is shown on Figure 10.

3.4 Computing the flux

After processing and cleaning the data, the next phase is to compute the flux, or the amount of seawater passing through the ice shelf front per second. This involves three steps. First we compute the area of each region, which is either a polygon, if the original Voronoi diagram is used, or a triangle, in the case of the smoothed variation. Then we compute the angle of the seawater movement with respect to the direction of the ice shelf front. Finally, we multiply the sine of that angle with the velocity and area of the corresponding region.

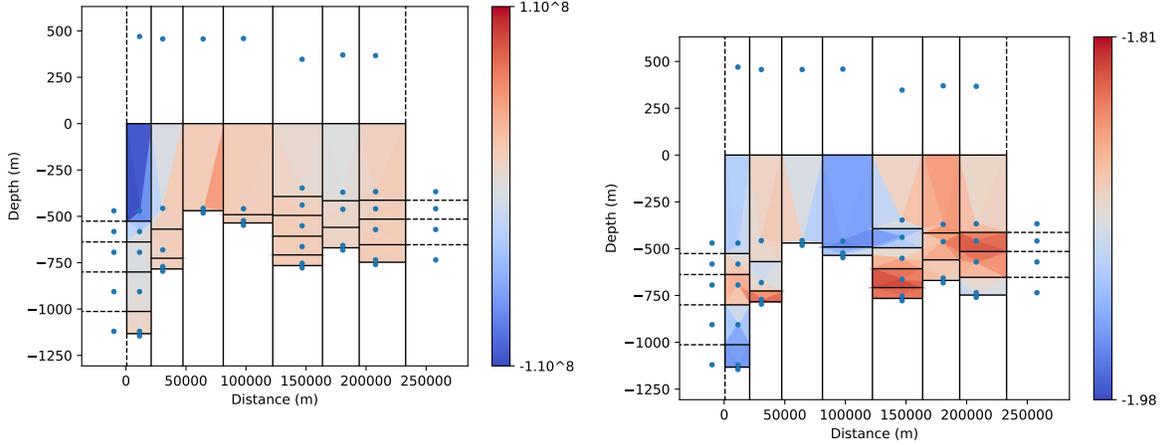


Figure 10: Voronoi diagrams illustrating smoothed average volumes (a) and temperatures (b) per region using multiple original values for computing approximations.

3.5 Deriving and applying the meltrate formula

Once we have estimated the flux, we can find the melt-rate by using the heat exchange formula. It gives the melt rate as a fraction, whose numerator is the heat difference of water going in and out of the ice shelf, and whose denominator is the amount of heat it takes to melt one kilogram of ice (1).

$$m_H = \frac{\text{heat in/out difference (J)}}{\text{heat per kg ice melt (J/kg)}}. \quad (1)$$

For the numerator, our computed flux is needed, the density of the water, the temperature of the the water going in and out of the shelf, and the water's heat capacity, formula (2)

$$H_{diff} = F \rho_{sw} (\theta_{in} - \theta_{out}) c_{sw}. \quad (2)$$

The denominator consists of three components, (3). Since the ice temperature is typically well below the freezing point, we need to calculate the energy it takes to warm ice to its melting point, which is proportional to the difference between the freezing point of sea water θ_f and the measured temperature of ice θ_{ice} , formula (4). The value

of θ_{ice} for the Amery ice shelf was taken from [3]. The value of the ice-capacity constant and other parameters are given in Figure 11. Once the ice is at melting temperature, it needs to consume additional energy to change its state from solid to liquid, which is given by the latent heat of ice, formula (5). Finally, the water leaving the ice shelf has temperature above the freezing point, so additional energy is consumed for warming up the water, resulting in formula (6).

$$H_{melt} = H_{warm_ice} + H_{melt_ice} + H_{warm_sw} \quad (3)$$

$$H_{warm_ice} = c_i(\theta_f - \theta_{ice}) \quad (4)$$

$$H_{melt_ice} = L \quad (5)$$

$$H_{warm_sw} = c_{sw}(\theta_{out} - \theta_f) \quad (6)$$

Replacing (2)–(6) in (1) gives the final heat-to-melt formula (7).

$$m_H = \frac{F \rho_{sw} (\theta_{in} - \theta_{out}) c_{sw}}{c_i(\theta_f - \theta_{ice}) + L + c_{sw}(\theta_{out} - \theta_f)} \quad (7)$$

For the values of the different parameters, given in the table in Figure 11, we used information provided in [3] and [6].

We applied formula (7) using measurements for each hour of the time interval of the study, and summed up the computed volumes of melted ice in order to compute a total for the entire year.

3.6 Coding environment

We used for coding Python language and multiple python libraries including numpy, sklearn, scipy, and matplotlib. We implemented or adapted our own functions for computing the Haversine distance, bearings, angles, areas, for plotting, etc. Our programs contain almost 1000 lines of code. We used Canopy as a python development environment.

| | | |
|----------------|------------------------------|--|
| F | flux | $0.52 \cdot 10^6 \text{ m}^3 \text{ s}^{-1}$ |
| ρ_{sw} | density | 1027 kg m^{-3} |
| θ_{in} | temp of DSW | -1.92° |
| θ_{out} | temp of ISW | -2.09° |
| c_{sw} | SW heat capacity | $4000 \text{ J kg}^{-1} (\text{ }^\circ\text{C})^{-1}$ |
| c_i | Ice heat capacity | $2090 \text{ J kg}^{-1} (\text{ }^\circ\text{C})^{-1}$ |
| θ_f | freezing temp of glacial ice | -2.769°C |
| θ_{ice} | temp of the basal ice | -15°C |
| L | latent heat of ice | $334,000 \text{ J kg}^{-1}$ |

Figure 11: Parameters used in the heat-to-melt exchange formula.

4 Results

4.1 Annual melt and flux estimates

After running our code using the smoothed Voronoi values (Figure 10), we found the water flux to be $6.27 \cdot 10^{12}$ cubic meters per year and the amount of melted ice is 46.85 gigatons per year. In order to compute the depth of ice this corresponds to, we divide the total melt by the ice-shelf area ($60,000 \text{ km}^2$) and use the density of ice (920 kg m^{-3}) [3] and get a value of 0.85 m of ice thickness melted per year.

Using the standard Voronoi representation (Figure 9), we compute an annual melt slightly lower, 45.5 gigatons per year. We assume the other value, 46.85 as likely to be more accurate.

4.2 Temporal analysis of the results

An advantage of our method is that it allows to analyze the flux and melt amounts on a daily basis. When being plotted, temperature and ice melt rates show seasonal changes. In Figure 12, the orange line represents the amount of melt for each day of the

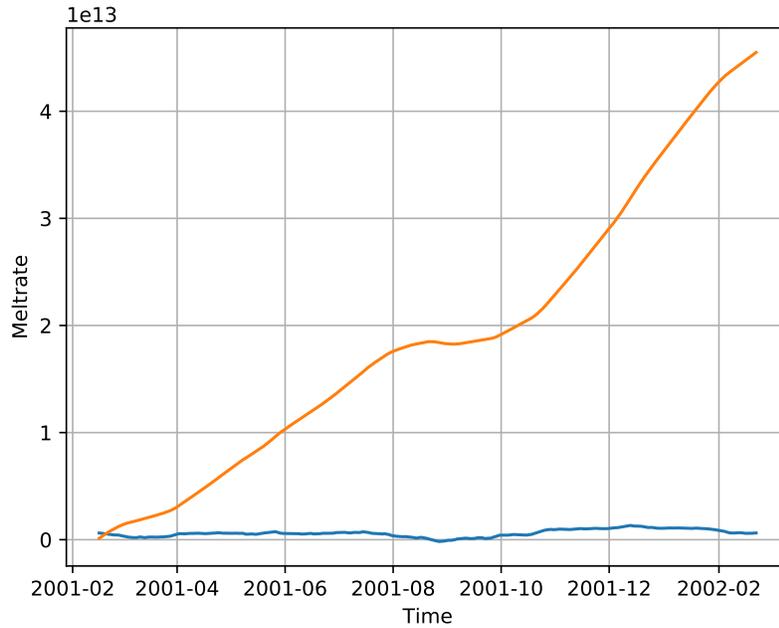


Figure 12: Plot showing melt rate during 2/2001-2/2002.

year of between February 2001 and February 2002, while the green is the accumulation of the melt until the corresponding point in time.

In order to explore the correlation between the sea water temperature and melt rate, we also plotted the temperature. Figure 13 shows the temperature variation for the period of study.

We see that the melt rate is not well correlated with the water temperature, and they both show seasonal variation. First of all, the variation in temperature is relatively small, the difference between the maximum and minimum is only 0.035°C . Note that this temperature is averaged over the entire shelf front, while there is more variation between temperatures in individual sub-regions of the front, as illustrated on Figure 9. Secondly, the warmer water going in will take some time to warm the ice to the melting point, hence there will be a delay on its effect on increasing the melt rate.

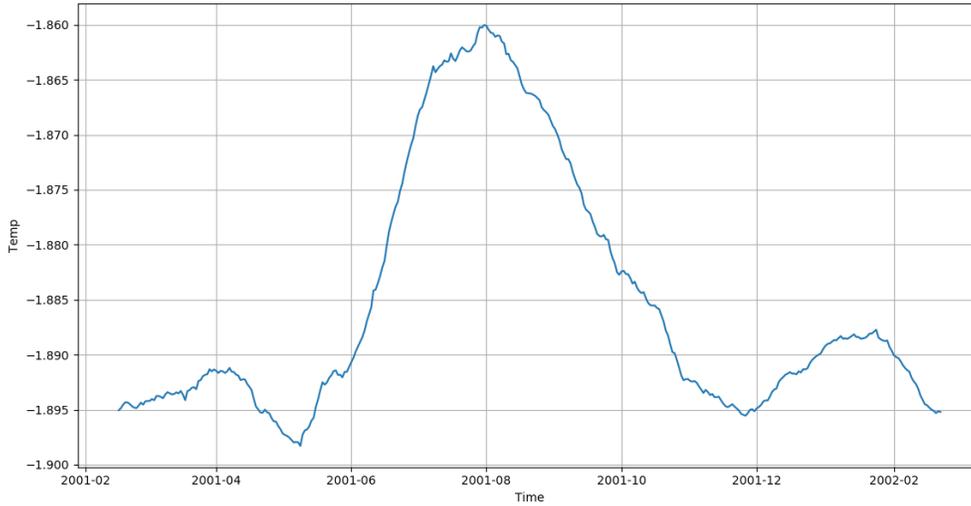


Figure 13: Plot showing average temperatures during 2/2001-2/2002.

Also, we can observe that the sea-water temperature maximum is in August, when it is summer in the northern hemisphere, but winter in Antarctica. This looks strange, but can be explained that there is delay of the warm water to get from the northern warmer regions to Antarctica and the ice shelf. The temperature of the sea water is less dependent on the sun and air temperature, and much more on the ocean currents. Similar delay of the sea water warming has been observed in [3].

4.3 Comparison with previous results

Other authors have used different methods in order to estimate the melt rates at the Amery ice shelf and their results are comparable to ours. Depoorter et al. [3] estimate the annual melt at 39 gigatons per year in average, while Rignot et al. [8] estimate the annual malt rate at 35.5 gigatons per year in average. Both papers use methods based on glaciological analysis, rather than sea water temperature. Yu [9] estimates

the annual rate at 27 gigatons per year using satellite and radar data. Finally, the study of Herraiz-Borreguero et al. [3], which uses the same data set as us, predicts a slightly larger amount, 57.4 gigatons per year.

All of these studies can agree that ice is melting from Amery in dangerously large amounts, putting in risk the health of the planet.

| Study | Annual ice-melt amount |
|------------------------|-------------------------|
| Here | 46.85 gigatons per year |
| Herraiz-Borreguero [3] | 57.4 gigatons per year |
| Deporter [2] | 39 gigatons per year |
| Rignot [8] | 35.5 gigatons per year |
| Yu [9] | 27 gigatons per year |

Figure 14: This table shows our melt rate compared to previous studies results.

5 Conclusion

We developed tools that allow the analysis of noisy data of sea water measurements from an ice shelf front that allow one to estimate daily and annual melt rates and applied it to data from the Amery ice shelf. Our predictions are consistent with the estimates of other researchers using different data and/or different methods.

One of the limitation of this approach is that we don't have measurements for each point of the ice shelf, but approximate large regions by measurements at a single point or a small number of points, leading to potential inaccuracies. Nevertheless, our results are quite accurate when compared to estimates given by other scientists. Another limitation is that we have to use data from 2001-2002, but more recent data

is not currently available. Finally, it is hard to positively validate the results since the real amount of ice melt is not known. However, our results are consistent with the predictions of the other studies.

The next step of this project could be computing the meltrate based on salinity. This would give alternative estimates and one can check how well they correlate with the heat-exchange predictions. The idea of such approach is that one can measure the salinity of the water going in and out and comparing how much the salinity has dropped, which can be used to give an estimate of the fresh water added as a result of the ice melt.

Another direction to continue this work is to use a data-based approach based on alternative data sources such as satellite data.

6 Acknowledgements

I would like to thank my mentor Mark Petersen for inspiring me to do this project and providing me with feedback and advice on the direction of the research, and Hristo Djidjev for helping with the programming and with directions about how to present my work.

References

- [1] National Snow & Ice Data Center. Quick facts on ice sheets. <https://nsidc.org/cryosphere/quickfacts/icesheets.html>. Assessed 3/2/2019.
- [2] M. A. Depoorter, J. L. Bamber, J. A. Griggs, J. T. M. Lenaerts, S. R. M. Ligtenberg, M. R. van den Broeke, and G. Moholdt. Calving fluxes and basal melt rates of antarctic ice shelves, Sep 2013.

- [3] L. Herraiz-Borreguero, J. A. Church, I. Allison, B. Peña-Molino, R. Coleman, M. Tomczak, and M. Craven. Basal melt, seasonal water mass transformation, ocean current variability, and deep convection processes along the Amery Ice Shelf calving front, East Antarctica. *Journal of Geophysical Research: Oceans*, 121(7):4946–4965, 2016.
- [4] Stanley S. Jacobs and Ray F. Weiss. *Ocean, ice, and atmosphere : interactions at the Antarctic continental margin / Stanley S. Jacobs and Raymond F. Weiss, editors*. American Geophysical Union Washington, D.C, 1998.
- [5] A. Jenkins and C.S.M. Doake. Ice-ocean interaction on ronne ice shelf, antarctica. *Annals of Glaciology*, 14:341–341, 1990.
- [6] W. M. J. Lazeroms, A. Jenkins, G. H. Gudmundsson, and R. S. W. van de Wal. Modelling present-day basal melt rates for Antarctic ice shelves using a parametrization of buoyant meltwater plumes. *The Cryosphere*, 12(1):49–70, 2018.
- [7] NASA. Sea level. <https://climate.nasa.gov/vital-signs/sea-level>. Assessed 3/1/2019.
- [8] E. Rignot, S. Jacobs, J. Mouginot, and B. Scheuchl. Ice-shelf melting around Antarctica. *Science*, 341(6143):266–270, 2013.
- [9] Jaehyung Yu, Hongxing Liu, Kenneth C. Jezek, Roland C. Warner, and Jiahong Wen. Analysis of velocity field, mass balance, and basal melt of the Lambert Glacier–Amery Ice Shelf system by incorporating Radarsat SAR interferometry and ICESat laser altimetry measurements. *Journal of Geophysical Research: Solid Earth*, 115(B11), 2010.

Detecting Shocks Waves with Artificial Intelligence

Overview

In this project, I want to learn how to code a program to detect shock waves using artificial intelligence (AI). The problem that I want to solve is finding a way to make the code as efficient as possible while still being super sensitive and still working as intended. The reason I picked this topic is because I wanted to do something with coding and AI. I think 2,500 training iterations will be the best number of times to train the AI algorithm. The reason I think this is a good number is because 2,500 won't take a long time to run in the code and it would give it a lot of training iterations. My hypothesis for the best number of training iteration is from previous times running the AI coding and knowledge of the code. I am hoping to learn more about coding and AI from this project.

Conclusion

My conclusion is that my hypothesis is wrong. The optimal number of training iterations was 5,000, not 2,500.

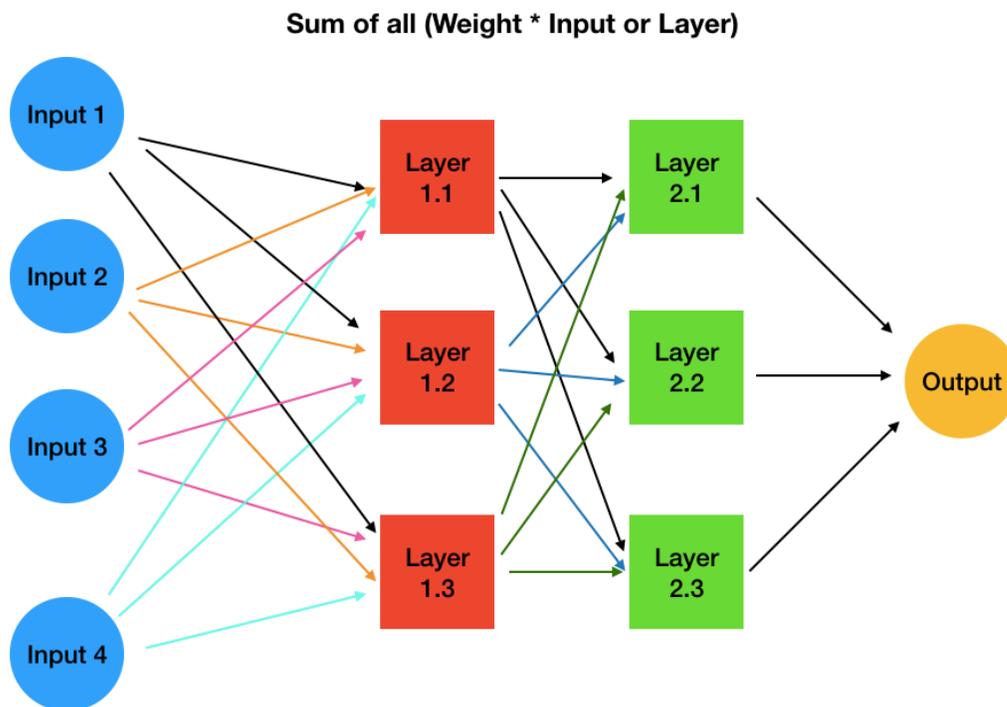


Fig: How AI works

Table 1

| y | x1 | y | x2 | y | x3 | y | x4 | | Smooth |
|---|------|---|------|---|------|---|------|--|--------|
| | 1 | | 1 | | 0 | | 0 | | 0 |
| | 0 | | 0.25 | | 0.5 | | 1 | | 1 |
| | 0 | | 0 | | 1 | | 1 | | 0 |
| | 0.75 | | 0.5 | | 0.25 | | 0 | | 1 |
| | 0 | | 0 | | 0.25 | | 0.75 | | 1 |
| | 0 | | 1 | | 1 | | 0.5 | | 1 |
| | 0 | | 0 | | 1 | | 0 | | 0 |
| | 0 | | 1 | | 1 | | 0 | | 0 |
| | 1 | | 0 | | 0 | | 1 | | 0 |
| | 1 | | 1 | | 0 | | 1 | | 0 |
| | 0.5 | | 0.5 | | 0.5 | | 0.5 | | 1 |

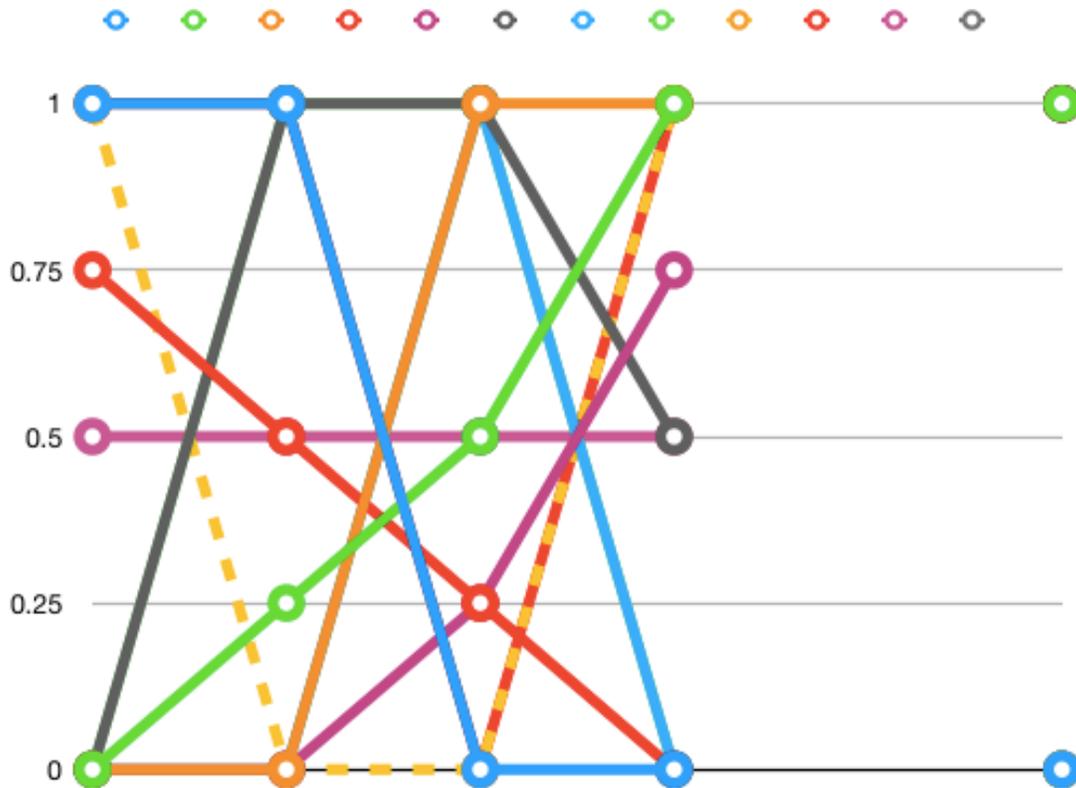
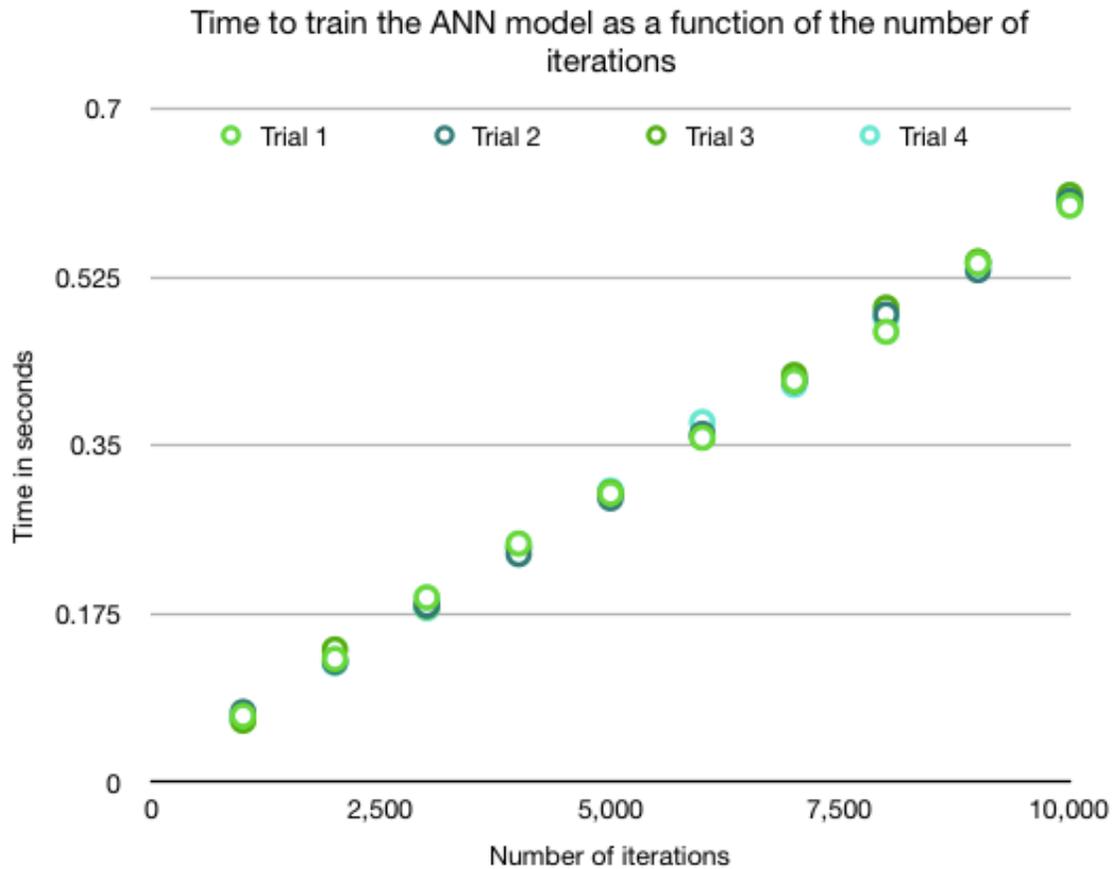


Fig: The training data for the AI code and plots of the data

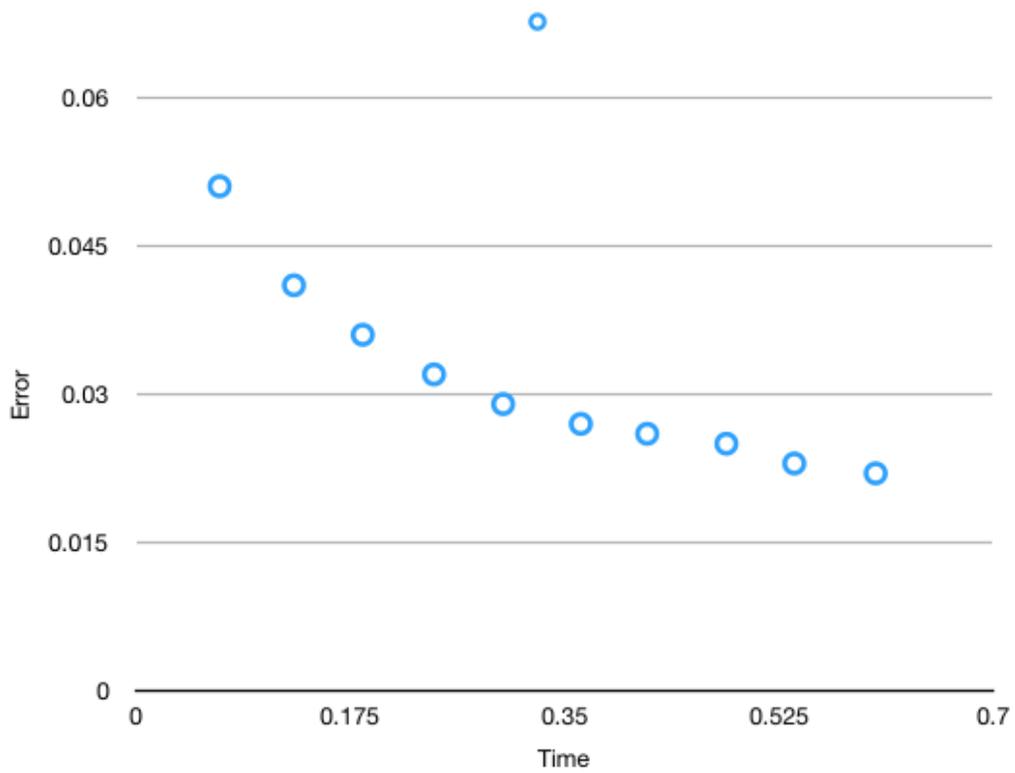
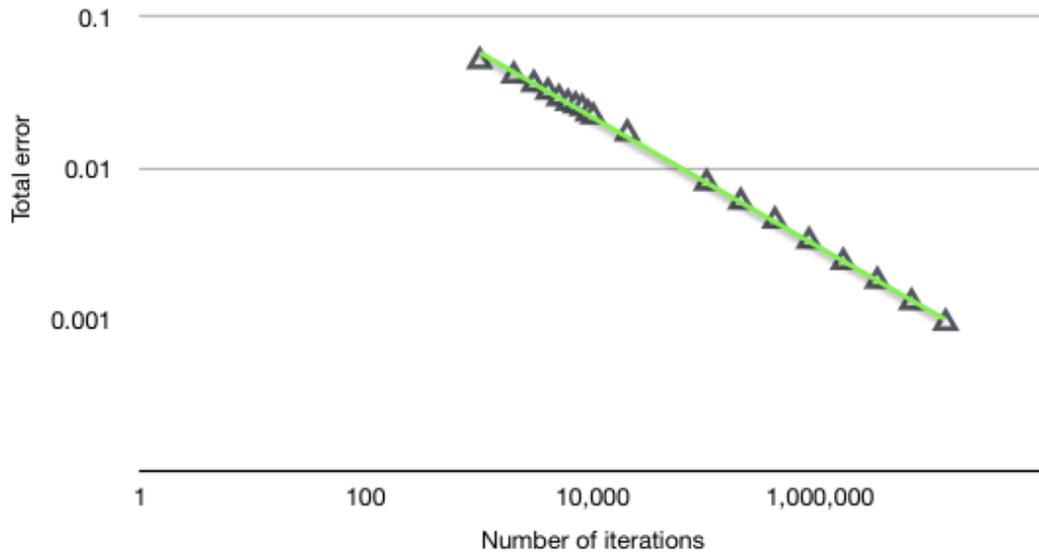
| | Trial 1 | Trial 2 | Trial 3 | Trial 4 | | | |
|---------------------|---------|---------|---------|---------|---------------------|--------------|-------|
| Number of trainings | Time | Time | Time | Time | Error at predicting | average time | Error |
| 1,000 | 0.069 | 0.073 | 0.064 | 0.066 | 0.051 | 0.068 | 0.051 |
| 2,000 | 0.128 | 0.125 | 0.138 | 0.124 | 0.041 | 0.12875 | 0.041 |
| 3,000 | 0.192 | 0.184 | 0.183 | 0.181 | 0.036 | 0.185 | 0.036 |
| 4,000 | 0.248 | 0.237 | 0.243 | 0.245 | 0.032 | 0.24325 | 0.032 |
| 5,000 | 0.300 | 0.295 | 0.301 | 0.303 | 0.029 | 0.29975 | 0.029 |
| 6,000 | 0.358 | 0.362 | 0.359 | 0.374 | 0.027 | 0.36325 | 0.027 |
| 7,000 | 0.417 | 0.417 | 0.423 | 0.413 | 0.026 | 0.4175 | 0.026 |
| 8,000 | 0.468 | 0.486 | 0.493 | 0.482 | 0.025 | 0.48225 | 0.025 |
| 9,000 | 0.539 | 0.532 | 0.541 | 0.539 | 0.023 | 0.53775 | 0.023 |
| 10,000 | 0.599 | 0.604 | 0.610 | 0.604 | 0.022 | 0.60425 | 0.022 |
| 20,000 | | | 1.170 | | 0.017 | 1.170 | |
| 100,000 | | | 6.047 | | 0.008 | 6.047 | |
| 200,000 | | | 11.840 | | 0.006 | 11.840 | |
| 400,000 | | | 24.608 | | 0.0045 | 24.608 | |
| 800,000 | | | 49.624 | | 0.0033 | 49.624 | |
| 1,600,000 | | | 93.894 | | 0.0024 | 93.894 | |
| 3,200,000 | | | 189.536 | | 0.0018 | 189.536 | |
| 6,400,000 | | | 373.83 | | 0.0013 | 373.83 | |
| 12,800,000 | | | 764.178 | | 0.00095 | 764.178 | |

Fig: The results from using the AI code



Total error at predicting the output values for 4 data sets as a function of the number of iterations used to train the ANN model

$$y = 1.1283x^{-0.43}$$
$$R^2 = 0.999$$



Code

three inputs to a N-node hidden layer to another N-node hidden layer to a single output

There will be N weights going in, then N weights and then N weights going out of hidden layer

```
import numpy as np
import time
```

```
def sigmoid(x):
    return 1.0/(1+ np.exp(-x))
```

```
def sigmoid_derivative(x):
    return x * (1.0 - x)
```

```
class NeuralNetwork:
    def __init__(self, x, y, w0, w1, w2):
        self.input = x
        self.weights0 = w0
        self.weights1 = w1
        self.weights2 = w2
        self.y = y
        self.output = np.zeros(self.y.shape)
```

```
def feedforward(self):
    self.layer0 = sigmoid(np.dot(self.input, self.weights0))
    self.layer1 = sigmoid(np.dot(self.layer0, self.weights1))
    self.output = sigmoid(np.dot(self.layer1, self.weights2))
```

```
def backprop(self, eta):
    # application of the chain rule to find derivative of the loss function with respect to
    weights2 and weights1
```

```
    D2 = 2*(self.y - self.output) * sigmoid_derivative(self.output)
    d_weights2 = np.dot( self.layer1.T, D2)
```

```
    D1 = np.dot(D2, self.weights2.T) * sigmoid_derivative(self.layer1)
    d_weights1 = np.dot( self.layer0.T, D1 )
```

```
    D0 = np.dot(D1, self.weights1.T) * sigmoid_derivative(self.layer0)
    d_weights0 = np.dot( self.input.T, D0 )
```

```

# update the weights with the derivative (slope) of the loss function
self.weights0 += eta*d_weights0
self.weights1 += eta*d_weights1 # add \eta in front
self.weights2 += eta*d_weights2

# This is the main program
if __name__ == "__main__":

    # the input values for training
    X = np.loadtxt("X.txt", delimiter=",")
    print(X)

    # the desired outputs
    Yrow = np.loadtxt("Y.txt")
    Y = Yrow.reshape(X.shape[0],1)
    print(Y)

    # initialize the weights
    size = 3 # the number of nodes in the hidden layer

    # to generate input values for the weights
    #W0 = np.random.rand(X.shape[1],size)
    #W1 = np.random.rand(size,size)
    #W2 = np.random.rand(size,1)

    W0 = np.loadtxt("w0.txt", delimiter=",")
    W1 = np.loadtxt("w1.txt", delimiter=",")
    readW2 = np.loadtxt("w2.txt", delimiter=",")
    W2 = readW2.reshape(size,1)

    # create the neural network
    nn = NeuralNetwork(X,Y,W0,W1,W2)

    eta = 1.0

    # calculate start time
    start_time = time.time()

```

```

# train the neural network
for i in range(1000):
    nn.feedforward()
    nn.backprop(eta)

# calculate end time
end_time = time.time()

# list the outputs
print(nn.output)

# time to run the code
print("Elapsed time was %g seconds" % (end_time - start_time))

# write weights to a file
np.savetxt("w0out.txt", W0, fmt="%3.16f", delimiter=",")
np.savetxt("w1out.txt", W1, fmt="%3.16f", delimiter=",")
np.savetxt("w2out.txt", W2, fmt="%3.16f", delimiter=",")

# use the neural network on data
dataX = np.array([[1,0.7,.3,0],
                  [1,0.8,0,0],
                  [0.2,0.4,0.8,1.0],
                  [0,0,0.9,1.0]])

# we should get these values
dataY = np.array([[1],
                  [0],
                  [1],
                  [0]])
datann = NeuralNetwork(dataX,dataY,W0,W1,W2)
datann.feedforward()
print(datann.output)

Error = np.abs(datann.output-dataY)
print('error =')
print(Error)
print(np.sum(Error))

```

References

1. ai.google
2. <https://www.instructables.com/id/Build-Your-Own-AI-Artificial-Intelligence-Assistant/>
3. images.google
4. youtube.com
5. <https://www.udacity.com/course/ai-programming-python-nanodegree--nd089>

Using QAOA to Solve NP-Hard Problems on NISQ Computers

New Mexico Supercomputing Challenge Final Report
Los Alamos High School, Team 29

Elijah Pelofske

April 2019

Abstract

NP-Hard problems are a class of optimization problems which can be solved in exponential time in the best case, which is a significant obstacle because problems of this type have many important applications, such as bioinformatics. Quantum computing has the potential to drastically increase the speed at which important problems, such as NP-Hard optimization problems, can be solved. Most quantum algorithms which can solve important and sufficiently large problems require fault tolerant qubits, and high qubit number and connectivity. Unfortunately the best quantum computers to date are referred to as Noisy Intermediate-Scale Quantum (NISQ) computers, and have low qubit number and connectivity. Therefore, in order to apply quantum computers to useful applications, quantum algorithms which can be used on cloud accessible NISQ devices must be implemented. In this case, the approximation hybrid quantum-classical optimization algorithm QAOA (Quantum Approximate Optimization Algorithm) will be tested because it can be implemented on currently accessible NISQ computers. As an additional comparison, an adiabatic quantum annealer device (DWave 2000Q) is compared. The viability of using NISQ computers to solve industry relevant NP-Hard problems will be investigated using QAOA in comparison to classical solvers and a DWave 2000Q device.

Contents

| | | |
|----------|-----------------------------|----------|
| 1 | Introduction | 2 |
| 1.1 | Quantum Computing | 3 |
| 1.2 | Technical Details | 4 |
| 1.2.1 | Software | 4 |
| 1.2.2 | Hardware | 5 |
| 1.3 | QAOA | 6 |
| 1.4 | D-Wave | 7 |

| | | |
|----------|---|-----------|
| 1.5 | Background and Related Research | 7 |
| 2 | Problem | 7 |
| 2.1 | NP-Hard Problems Tested | 8 |
| 2.1.1 | Maximum Clique | 8 |
| 2.1.2 | Maximum Cut | 9 |
| 3 | Experimental Procedures | 9 |
| 4 | Results | 11 |
| 5 | Analysis | 12 |
| 6 | Conclusion | 13 |
| 7 | Acknowledgements | 16 |

1 Introduction

Quantum computers offer the potential to solve many industry relevant problems in areas such as chemistry, artificial intelligence, optimization, and cryptography. All of these applications are based on quantum algorithms, which have been created over the past roughly 25 years, including the most well known of which are Shors Algorithm [17] and Grover Search Algorithm [6]. Cloud accessible Noisy Intermediate-Scale Quantum (NISQ) computers are, at present, the current state of quantum computing in creating fully scalable and fault tolerant gate based quantum computers. In general, NISQ devices cannot run most quantum algorithms on problems of any meaningful size, largely due to limited connectivity, small number of qubits, and high gate and measurement error rate. These limitations of NISQ devices lead to restrictions on embeddable problem size, and shallow quantum circuit depth.

The Quantum Approximation Optimization Algorithm (QAOA) [13] is specifically designed to run in polynomial time on NISQ devices, and find optimal (and therefore possibly optimum) solutions to optimization problems. This algorithm is therefore appealing as a means to solve critical optimization problems which classically take exponential computational complexity to solve exactly.

The structure of this paper is as follows; the introduction outlines the background and details of this area of research, as well as overall project objectives. Section 2 defines and particular problem which will be considered, as well as their relevance. Section 3 defines the experimental setup and procedures. Section 4

describes the various results from the experiments, and section 5 analysis the specifics of these results, and finally we conclude with section 6.

1.1 Quantum Computing

The concept of quantum computing is based on the idea of a quantum bit (qubit). Qubits are typically small loops of superconductive material, which have been cooled to 15 millikelvin. Microwave pulses directed at specific points on these small circuits can apply quantum "gates" or unitary matrices to these qubits, which causes a change in the wavefunction of the qubit. If a qubit is not measured (i.e. it has zero decoherence), then it is in a superposition between two values (0 and 1). Once measured, that value "collapses" into one of those two. However, if we took a statistically significant series of measurements, we would begin to see the statistical distribution of the superposition of the qubit. For example we might see that 70 percent of the time the qubit is in the 1 state, and 30 percent of the time it is in the 0 state.

Next, once we add multiple qubits and multi qubit gate connectors (in this case, Controlled-NOT gate), we can begin to look at quantum entanglement. This is the idea that multiple qubits can have correlated state values - where if we measure one qubit on an imaginary fault tolerant quantum computer, after applying some specific gates, the state of the other qubits can be determined. After some gates have been applied, the final state can be measured by some classical sensors, and thus we get a binary string out. That binary string can be one of 2^N states, where N is the number of qubits on the quantum computer.

The primary properties of NISQ devices are limited qubit number and connectivity, as well as relatively high decoherence (measurements from thermal fluctuations in the environment), which means high error rates with gate and measurement operations. As more gates are applied in a given circuit, these error rates combine, which leads to shallow usable circuit depth (i.e. only a few gate operations can be performed in a single run). For all NISQ devices, multiple runs (for example up to 8192 runs) can be measured, in order to verify the results are consistent.

Due to the computational speed ups offered by quantum computers, many companies are becoming interested in investing in research and development on quantum computers. Dwave was the first to begin working on quantum computing nearly two decades ago. Many large technology companies such as Intel, Microsoft, Google, and Honeywell are also beginning to invest in these devices. Although it is not detailed in this paper, Google has many NISQ devices which are internally at. In general, each of these companies is investing in one of three types of quantum computing. The first is Adiabatic Quantum Annealing (D-Wave is the only such device in this area), the second is ion trap based quantum computing (IONQ), and the

third is the most popular because of its similarity to classical microchip fabrication -i.e. printed gate based quantum computing (similar fabrication process as D-Wave). Within that third category, there are a few outliers, for example some companies are investing in using silicon or diamond imperfections to create a 3D lattice for quantum computing. And others such as Microsoft are focusing on discovering a new material to create fault tolerant quantum computing, and in the meantime they are focusing on software development.

Many of these companies have also invested in the human interfacing with these devices, specifically the programming languages to be used. IBM created QASM, Rigetti created Quil, and Microsoft created Q # . However, only two of these many companies have allowed public access to some or all of their devices. These are IBM and Rigetti. For this research, only those cloud accessible NISQ devices will be used, as well as classical simulators. The devices these companies created both fall into the third category - chip manufactured gate based quantum computing.

1.2 Technical Details

This section is purely for background knowledge of the hardware and software used in this research, and is meant only for the reader who might not have a full background on the topic.

All modern quantum computers are inherently designed to be cloud accessible, because the interface is too inaccessible to most people, and especially the general public, because of the extreme environmental condition requirements for a quantum device to operate. Namely, a very large dilution refrigerator to cool the actual physical chip down to about 15 millikelvin, and a near perfect vacuum inside the device. All of this is an effort to reduce noise and interactions from the atmosphere or thermal vibrations.

1.2.1 Software

At present, two companies (IBM and Rigetti) have publicly released software kits for quantum algorithm development as well as a system for distributing API keys for running quantum algorithms on NISQ devices constructed by these companies. Due to the nature of the error rates on NISQ devices, both IBM and Rigetti have also provided simulators, including an option for remote HPC simulators, which can run up to roughly 32 qubits. Additionally, these software kits offer the ability to apply noise models to these quantum computing simulators, in order to simulate the error rates which will occur on a NISQ device. Both of these companies have also released low level programming languages which are designed specifically for constructing and running quantum programs on simulators as well as real world quantum devices. These two programming languages are QASM [11] and Quil [20]. The Github repositories [4], [2] are the largely Python

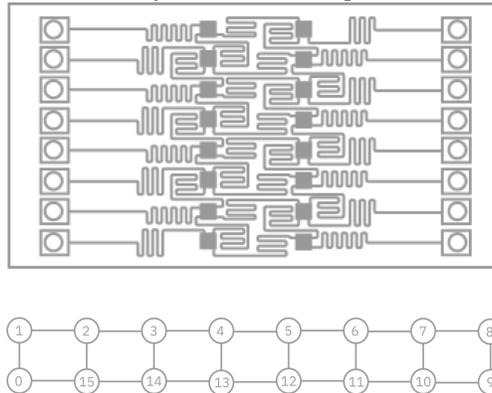
based scripts which provide a basis for quantum algorithm development. Their are algorithm development kits created by Rigetti and the IBM Quantum Experience, these are called Qiskit Aqua [3] and Grove, respectively.

Specifically for this research, personal API keys will be used on available quantum devices, IBMQX4, IBM Q 16 Melbourne, and Agave. The programming architecture for running quantum algorithms is to a large extent already implemented on these platforms. This research focuses on software which implements more and new problems on these devices using QAOA. Right now, there are examples and scripts in the respective Github repositories of some of the NP problems listed, such as maximum cut. However, this project focuses on first converting into an Ising model, which then can be converted into a Pauli operator, and then a quantum program in either QASM or Quil.

1.2.2 Hardware

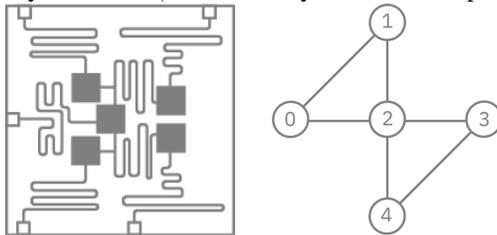
The IBM quantum experience offers a total of 4 NISQ devices for use over the cloud, 2 of which are available to anyone in the public. This research will only use those 2 which are publicly available. The names of these 2 devices are IBM Q 16 Melbourne, and IBMQX4 Tenerife. The numbers in the name represent the number of qubits. The connectivity of the qubits in these devices is not complete - full coupler maps are shown in Figure 1.2.2 and Figure 1.2.2.

Figure 1: IBM Q 16 Melbourne qubit connectivity



Every 24 hours each device is re calibrated, and for every experiment run, that calibration data is stored along with results of the experiment. the calibration data includes measured T1 and T2 times, as well as gate and readout error rates. T1 and T2 times refer to two different, but related, relaxation times of qubits. Essentially, these describe how the gate fidelity decreases over time. The IBM devices are Niobium and Aluminum Oxide (for superconductive Josephson junctions) circuits printed on silicon wafers. They are

Figure 2: IBM Q 5 Tenerife, and IBM Q 5 Yorktown qubit connectivity



fixed frequency qubits, and they have microwave oscillators which allow the readout of the qubits to occur.

Rigetti computing offers 3 NISQ devices accessible over the cloud, both of which are available to the public if they wish to do perform some research on the devices for a set period of time. the first is Agave, which has 8 qubits, and the second is called Acorn, which has 19 qubits (19 logical qubits, 20 physical qubits). The latest is called Aspen-1, and has 16 qubits. The physical qubits are joined via capacitive couplings. Both of these devices have an alternating set frequency to tunable qubit arrangement, although their connectivity differs greatly. This research does not use these devices at present however, because access is still somewhat limited.

The IBM quantum experience offers an HPC simulator for up to about 32 qubits. Both Rigetti and IBM offer basic local simulators, which can simulate up to about 28 qubits fine on a local machine, depending on that machines specifications.

1.3 QAOA

Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm which is iterative and relies on another quantum algorithm known as variational quantum eigensolver (VQE), the driver and cost hamiltonians, which are problem specific, and classical optimizer algorithms. The full details and proof for the algorithms functionality is given in [13]. QAOA is especially appealing because it runs in polynomial time.

This research focuses on using QAOA to solve NP-Hard problems. 21 of the most famous NP-Complete problems are described in [18]. These NP problems can be formulated as Ising (electronic structure) or QUBO (Quadratic Unconstrained Binary Optimization) models, and these formulations can be translated into driver and cost hamiltonians represented as Pauli operators, which means that these problems can be at least approximately solved by QAOA. In this case, the NP problems will be converted into Ising or QUBO models, the standard hamiltonian formulation of a problem is shown in equation 1. The difference between QUBO and Ising formulations in terms of variables output is that Ising variables will have an output value

of either -1 or 1, and QUBO variables will have an output value of either 0 or 1. The primary tunable parameter for QAOA, besides the constructed hamiltonians, is the number of times that the cost and driver hamiltonian is applied sequentially. In a generally consistent trend, the accuracy of the output increases as the number of steps increases.

1.4 D-Wave

Although not the focus of this research, the D-Wave devices are a specific type of quantum computer, called an adiabatic quantum annealer, which solves only optimization problems.

These devices are quantum annealing devices and are constructed and sold by D-wave [1], [9], [19]. In relation to optimization problems applied to D-wave systems, there is a massive amount of research in this area [7], [5], [14], [21], [12], [8]. Later on in this article, we briefly mention a comparison between a D-Wave 2000Q device and QAOA. In general however, for smaller problems, D-Wave is substantially better than such approximate algorithms, at least right now.

1.5 Background and Related Research

Studying hybrid quantum-classical algorithm is an increasing subject of interest. For example, [22] went into an in depth analysis of applying QAOA to the NP-Hard problem of Maximum Cut, specifically using the algorithm as an exact solver by applying it until the optimum was reached. Additionally, [15] and [10] are similar studies of QAOA, specifically looking into the future usage of QAOA on newer hardware topologies. Such research into the future applications and required performance measurements for hybrid algorithms is one of most viable means to demonstrate some quantum advantage.

2 Problem

There are many non-deterministic polynomial problems which have many important applications in increasing efficiency in industries ranging from pharmaceuticals, bioinformatics, manufacturing, traffic analysis, and massive data analysis relating to social media on the internet. The problem is that algorithms which find exact solutions to these NP (nondeterministic polynomial time) problems are not efficient, i.e. they do not run in polynomial time - instead they typically run in exponential time in the worst case.

From [16] we know that all of these problems are reducible in a specific manner, i.e. most NP-Complete problems can be reduced into different and simpler NP-Complete problems. This research focuses specifically

on the QAOA algorithm, as it seems to have the most potential to have immediate real world impacts as NISQ devices are continuing to be improved and expanded. As NISQ devices continue to improve, quantum optimization algorithms such as QAOA have the ability to begin having real world impacts faster than other quantum algorithms such as Shors algorithm, which requires fault tolerance, large number of qubits, high circuit depth, and high qubit connectivity.

2.1 NP-Hard Problems Tested

All NP problems in this research can be represented in terms of the mathematical objects called graphs. Graphs are mathematical objects defined by a collection of vertices, which are arbitrarily connected by a series of edges. These nodes and edges can have a numerical value assigned to them, these values are called weights. For the purpose of this research, only weighted versions will be considered, because it is trivial to use the weighted version as an unweighted version - i.e. the weight simply being 1.

2.1.1 Maximum Clique

Let G be a simple undirected graph defined as $G = (V, E)$, where each edge is not weighted. A clique is defined as a complete subgraph of G (i.e. every node in that subgraph is connected to every other node in that subgraph). A maximum clique is therefore the largest clique in G .

A weighted version of both maximum clique is identical to the versions described above, except that each vertex has a weight associated with it, and the value that is maximized is the sum of the weights of the vertices in the respective subsets generated (i.e. either an independent set or a clique), instead of just the number of vertices.

The decision version (NP-Complete Version) of the Maximum Clique problem asks the question of whether there is a clique of size N in a given graph. If the NP-Complete version is solved, the NP-Hard problem can be solved as well. The decision version of Maximum Clique is known to be NP-Complete, and the more general case described above (including the weighted version) are also known to be NP-Hard. Classically, heuristics have been developed to solve these problems more efficiently than an enumeration approach. However, none of these run in polynomial time. There are however approximate algorithms which run very fast, such as the minimal independent set approximation heuristic provided by the Python module Networkx.

2.1.2 Maximum Cut

Let G be a simple undirected graph defined as $G = (V, E)$ where each edge has a weight associated with it. A cut, C , is any nontrivial subset of V , and the cut weight is the sum of the weights of all of the edges crossing the cut. A maximum cut is therefore a cut of G with maximum weight. Determining the maximum cut of a graph is NP-Hard [18]. The decision version of maximum cut is answering the question of does a cut of weight N exist for a graph. This decision version is known to be NP-Complete.

3 Experimental Procedures

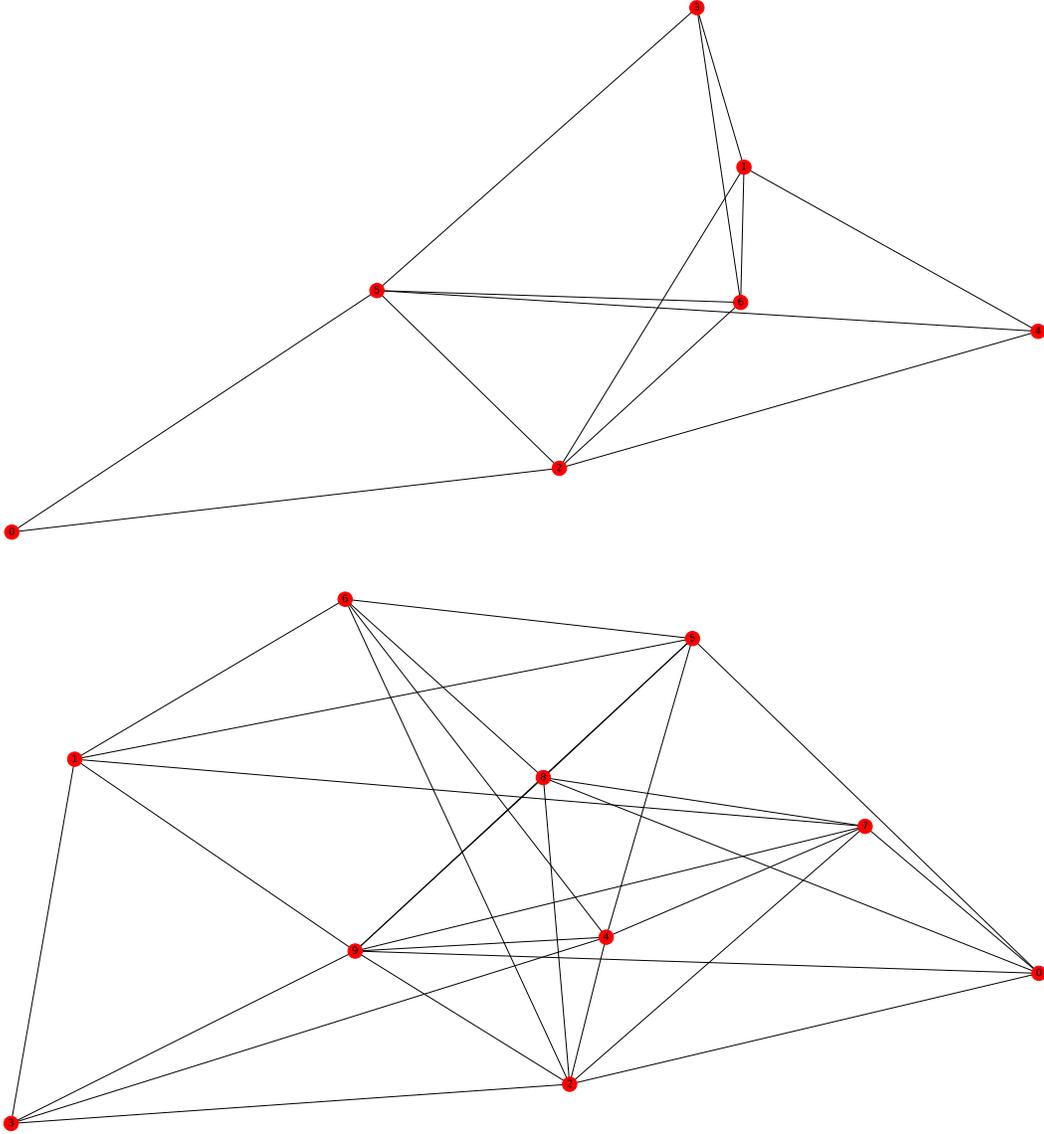
Table 3 shows the QUBO formulations for the two NP-Hard problems which will be solved by QAOA, each of these formulations generally take the form of equation 1.

$$H = \sum_{i \in V} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j \tag{1}$$

| NP-Hard Problem Hamiltonian Formulations | |
|--|--|
| NP-Hard problem | Hamiltonian formulation |
| Maximum Clique Nodes | $-A \sum_{i=1} x_i - B \sum_{i,j \in \bar{E}} x_i x_j \tag{2}$ |
| Maximum Cut | $\sum_{i,j \in E} (x_i \bar{x}_j + x_j \bar{x}_i) \tag{3}$ |

Because of the limited size of problems that can be run on current operating NISQ devices, as well as on current available simulators, only small problems are tested. The problem instances (which will be in the form of graphs) will be implemented on the two local simulators available from both Rigetti and IBM (including a noise model test), the real NISQ devices available from IBM. However, only averaged the qasm simulator will be used for large sample sizes involving averaging results. The purpose of these tests is to analyze the viability of NISQ devices - the NISQ device tests can be compared to the simulated noise models and the ideal simulators. Additionally, it is helpful to be able to differentiate between errors due to noise and the inherent nature of QAOA - where in some cases it only finds optimum solutions.

Figure 3: Gnp fixed test graphs

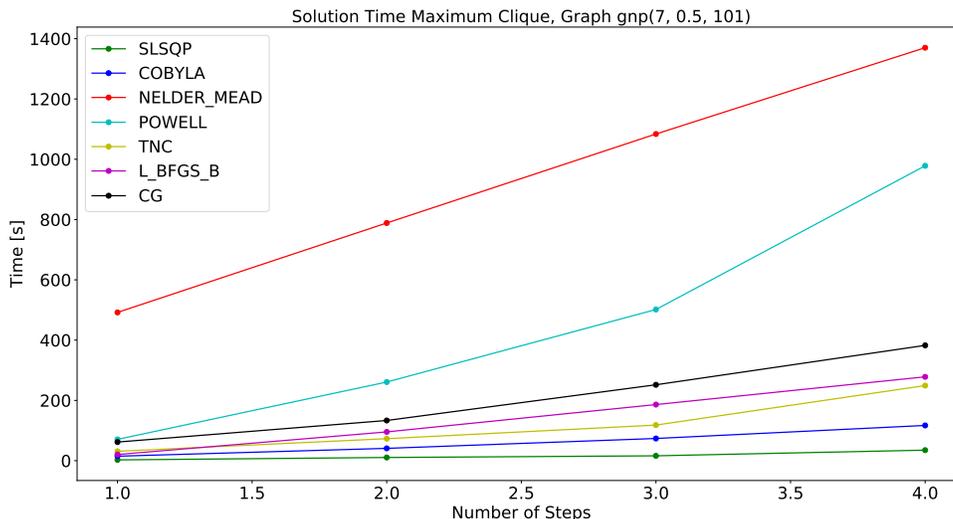


$$\textit{Approximation - Ratio} = \frac{\textit{Average - Experimental - Solution - Size}}{\textit{Optimum - Solution - Size}} \quad (4)$$

The accuracy of the results from running the QAOA algorithm will be determined using the approximation ratio metric, defined in equation 4.

The problem graphs tested are shown in figure 3, and were generated using the Python library Networkx, and fixing the random seed as 101 for consistent graph usage.

Figure 4: Maximum Clique Optimizer Run Time, averaged over 20 runs



4 Results

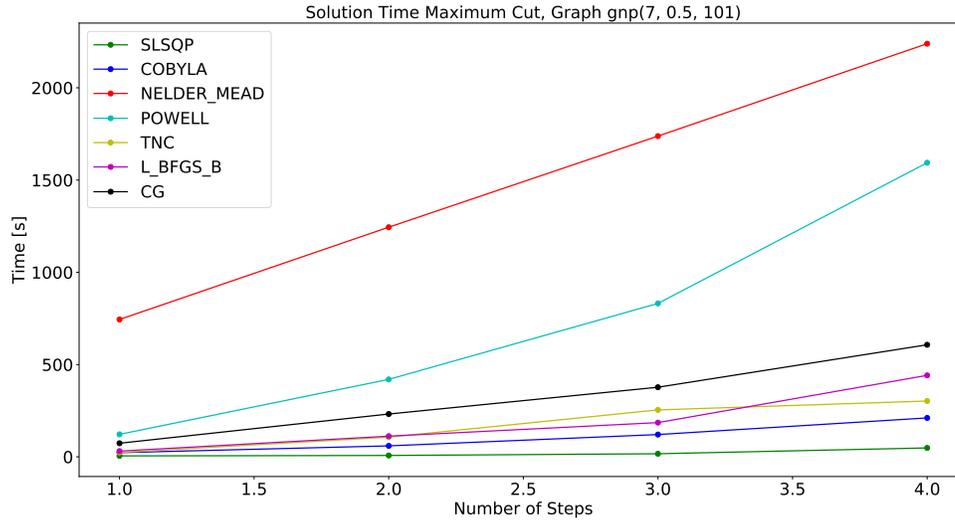
All of the code and code documentation can be found at the authors github repository for this project; https://github.com/epelofske/quantum_optimization. Further results are also stored in their raw form as well as in various graphs in the the project repository. The results in the repository include various unaveraged runs from using the Rigetti Quantum Virtual Machine (QVM), as well as results from a D-Wave 2000Q, accessed through the D-Wave Leap program. However, these results are not included in this paper because they are unaveraged, and the D-Wave results do not show anything novel at this comparatively small problem size. The results presented in this paper are averaged over at least 20 repetitions, and use the *C++* QASM simulator as the classical backend, with the exception of the real world NISQ computers.

Figures 4 and 4 show the run time when using QAOA for several classical optimizers. The best optimizer from that data, SLSQP, was then used for the rest of the accuracy tests because computation time is an important metric for determining the viability of using QAOA. It should be noted that the classical simulation time is included in these graphs, but this does not change the relative measures of run time when comparing these classical optimizers.

Figures 4, 4, 4, 4 show the approximation ratio from both random guesses and using QAOA. The random approximation ratio was averaged over 10000 runs, and is a constant for each graph. The QAOA data was averaged over 100 runs. Once again, figure 3 shows the exact graph which were used for these experiments.

Figures 4 and 4 are unaveraged runs for the problem of maximum cut. These graphs are unique for this

Figure 5: Maximum Cut Optimizer Run Time, averaged over 20 runs



research because this is data from the two physical NISQ devices available from IBM. The problem which was being applied to each of these was simply the undirected version of the architecture defined in 1.2.2 and 1.2.2.

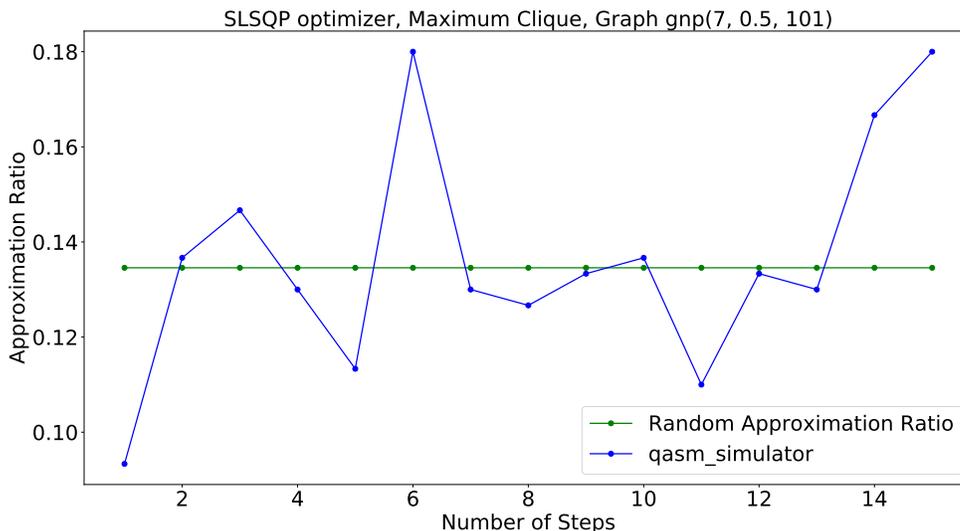
Finally, if the reader is interested in more specifics of the operation of the QAOA algorithm, the Github link provided includes some example quantum machine assembly language in the form of both Quil and QASM. Additionally, all back end software used in this research is open source and linked in the provided Github link.

5 Analysis

There are several important things which are observed in Figures 4 and 4. The first is that when considering the viability of an algorithm, run time is an important consideration, and in this case we find that SLSQP is the most efficient optimization algorithm. The other thing to note is that we can clearly see the polynomial complexity behavior of the algorithm, particularly because at these small problem sizes, the classical run time of simulating quantum systems is negligible.

Figures 4 and 4 show that overall we need faster communication time to the NISQ computers for better averaging, and it is important that we implement simulation noise models of real world NISQ devices to determine a more concise viability threshold. These figures as well as the approximation ratio figures show that overall accuracy, as well as run time, can be impacted by the type and complexity of the QUBO for the

Figure 6: Maximum Clique



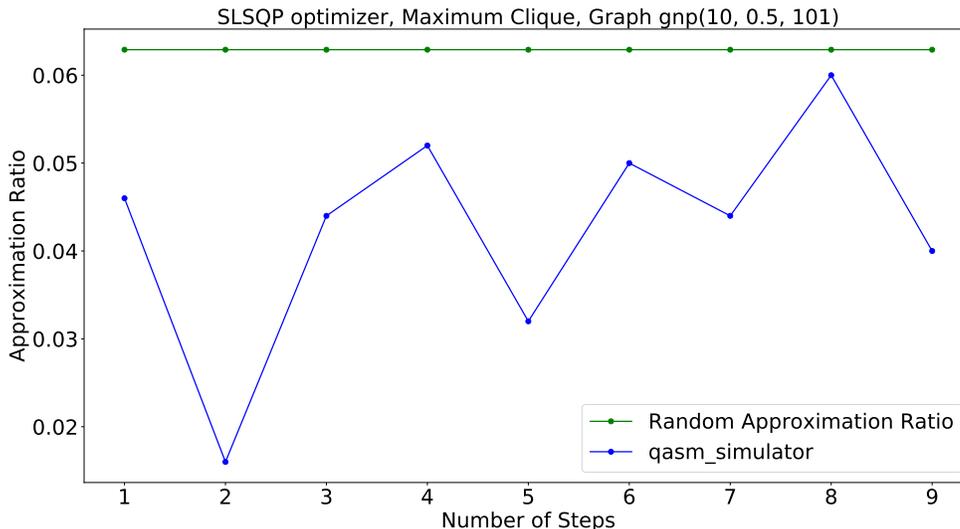
particular problem we are considering.

6 Conclusion

From the analysis of the results, we conclude the following regarding using QAOA to extend the capacity to solve such optimization problem such as NP-Hard problems.

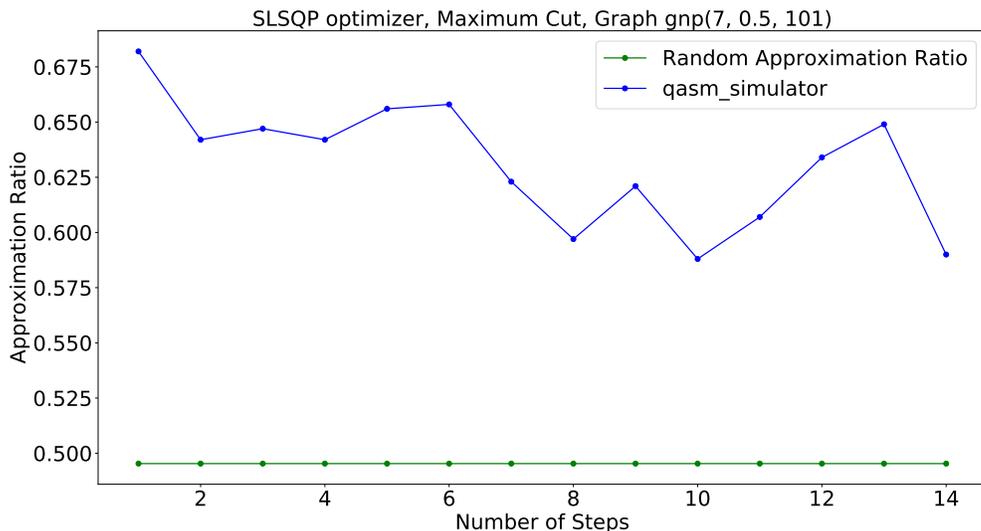
1. The network connectivity time is a significant current limitation - speeding up this communication time, such as the user specific Quantum Virtual Machine from Rigetti is a desirable property for such NISQ devices and the associated applications.
2. The complexity of the QUBO models for these optimization problems heavily affect the run time as well as accuracy for found solution sizes for the QAOA algorithm.
3. Currently, the D-Wave quantum annealers perform much better in comparison to QAOA, particularly for small problems. Further research using larger simulations of NISQ devices and, as the manufacturing improves, larger physical NISQ devices.
4. The performance of the classical optimization algorithm used in the QAOA algorithm is critical for the overall performance of the algorithm. Of all of the algorithms tested, the algorithm SLSQP (Sequential Least Squares Programming) was found to be the most efficient.

Figure 7: Maximum Clique



5. The trotterization order of the QAOA algorithm is critical for the overall performance of the algorithm. As a general trend, accuracy increases as the number of steps applied increases. However, a high number of steps is required for some types of optimization problems. For example, maximum cut is the easiest to demonstrate an improvement over random guessing, even at short circuit depth. However, for more constrained problems, such as maximum clique, a larger circuit depth is required for increased accuracy.
6. The QUBO or Ising formulation of the problem is critical when considering what NISQ device to use, and could be a very specific avenue to demonstrate quantum advantage in the future. For example, the Maximum Clique QUBO formulation tells us that the only quadratic interacting terms come from \overline{G} (the complement of the problem graph). This means that an especially dense problem graph could be solved on very sparse NISQ architecture. This is important because in general it seems that sparse device connectivity is the most viable device construction for the near term, and higher problem graph density is more computationally intensive for classical solvers.
7. In order to make meaningful use of QAOA, not only do we need larger NISQ computers, we also need significantly reduced error rates. Even for a comparatively short circuit depth algorithm, such error rates are a problem. Consequently, even for other quantum algorithms, these hardware improvements are necessary. Even so, this research and several others have demonstrated the hypothetical capabilities of QAOA.

Figure 8: Maximum Cut

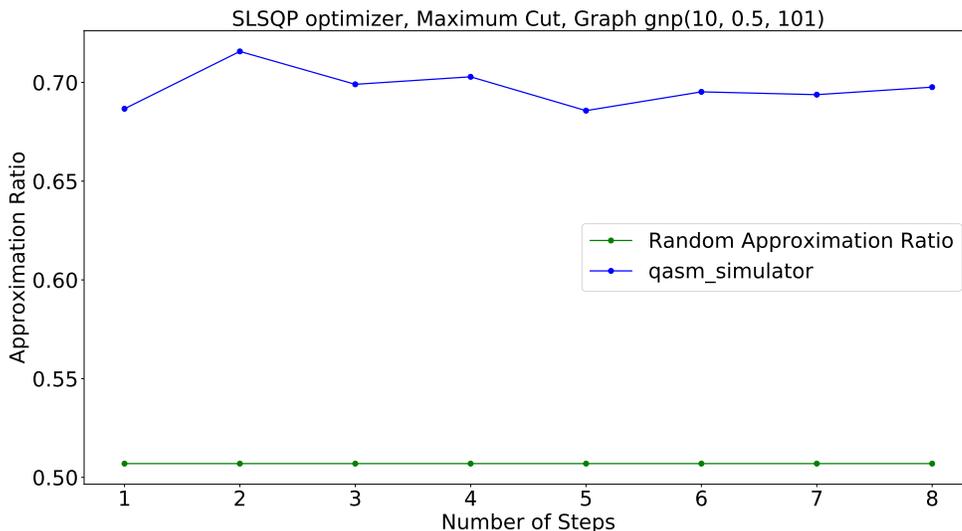


8. The IBM and Rigetti software packages include converters for many NP-Hard problems, and any optimization problem which can be represented in the particular Hamiltonian formulation of QUBO or Ising, and therefore QAOA can be implemented on physical NISQ computers using these software platforms.

For future research directions, we propose the following specific areas:

1. Testing more NP-Hard problems, such as TSP, Minimum Vertex Cover, Hamiltonian Cycle, Chromatic Number, and Maximum Independent Set.
2. Potentially paralyzing and increasing the sample size for the averaging process as well as increasing the problem sizes.
3. Applying QAOA on larger quantum computers, including NISQ computers released through IONQ, Rigetti, and future IBM devices.
4. Implementing noise models for various real world NISQ devices, and simulating higher qubit connectivity on classical simulators, thus providing an insight into the performance of higher connectivity NISQ architectures.

Figure 9: Maximum Cut



7 Acknowledgements

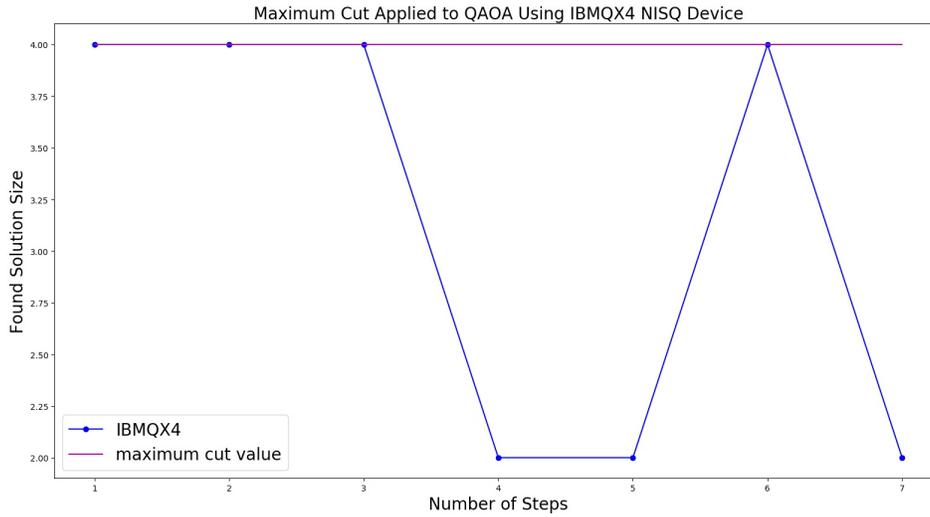
We acknowledge use of the IBM Q for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team.

We also thank the Slack communities of Rigetti-Forest and Qiskit, and the D-Wave Leap program.

References

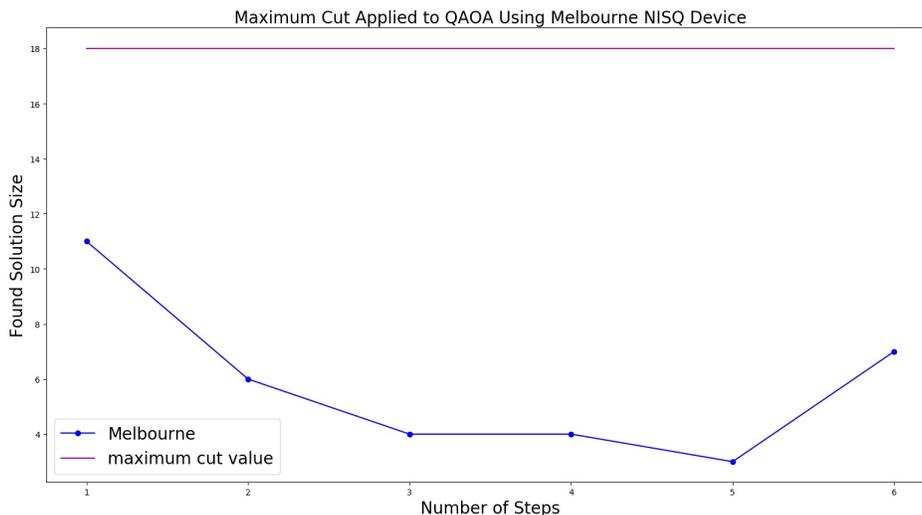
- [1] Quantum Computing for the Real World Today. Technical report.
- [2] *Pyquil*, 2018 October. A Python library for quantum programming using Quil.
- [3] *Qiskit Aqua*, 2018 October. Aqua provides a library and tools to build applications for Noisy Intermediate-Scale Quantum (NISQ) computers.
- [4] *Qiskit Terra*, 2018 October. Terra provides the foundations for Qiskit. It allows the user to write quantum circuits easily, and takes care of the constraints of real hardware.
- [5] Evgeny Andriyash, Zhengbing Bian, Fabian Chudak, Marshall Drew-Brook, Andrew D King, William G Macready, and Aidan Roy. Boosting integer factoring performance via quantum annealing oosets TECHNICAL REPORT. Technical report.

Figure 10: IBMQX4 NISQ device



- [6] Eva Borbely. Grover search algorithm. Technical report.
- [7] Guillaume Chapuis, Georg Hahn, and Guillaume Rizk. Finding Maximum Cliques on the D-Wave Quantum Annealer. Technical report.
- [8] Carleton Coffrin, Harsha Nagarajan, and Russell Bent. Ising Processing Units: Potential and Challenges for Discrete Optimization. Technical report.
- [9] Steve Conway, Earl C ; Joseph, and Robert Sorensen. The Promise of Quantum Computing Strengths and Limitations of Classic Supercomputers. Technical report, 2016.
- [10] Gavin E Crooks. Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem. pages 15–17, 2018.
- [11] Andrew W. Cross, Lev S. Bishop, John A. Smolin, and Jay M. Gambetta. Open Quantum Assembly Language. 7 2017.
- [12] Hristo Djidjev, Guillaume Chapuis, Georg Hahn, and Guillaume Rizk. Efficient Combinatorial Optimization Using Quantum Annealing. Technical report, 2016.
- [13] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. 11 2014.

Figure 11: Melbourne NISQ Device



- [14] Phil Goddard, Susan Mniszewski, Florian Neukart, Scott Pakin, and Steve Reinhardt. How Will Early Quantum Computing Benefit Computational Methods? Technical report.
- [15] G G Guerreschi and A Y Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. pages 1–14, 2018.
- [16] Richard M. Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. 2010.
- [17] Samuel J Lomonaco. A LECTURE ON SHOR’S QUANTUM FACTORING ALGORITHM VERSION 1.1. Technical report, 2000.
- [18] Andrew Lucas. Ising formulations of many NP problems. 2 2013.
- [19] Technology Overview. The D-Wave 2000Q™ Quantum Computer. Technical report.
- [20] Robert S Smith, Michael J Curtis, and William J Zeng. A Practical Quantum Instruction Set Architecture. Technical report.
- [21] Hayato Ushijima-Mwesigwa, Christian F A Negre, and Susan M Mniszewski. Graph Partitioning using Quantum Annealing on the D-Wave System. Technical report.
- [22] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A*, 2018.

What is the Radius of the Earth's Core?

New Mexico

Supercomputing Challenge

Final Report

April 3, 2019

Team 32

Los Alamos Middle School

Team Member

Veronica Sofia Parra

Sponsoring Teacher

Curtis B. Terrill

Table of Contents

| | |
|---|----|
| Executive Summary | 4 |
| 1.0 Introduction | 5 |
| 1.1 Background | 5 |
| 1.2 Scientific Accepted Outer Core Radius | 8 |
| 2.0 Methodology | 9 |
| 2.1 Seismic Data Analysis | 10 |
| 2.2 Seismic Data Criteria | 10 |
| 2.3 Distance Between Two Points Approximation | 12 |
| 2.4 Earth Mantle Approximation | 12 |
| 2.5 Verification and Validation Analysis | 14 |
| 3.0 Results | 14 |
| 3.1 Future Development | 15 |
| 4.0 Conclusions | 16 |
| 5.0 Significant Achievement | 17 |
| 6.0 Acknowledgements | 17 |
| 7.0 References | 17 |
| 8.0 Appendix | 19 |
| Appendix A, Seismic Station Event Information | 19 |

List of Figures

| | |
|--|----|
| Figure 1. Seismogram for M6-116 km SE of Lata, Solomon Islands Earthquake | 6 |
| Figure 2. View of the Secondary-Wave Shadowing Zone | 7 |
| Figure 3. View Showing the Triangle for Calculation of the Outer Core Radius | 8 |
| Figure 4. Correction for Refraction of the Secondary-Wave | 13 |
| Figure 5. Calculated Average Radius for the Earth's Outer Core | 15 |

List of Tables

| | |
|---|----|
| Table 1. Earthquake Event Information | 11 |
| Table A. Seismic Stations Meeting the Data Criteria | 19 |

Executive Summary

The Earth's magnetic field protects the planet from the Sun's harmful solar radiation wind and is created by the rotation of the Earth's metallic core. The magnetic field depends on the size and rotational speed of the Earth's core. A computer code using the Python programming language has been developed called "Earth Outer Core Radius" that analyzes the seismic data obtained from the United States Geological Survey to calculate the radius of the Earth's outer core. The computer code "Earth Outer Core Radius" calculated an average radius of the Earth's outer core of 3580 km. The scientific accepted radius of the Earth's outer core is 3485 ± 3 km. The "Earth Outer Core Radius" program over predicts the outer core radius by 2.7% due to the primary assumption that the Earth's mantle is made up of a single soil layer having a uniform composition and density.

1.0 Introduction

The planet Earth has a core made up of a solid inner core surrounded by a molten-liquid outer core. These two cores combined make up the Earth's core and produces the magnetic field. The Earth's magnetic field surrounds the planet to protect every life form from the Sun's strong solar wind, a stream of charged particles.

The inner core of the Earth is a solid iron core. The outer core is a molten-liquid layer surrounding the solid inner core. It is composed of mostly iron and nickel. It is this molten-liquid core that allows the solid inner core to rotate faster than the rotation of the planet Earth and to produce the magnetic field. The magnetic field depends on the size and rotational speed of the Earth's core. The seismic data from earthquakes is collected by the United States Geological Survey (USGS) and the seismic data is utilized to calculate the radius of the Earth's outer core.

Earthquakes are major geologic events that occur suddenly. They most often occur near the boundaries between tectonic plates (fault zones) because the plates push against each other. When one tectonic plate moves on top of another plate or rubs with another plate, it releases a tremendous amount of energy that produces multiple seismic waves that travel through the Earth. The speed of the seismic waves depends on the composition and density of the soil. By analyzing the detected seismic waves, the Earth's interior composition can be determined. A computer code named "Earth Outer Core Radius" was developed using the Python programming language to analyze the seismic data from the USGS in order to calculate the radius of the Earth's outer core.

1.1 Background

Seismic stations detect and record the earthquakes on seismograms. The USGS collects the seismic (seismograms) data from 486 seismic stations located throughout the United States through its Earthquake Hazards Program [1]. There are two types of seismic waves produced from an earthquake and recorded on seismograms: Body waves and Surface waves.

Most seismic stations only record the body waves because body waves can travel through the interior of the Earth and can be detected far away. Surface waves (i.e., Rayleigh-waves and Love waves) are only recorded by seismic stations located very close to the earthquake because surface waves travel near the surface of the Earth and dissipate rapidly.

Furthermore, there are two types of body waves called Primary-waves (P-waves) and Secondary-waves (S-waves). The seismogram measured by the State Center, IA Seismic Station (US SCIA) for the earthquake that occurred at Lata, Solomon Islands on July 17, 2018 is presented in Figure 1. The seismogram shows the P-wave and the S-wave for the earthquake [2]. The P-waves are compression waves that speed and slow down depending on the composition and density of the soil they are traveling through. The P-waves can travel through the Earth's inner and outer core. Because the P-waves are the first to arrive at the seismic station, they are called Primary-waves (P-waves). The S-waves are shear waves that travel through the mantle of the Earth with more destructive force than the P-waves. The S-waves do not travel through the Earth's core because they get attenuated by the molten-liquid outer core. Because the S-waves arrive later (after the P-waves), they are called Secondary-waves [3].

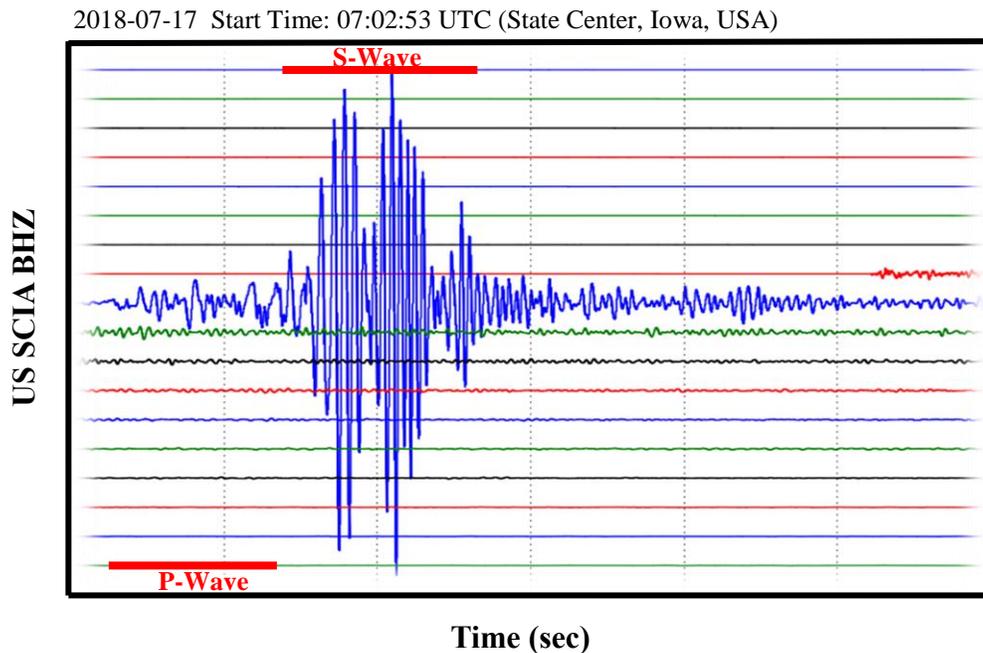


Figure 1. Seismogram for M6-116 km SE of Lata, Solomon Islands Earthquake.

As a result of the S-waves not being able to travel through the Earth's outer core, the outer core shields an area on the opposite side of the earthquake from receiving S-waves and preventing seismic stations from detecting S-waves directly. This shielded area represented is called the S-wave shadowing zone, see Figure 2.

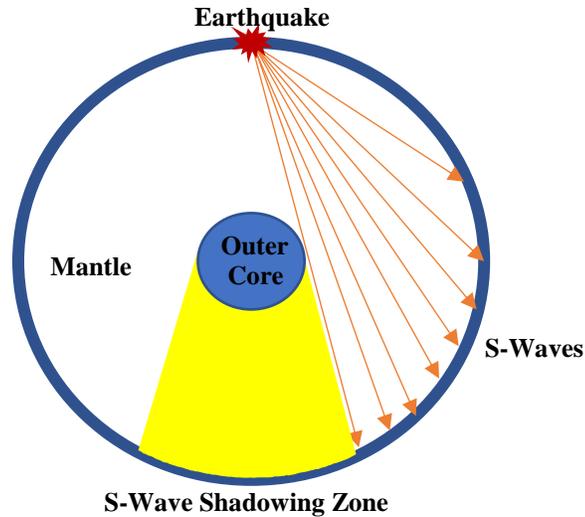


Figure 2. View of the Secondary-Wave Shadowing Zone.

By measuring the angle (as measured from the center of the Earth) or the distance from the earthquake's epicenter and the beginning of the S-wave shadowing zone, the radius of the Earth's outer core can be calculated, see Figure 3. The radius of the Earth's outer core can be calculated by forming an isosceles triangle from three points: Earth's core center, earthquake's epicenter, and the seismic station's location. If the isosceles triangle is cut in half to form a right triangle, the radius of the Earth's outer core is the adjacent side with the obtuse side equal to the Earth's radius and the opposite side equal to half the distance from the earthquake's epicenter to the seismic station's location [4].

The S-waves do not travel in a straight line because the earth is composed of varying concentric soil layers having different compositions and densities. The velocity of the S-waves in the Earth's mantle increases with the depth. Furthermore, the refraction of the S-waves causes the path to follow an upward curvature (bending) direction due to passing through varying soil layers. The velocity and curvature of the S-waves obey Snell's Law [3]. Snell's Law describes the relationship between the velocities and incidence/reflection angles of the S-wave as it travels through a boundary between two different types of soil layers (varying compositions and densities). Snell's Law is utilized to correct the calculated radius of the Earth's outer core due to the curvature (bending) path of the S-wave as it travels from the earthquake's epicenter to the seismic stations.

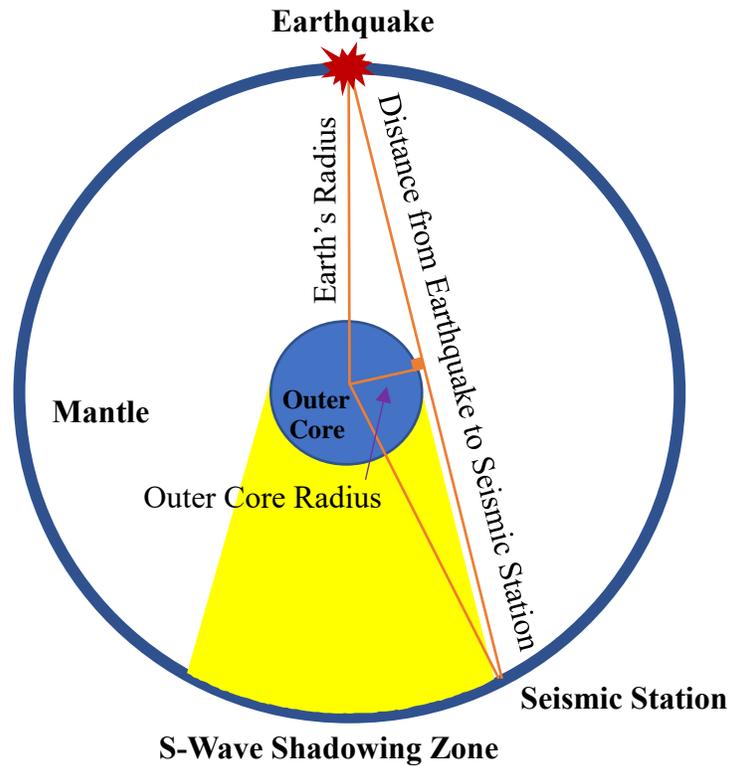


Figure 3. View Showing the Triangle for Calculation of the Outer Core Radius.

1.2 Scientific Accepted Outer Core Radius

The first recording of seismic waves from a remote earthquake occurred in 1889. This scientific breakthrough stimulated the deployment of worldwide observatories and the International Organization of Seismology. The identification of the P-waves and S-waves in seismological records in 1906 allowed the detection of the Earth's core. Because the S-waves do not pass through liquids and the diffraction of the P-waves, the Earth's core was deduced to be made up of two different parts: A solid inner core spinning within an outer liquid core. Seismologists have accurately measured the radius of the Earth's inner and outer core from measuring the seismic waves produced from natural earthquakes and nuclear explosions [5].

Three different scientific methods have been utilized to measure the radius of the Earth's outer core. The first and most accurate scientific method is to determine the distance at which the P-wave is diffracted as it contacts the outer core and use it to calculate the outer core radius. The Earth's outer core shields an area on the opposite side of the earthquake from receiving P-waves and preventing seismic stations from detecting the P-waves directly. This shielded area is called

the P-wave shadowing zone and it is similar to the S-wave shadowing zone. Nuclear explosion data, where the location and the origin time are exactly known, were used to calculate the Earth's outer core using this methodology [6, 7]. The second scientific method is to determine the outer core radius from inversion of the normal mode data obtained from analyzing the seismic data. The free oscillation period of the Earth can be calculated from analyzing the seismic data utilized to measure the density of the varying Earth's soil layers. The Earth's outer core has a very high density. The location where the soil density dramatically increases is used to calculate the Earth's outer core radius [8]. The third scientific method is to measure the geomagnetic field from space (satellite geomagnetic data) because the magnetic field produced by magnetization in the Earth's mantle can be distinguished from the field produced by the electric current in the outer core. Based on the detected variation in the magnetic field, the radius of the Earth's outer core can be calculated [9]. The radius of the Earth's outer core has been measured to be 3485 ± 3 km [6].

2.0 Methodology

The Python programming language was utilized to develop a computer code called "Earth Outer Core Radius." The "Earth Outer Core Radius" program analyzes the seismic data collected by the USGS to calculate the radius of the Earth's outer core. The computer code's methodology utilized to analyze the seismic data and to calculate the outer core's radius is:

1. Analyze the seismic data collected by the USGS from all earthquakes that occurred in 2018 from around the world. Only the earthquakes having a Moment Magnitude (M_w) of 6 M_w or greater and occurring within 50 km from the Earth's surface (epicenter) are utilized.
2. Determine which earthquakes are measured by seismic stations located within the United States that showed an S-wave shadowing zone (no S-wave is found on the seismograms). The seismogram data from various seismic stations that are progressively more distant from the earthquake are reviewed to find two seismic stations located close to each other (within 500 km) where by one seismic station detected the S-wave and the other seismic station did not. The angle/distance for the S-wave shadowing zone is measured from the earthquake's epicenter to where the S-wave is no longer detected.
3. Calculate the initial radius of the Earth's outer core using the angle and distance information obtained from calculating the S-wave shadowing zone.

4. Correct the initial calculated outer core radius for the curvature (bending) of the S-wave through the Earth's mantle due to refraction by using Snell's Law and assuming that the density of the mantle is uniform (1-layer soil approximation).
5. Print the calculated average radius of the Earth's outer core, average angle from the earthquake's epicenter to the start of the S-wave shadowing zone, and the average distance from the earthquake's epicenter to the start of the S-wave shadowing zone.

The computer code, "Earth Outer Core Radius," was developed using Canopy Python version 2.7 programming language supported by Enthought, Inc [10].

2.1 Seismic Data Analysis

The USGS seismic data was obtained from the Incorporated Research Institutions for Seismology (IRIS). The USGS collects the seismic data from 486 independent and government funded seismic stations throughout the United States through its Earthquake Hazards Program and gives the seismic data to IRIS which makes the data available to the public via the Data Management System (DMS) [11]. The Earthquake Hazards Program monitors earthquake activity worldwide [1]. The DMS is funded by the National Science Foundation under Cooperative Agreement EAR-1261681 through the "Seismological Facilities for the Advancement of Geoscience and Earth Scope" project [12].

The seismic data received from DMS was preprocessed because only the first line of the seismic data collected by each seismic station for each earthquake was utilized by the computer code. The data was collected into a text file which is read by the computer code and consisted of:

Seismic Station Name, Geophysical Coordinate Location, Earthquake Name, Date of Earthquake, Time of the Earthquake, Magnitude, Depth, Geophysical Coordinate Location, Angle from Epicenter to Seismic Station, Distance from Epicenter to Seismic Station, S-Wave Recording.

The preprocessing of the seismic data was performed to reduce the amount of seismic data that was used by the computer code.

2.2 Seismic Data Criteria

To minimize errors from not knowing the exact location of the earthquake and the start of the S-wave shadowing zone, three criteria are imposed on the seismic data. The computer code checks

for these three criteria in the seismic data and will only utilize the data to calculate the radius of the Earth’s outer core if the data meets all three criteria. The three criteria are:

1. The earthquake must have a Moment Magnitude of 6 M_w or greater to make it easier to identify the S-wave on the seismogram.
2. The earthquake must be a “shallow” earthquake to eliminate the effects of the earth’s varying soil composition and density and the assumption that the earthquake occurs on the surface of the Earth. A shallow earthquake is an earthquake that occurred within 50 km from the Earth’s surface (epicenter). This requirement also maximizes the angle and distance from the earthquake’s epicenter to the start of the S-wave shadowing zone.
3. The two seismic stations (one measured the S-wave and the other did not) must be located within 500 km of each other to minimize the error in calculating the start of the S-wave shadowing zone.

There were 134 earthquakes around the world in 2018 that had a Moment Magnitude of 6 M_w or greater [1]. Only 10 of the earthquakes met the three criteria stated above. The 10 earthquakes are listed in Table 1.

Table 1. Earthquake Event Information.

| Earthquake | Date | Time (UTC) | Magnitude (M_w) | Depth (km) | Location | |
|---|------------|------------|---------------------|------------|----------|-----------|
| | | | | | Latitude | Longitude |
| South of the Fiji Islands | 4/2/2018 | 5:57:35 | 6.1 | 42 | -24.719 | -176.886 |
| Southeast of Lata, Solomon Islands | 7/17/2018 | 7:02:53 | 6.0 | 38 | -11.594 | 166.432 |
| West of Kandrian, Papua New Guinea | 7/19/2018 | 18:30:32 | 6.0 | 30 | -6.114 | 148.730 |
| Central Mid-Atlantic Ridge | 7/23/2018 | 10:35:59 | 6.0 | 10 | -0.299 | -19.252 |
| West-Northwest of Ile Hunter, New Caledonia | 9/10/2018 | 19:31:37 | 6.3 | 12 | -21.988 | 170.158 |
| Drake Passage | 10/29/2018 | 6:54:21 | 6.3 | 10 | -57.434 | -66.383 |
| South-Southeast of Pangai, Tonga | 11/10/2018 | 8:33:21 | 6.1 | 35 | -20.454 | -174.008 |
| East of Visokoi Island South Georgia and South Sandwich Islands | 11/15/2018 | 20:02:22 | 6.4 | 15 | -56.706 | -25.546 |
| Southeast of Pacific Rise | 11/15/2018 | 23:09:01 | 6.3 | 10 | -56.236 | -122.044 |
| Southeast of Easter Island | 12/19/2018 | 1:37:40 | 6.2 | 10 | -36.118 | -101.019 |

The seismic station information that recorded each earthquake listed in Table 1 and used to calculate the S-wave shadowing zone is presented in Appendix A, “*Seismic Station Event Information.*”

2.3 Distance Between Two Points Approximation

The distance between two seismic stations is calculated using the equation for a spherical earth projected to a plane [13]. As a result, the shortest distance between two geographical locations in a flat Earth approximation is a straight line and it is based on the Pythagorean theorem. The equation utilized in the computer code is:

$$D = R\sqrt{(\Delta\alpha)^2 + (\cos(\alpha_m)\Delta\beta)^2}$$

where

D = Distance between two geographical coordinates (Latitude, Longitude) in kilometers

R = Radius of the Earth (6378 km)

$\Delta\alpha$ = Difference in Latitude points ($\alpha_2 - \alpha_1$) in radians

α_m = Mean Latitude [$(\alpha_1 + \alpha_2)/2$] in radians

$\Delta\beta$ = Difference in Longitude points ($\beta_2 - \beta_1$) in radians

For short distances (less than 500 km) and geographical coordinates not located close to the poles, the spherical earth projected to a plane equation gives very good accuracy.

2.4 Earth Mantle Approximation

The S-waves do not travel in a straight line from the location of the earthquake to the seismic stations because the earth is composed of varying concentric soil layers having different compositions and densities. As the S-wave travels from one soil layer to another denser soil layer in the mantle of the Earth, the velocity increases and the direction of the wave curves upward (bending). Figure 4 shows the actual curvature of the S-wave as it travels from the epicenter of the earthquake to the seismic station. The velocity and the curvature direction of the S-wave through the mantle obeys Snell's Law [3]. Snell's Law describes the relationship between the velocities and incidence/reflection angles of the S-wave as it travels through a boundary between two different types of soil layers (varying compositions and densities). Each boundary layer can be represented by a set of multi-variable equations based on Snell's Law. These multi-variable equations can be solved simultaneously as a solution to the multi-variable array. This advance mathematics (e.g., linear algebra) is beyond the scope of this research project.

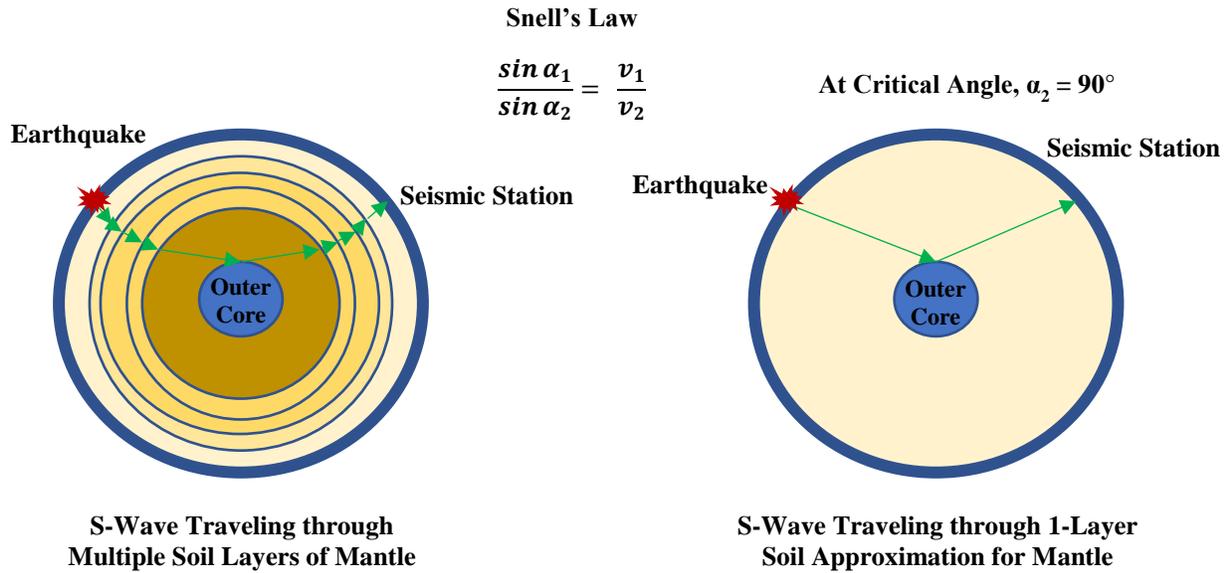


Figure 4. Correction for Refraction of the Secondary-Wave.

A simplification of the Earth's various soil boundary layers is utilized in the calculation. The Earth is assumed to be made up of a single soil layer composition having a constant density. Snell's Law is applied to this simplification to obtain a single equation that approximates the velocity and direction that the S-wave travels from the earthquake to the seismic station locations as it bounces off the outer core layer, see Figure 4.

$$\frac{\sin \alpha_1}{\sin \alpha_2} = \frac{v_1}{v_2}$$

The incidence of the S-wave onto the Earth's outer core causes the S-wave to travel perpendicular to the outer core surface before it reflects back toward the surface of the Earth. Therefore, at the critical angle (α_1) where the S-wave contacts the Earth's outer core, $\alpha_1 = 90^\circ$ and $\sin 90^\circ = 1$. Furthermore, the S-wave is traveling at a velocity (v_1) of 7.49 km/s when it makes contact with the Earth's outer core [5]. As the S-wave travels on the outer core surface, it does not significantly gain speed. Therefore, v_2 is approximately 7.50 km/s. The refraction angle (α_2) can be obtained from Snell's Law. The computer code calculated the refraction angle (α_2) to be 87.04° .

The refraction angle (α_2) is utilized to calculate the curvature of the S-wave as it travels from the earthquake's epicenter to the seismic station location. The calculated curvature distance (e.g., the

difference in distance between the maximum curvature of two points and the straight line of two points) is subtracted from the initial calculated radius of the Earth's outer core to obtain the final radius.

2.5 Verification and Validation Analysis

The results of the computer code were independently verified by developing an Excel spreadsheet. The same methodology used in the computer code was utilized in the Excel spreadsheet for only the 10 earthquakes presented in Table 1. The results from the computer code were the same as the results from the Excel spreadsheet. The calculated radius of the Earth's outer core was compared to the scientific accepted outer core radius. The computer code over predicted the outer core radius by 95 km (i.e., 2.7% from the scientific accepted outer core radius).

The location of the seismic stations meeting the third criteria "*two seismic stations must be located within 500 km of each other...*" was validated by using the Global Earthquake Explorer (GEE) program. The GEE program provides an interactive map-based program for viewing earthquakes and the seismograms recorded by seismic stations from around the world. The GEE program was developed by the Department of Geological Sciences of the University of South Carolina [2]. The earthquake data from the GEE program was also used to validate the earthquake data (i.e., location, magnitude, depth) from the DMS because the GEE program uses seismic data from seismic stations located in foreign countries and from the USGS. The earthquake data obtain by foreign seismic stations and used in the GEE program was the same as the earthquake data obtained from the DMS.

3.0 Results

The calculated radius of the Earth's outer core using the seismic data from the earthquakes meeting the above criteria is 3580 km, see Figure 5. The "Calculated Average Radius" line on the graph begins with the first earthquake in 2018 that meet the above criteria. The average calculated angle and distance for the S-wave shadowing zone from shallow earthquakes are 105.0 degrees and 11677 km. The scientific accepted radius of the Earth's core is 3485 ± 3 km [6]. The reported angle and distance for the S-wave shadowing zone for shallow earthquakes are approximately 105 degrees and 11680 km [5]. The computer code calculated outer core radius is off by 95 km (2.7%) due to the following assumptions utilized in the methodology:

- a. The earthquake’s epicenter is assumed to occur on the surface of the Earth.
- b. The average distance between the two seismic stations (one measuring the S-wave and the other not) is assumed to be the beginning of the S-wave shadowing zone and this average distance is used to calculate the angle and distance from the earthquake to the S-wave shadowing zone.
- c. The planet Earth’s radius is assumed to be constant (6378 km).
- d. The density of the Earth’s mantle is assumed to have a uniform soil density (1-layer soil approximation).

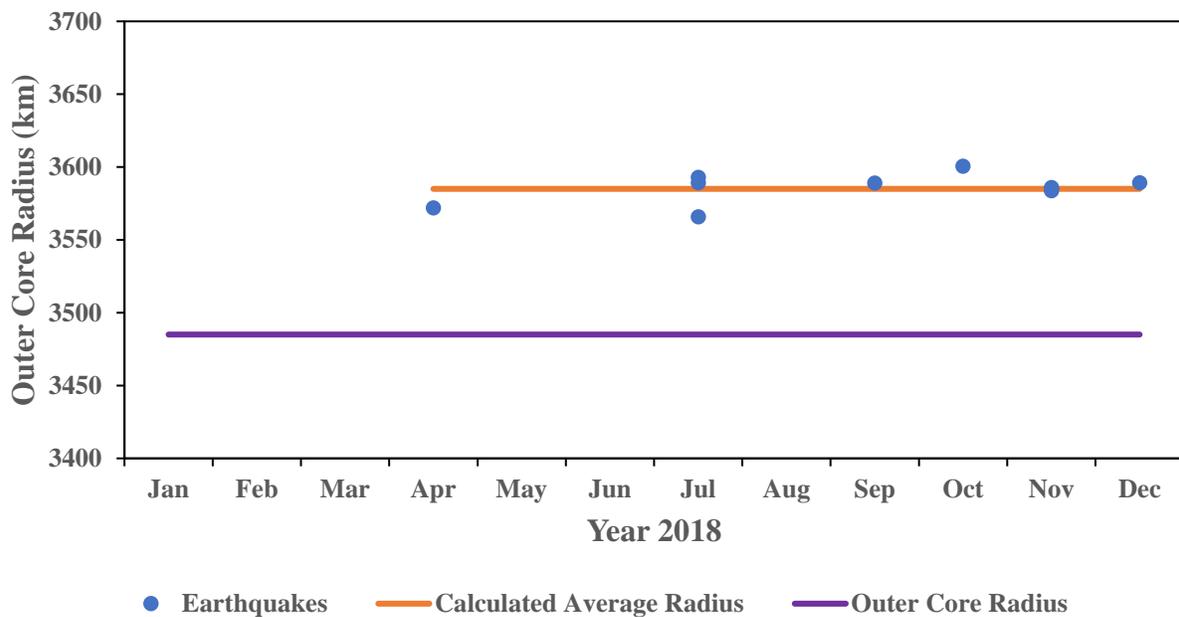


Figure 5. Calculated Average Radius for the Earth’s Outer Core.

3.1 Future Development

There are many improvements that can be made to the “Earth Outer Core Radius” program. The short-term improvements are incorporating the varying soil layers of the Earth’s mantle to reduce the calculational error. The mantle can be modeled as three distinct boundary soil layers (i.e., Crust, Upper Mantle, and Lower Mantle). This improvement will involve developing a set of multi-variable equations based on Snell’s Law to represent the various soil boundary layers in the mantle and simultaneously solving them. The computer code can also be improved by eliminating the assumption that all of the earthquakes occur on the surface of Earth (epicenter). This

improvement will involve using spherical trigonometry to correct for the depth of the earthquake at the epicenter location.

By eliminating these two assumptions, the computer code will more accurately calculate the radius of the Earth's outer core within 1% of its true value. Compared to the current scientific methodologies utilized to calculate the outer core radius, this simple methodology calculates the radius of the Earth's outer core very precisely.

The long-term improvement is to have the computer code calculate the P-wave shadowing zone to be able to calculate the radius of the Earth's inner core. The P-waves are refracted by the molten-liquid outer core and are not detected between 104 and 140 degrees from the earthquake's epicenter. By knowing the radius of the inner and outer cores, the volume of molten-liquid can be calculated to better predict the magnitude of the Earth's magnetic field.

4.0 Conclusion

A computer code named "Earth Outer Core Radius" was developed using the Python programming language to analyze the seismic data collected by the USGS in order to calculate the radius of the Earth's outer core. The computer code over predicts the radius of the Earth's outer core by 95 km because the computer code uses a simple 1-layer soil (uniform density) approximation for the Earth's mantle. The Earth's mantle is actually made up of multiple concentric soil layers of varying composition and density that refract/reflect and increase/decrease the velocity of the seismic waves as they pass through the boundary of the varying concentric soil layers owing to Snell's Law. Nevertheless, the difference in the calculated radius to the scientific accepted radius of the Earth's outer core is less than 2.7%.

The methodology of using the S-wave shadowing zone to calculate the radius of the Earth's outer core is simple to use and gives an acceptable result compared to the other scientific methods utilized to calculate the outer core radius. The size of the outer core is important because the molten-liquid iron/nickel layer surrounding the solid inner core allows the inner core to move independently of the planet Earth's rotation which produces an electric current in the molten-liquid outer core due to circulation and convection and thus, generates the Earth's magnetic field. This simpler methodology can be used to model the effects of the size of the Earth's core within programs developed to model the Earth's magnetic field. One such program widely used by

seismologists is the Preliminary Reference Earth Model (PREM) developed by the International Association of Geodesy [14]. Incorporating this simpler methodology into PREM will simplify the model without sacrificing accuracy.

5.0 Significant Achievement

This research project offers a simple solution to a complicated science problem. It shows that anyone with a curious mind and determination can perform scientific research and solve complicated science problems.

This research project was very difficult to perform because of the initial lack of mathematical and physics knowledge. I had to learn Python programming language, algebra, trigonometry, physics, and geology to perform this project. My significant achievement was completing this difficult research project. I am very proud of myself for this major accomplishment. I learned that I can accomplish anything if I set my mind to doing it.

6.0 Acknowledgement

I would like to acknowledge Curtis B. Terrill (my sponsoring teacher), Mark B. Petersen (my Python computer programming mentor), and Ruben Rangel (my geologist mentor). Terrill is a Science Teacher at Los Alamos Middle School, Petersen is a Climate Scientist at Los Alamos National Laboratory, and Rangel is a Geology Consultant for the United States Department of Energy - Los Alamos Field Office. I would also like to thank my parents, Santiago and Norma Parra, for encouraging me to complete this project and not letting me quit.

7.0 References

- [1] United States Geological Survey, <www.usgs.gov>.
- [2] Global Earthquake Explorer, <www.seis.sc.edu/gee>.
- [3] **International Handbook of Earthquake and Engineering Seismology**, Academic Press, 2002.
- [4] Science Buddies, *Measuring the Diameter of the Earth's Core with Seismic Waves around the Globe*, <www.sciencebuddies.org>.
- [5] **Treatise on Geophysics**, 2nd Ed, Elsevier B.V., Vol 1, 2015.

- [6] Engdahl, E.R. and Johnson, L.E., *Differential PcP Travel Times and the Radius of the Core*, Geophysics Journal of the Royal Astronomical Society, Vol 39, Pg 435-456, 1974.
- [7] Taggart, J. N. and Engdahl, E.R., *Estimation of PcP Travel Times and the Depth to the Core*, Bulletin of Seismology Society of America. Vol 58, Pg 1293-1304, 1968.
- [8] Dziewonski, A. M., and Haddon, R. A. W., *The Radius of the Core-Mantle Boundary Inferred from Travel Time and Free Oscillation Data; A Critical Review*, Physics of the Earth and Planetary Interiors, Vol 9, Pg 38-35, 1974.
- [9] Voorhies, C. V., and Conrad, J, *Accurate Predictions of Mean Geomagnetic Dipole Excursion and Reversal Frequencies, Mean Paleomagnetic Field Intensity, and the Radius of the Earth's Core Using McLeod's Rule*, Technical Memorandum 104634, National Aeronautics and Space Administration – Goddard Space Flight Center, April 1996.
- [10] Sande, W. and Sande, C, **Hello World: Computer Programming for Kids and Other Beginners**, Manning Publication Company, 2014.
- [11] Ahern, T.K., and Titus, R., *Access to Seismological Data at the IRIS Data Management Center (IRIS-DMC)*, American Geophysical Union Fall Meeting, Vol 74, Pg 86, 1993.
- [12] Incorporated Research Institutions for Seismology, <www.ds.iris.edu>.
- [13] Wikipedia, *Geographical Distance*, <www.en.wikipedia.org/wiki/geographical_distance>.
- [14] Dziewonski, A.M. and Anderson, D.L., *Preliminary Reference Earth Model (PREM)*, Physics of the Earth and Planetary Interiors, Vol 25, Pg 297-356, 1981.

8.0 Appendix

Appendix A. Seismic Station Event Information.

Table A. Seismic Stations Meeting the Data Criteria.

| Earthquake Event | Date | Time (UTC) | Magnitude (M _w) | Depth (km) | Location | | Seismic Station | Angle (Deg) | Distance (km) | Location | | S-Wave Recording |
|---|------------|------------|-----------------------------|------------|----------|-----------|---|----------------------------|----------------------------------|-------------------------|----------------------------|------------------|
| | | | | | Latitude | Longitude | | | | Latitude | Longitude | |
| M6.1-S of the Fiji Islands | 4/2/2018 | 05:57:35 | 6.1 | 42 | -24.719 | -176.8865 | Northview High School, Brazil, IN | 105.23 | 11700.37 | 39.52 | -87.17 | Y |
| M6.0-1.16km SE of Lata, Solomon Islands | 7/17/2018 | 07:02:53 | 6.0 | 38 | -11.5936 | 166.432 | State Center, IA | 105.38 | 11717.38 | 41.91 | -93.22 | Y |
| M6.0-9.1km W of Kandrian, Papua New Guinea | 7/19/2018 | 18:30:32 | 6.0 | 30 | -6.1139 | 148.7302 | MacAlester College, St. Paul, MN L.A.S.A. Array, MT | 105.49 104.75 | 11729.77 11646.91 | 44.94 46.69 | -93.17 -106.22 | N Y |
| M6.0-Central Mid-Atlantic Ridge | 7/23/2018 | 10:35:59 | 6.0 | 10 | -0.2994 | -19.252 | Casper, WY Anaktuvuk Pass, AK | 104.97 104.88 | 11671.54 11661.31 | 42.65 68.13 | -106.52 -151.81 | N Y |
| M6.3-200km W-NW of Ile Hunter, New Caledonia | 9/10/2018 | 19:31:37 | 6.3 | 12 | -21.988 | 170.1584 | Knifeblade Ridge, AK Hockley, TX | 105.00 104.07 | 11674.84 11571.44 | 69.16 29.96 | -154.78 -95.84 | N Y |
| M6.3-Drake Passage | 10/29/2018 | 06:54:21 | 6.3 | 10 | -57.434 | -66.3894 | Oklahoma Geological Survey Observatory, Leonard, OK Troy Canyon, Curran, NV | 105.82 104.30 | 11766.23 11579.61 | 35.91 38.35 | -95.74 -115.59 | N Y |
| M6.1-80km SSE of Pangai, Tonga | 11/10/2018 | 08:33:21 | 6.1 | 35 | -20.4538 | -174.0081 | Dugway, Tooele County, UT Blacksburg, VA | 105.10 104.95 | 11686.23 11669.72 | 40.19 37.21 | -112.81 -80.42 | N Y |
| M6.4-101km E of Visokoi Island South Georgia and South Sandwich Islands | 11/15/2018 | 20:07:22 | 6.4 | 15 | -56.7065 | -25.546 | Nordonia Hills Middle School, Northfield, OH Blacksburg, VA | 105.13 104.71 | 11689.69 11642.94 | 41.32 37.21 | -81.54 -80.42 | N Y |
| M6.3-SE of Pacific Rise | 11/15/2018 | 23:09:01 | 6.3 | 10 | -56.2363 | -122.0441 | Tazewell, TN Erie, PA Minisink Valley Middle School, Slate Hill, NY | 105.31 104.56 105.55 | 11709.34 11626.33 11735.57 | 36.54 42.12 41.38 | -83.55 -79.99 -74.52 | N Y N |
| M6.2-SE of Easter Island | 12/19/2018 | 01:37:40 | 6.2 | 10 | -36.118 | -101.019 | Contact Creek, Katmai, AK Pilot Point, AK | 104.88 105.00 | 11661.60 11674.46 | 58.26 57.57 | -155.89 -157.57 | Y N |

Optimizing Flapping-Wing Flight

New Mexico

Supercomputing Challenge

Final Report

April 3, 2019

Team 56

New Mexico School for the Arts

Team Member:

Occam Kelly Graves

Teacher:

Jennifer Black

Table of Contents

| | |
|--|-----------|
| Table of Contents | 2 |
| Executive Summary | 3 |
| The Problem | 4 |
| Method | 4 |
| Figure 1: Variation on Wing Dimensions | 5 |
| Model Validation | 5 |
| Results | 6 |
| Figure 2: Large Range Analysis | 6 |
| Figure 3: Small Range Analysis | 7 |
| Figure 4: Example Console Output | 8 |
| Figure 5: Error In Graphing | 9 |
| Conclusion | 10 |
| Recommendations | 10 |
| Acknowledgments | 11 |
| Sources | 12 |

Executive Summary

This project focuses on finding the optimal wing dimensions and flapping frequency based on user-input wing and load parameters. These include load mass, wing density, wing thickness, and much more.

For each test, the upward force generated by the pair of wings is calculated and it is compared to the power to move the wing. The goal is to find the wing parameters that result in the most power efficient flight. This is achieved by normalizing the power data and the force data and calculating the difference.

This program currently offers two basic shapes for wings: ovals and rectangles. Rectangles tend to be slightly more energy efficient than the ovals, but they do require more power overall, so they could be more useful to lifting larger loads.

As was expected, larger wing width and flapping frequency produce more efficient results to a point in terms of energy consumption versus force generated. At some point, however, these large widths and fast frequencies become less efficient. What was unexpected was that the wing length did not seem to follow this pattern; longer wings always appeared more efficient.

The Problem

The purpose of this project is to find the optimal wing shape and frequency for flapping-wing flight given specific parameters about the load and wings. Fixed wings are great for large aircrafts, but for smaller objects such as robots or individual humans, fixed wings can consume much more power than flapping wings. (Gold, 2009) This project aims to eliminate this inefficiency by providing the precise parameters for the optimal flapping wing based on its purpose.

There are four main stages of insect and bird flight: downstroke, upstroke, pronation and supination. The pronation and supination are necessary so that the wing has less surface area on the upstroke than it does on the downstroke (Chin & Lentink, 2016). When designing this program, these were the four different states that I made calculations for.

The equations I developed were more than simple force calculations. To truly model a flapping wing, I had to model the velocity of the wing as a function. This posed some problems because now the velocity variable in the source equation I used was a function rather than an easily substitutable variable. This especially made the integration of the force equations more difficult.

Method

Using the same four stages of flight, I created four equations to correspond to the net upward or downward forces and the overall power required throughout these stages.

I then summed these calculations to arrive at an equation for total force and power per cycle.

For each equation, I started with the equation for air resistance: $F_D = \frac{1}{2} \rho C_D A v^2$ where ρ = air density = 1.225 kg/m^3 , C_D = coefficient of drag which is 2 for flat surfaces like the wing, A = area, and v = velocity. I used a negative cosine function for the velocity over time. Using integration across the wing and integration again across the swath of the wing downstroke (or upstroke), I arrived at a final equation for the downstroke and upstroke. I then subtracted the downward force of gravity based on mass of load and mass of wings using density and dimensions.

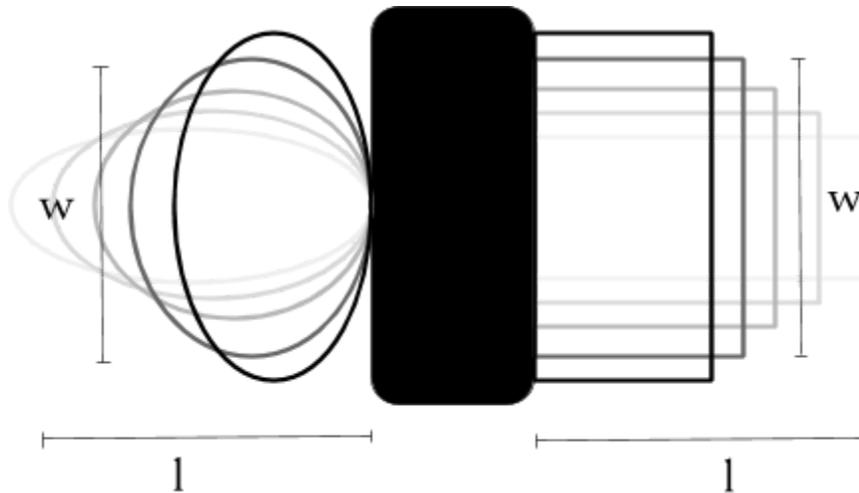


Figure 1: Variation on Wing Dimensions

Model Validation

To validate the model, I cross referenced basic properties of aerodynamics to make sure the results produced were logical. As expected, my results showed lower force

generation with smaller wings and lower frequencies. The power requirements also followed this pattern as expected. When differencing the normalized power and force data, the

Results

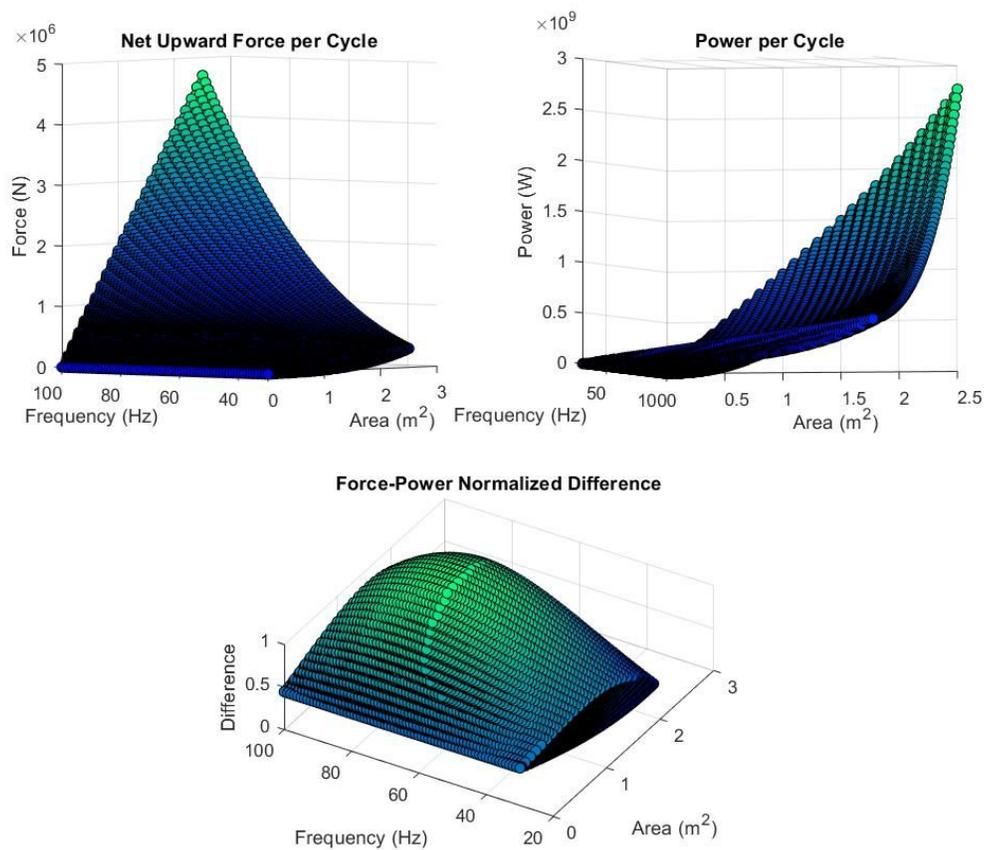


Figure 2: Large Range Analysis

I used larger ranges with less accuracy to locate the peaks of the normalized difference graphs.

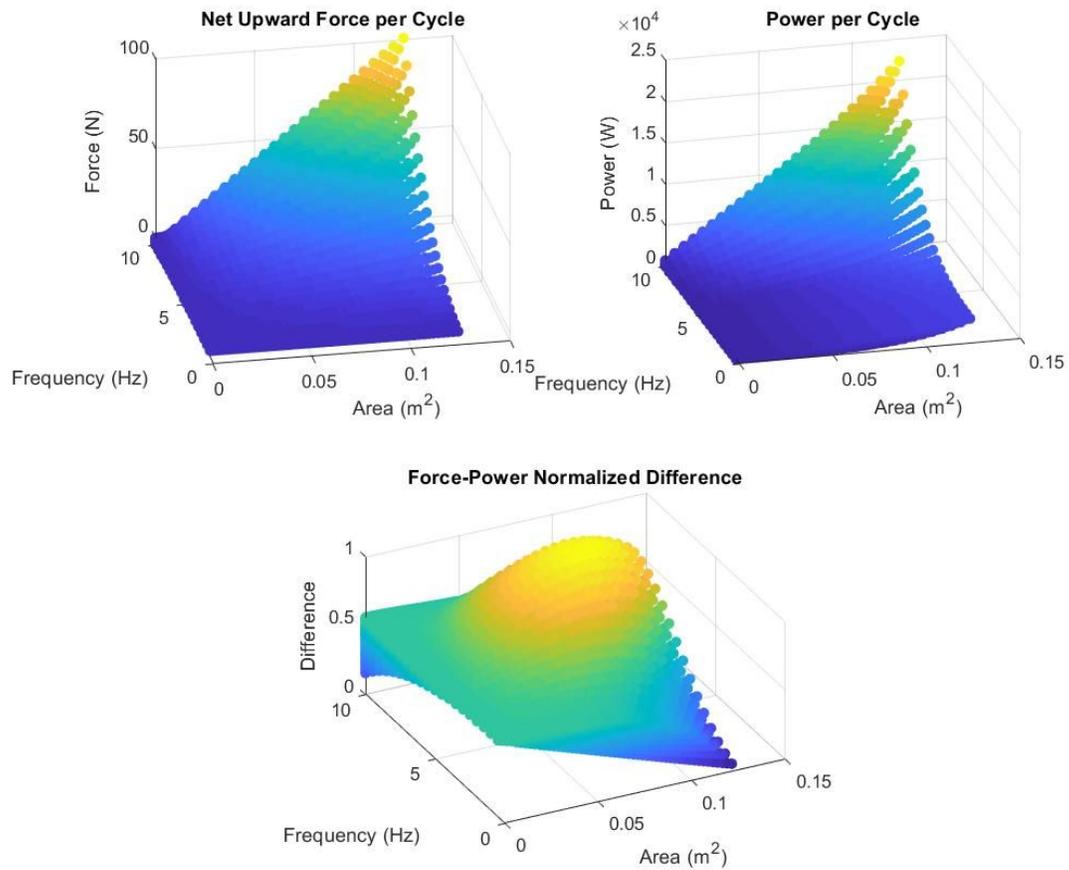


Figure 3: Small Range Analysis

After using larger ranged graphs to find the peaks, I can use smaller ranges to more precisely locate the optimal wing parameters.

```
>> findOptimalParameters(.5:.5:10, 0:.01:.5, 0:.01:.25, .001, 2*pi/3, 0.01, 10, .02, "oval")
Optimal parameters at:
2.0944 Radians Stroke Angle,
0.0010 m Wing Thickness,
10.0000 kg/m^3 wing Density,
0.0100 s proation/supination rotation time,
and 0.02 kg Load Mass

Optimal Width: 0.18 m
Optimal Length: 0.50 m
Optimal Frequency: 8.0 Hz

Upward Force Generated: 32.12 N
Power Required: 5196.12 W
```

Figure 4: Example Console Output

Output to the console is the data about the highest point on the normalized difference graph. This is useful if designing a wing because it shows the precise optimal values the computer found.

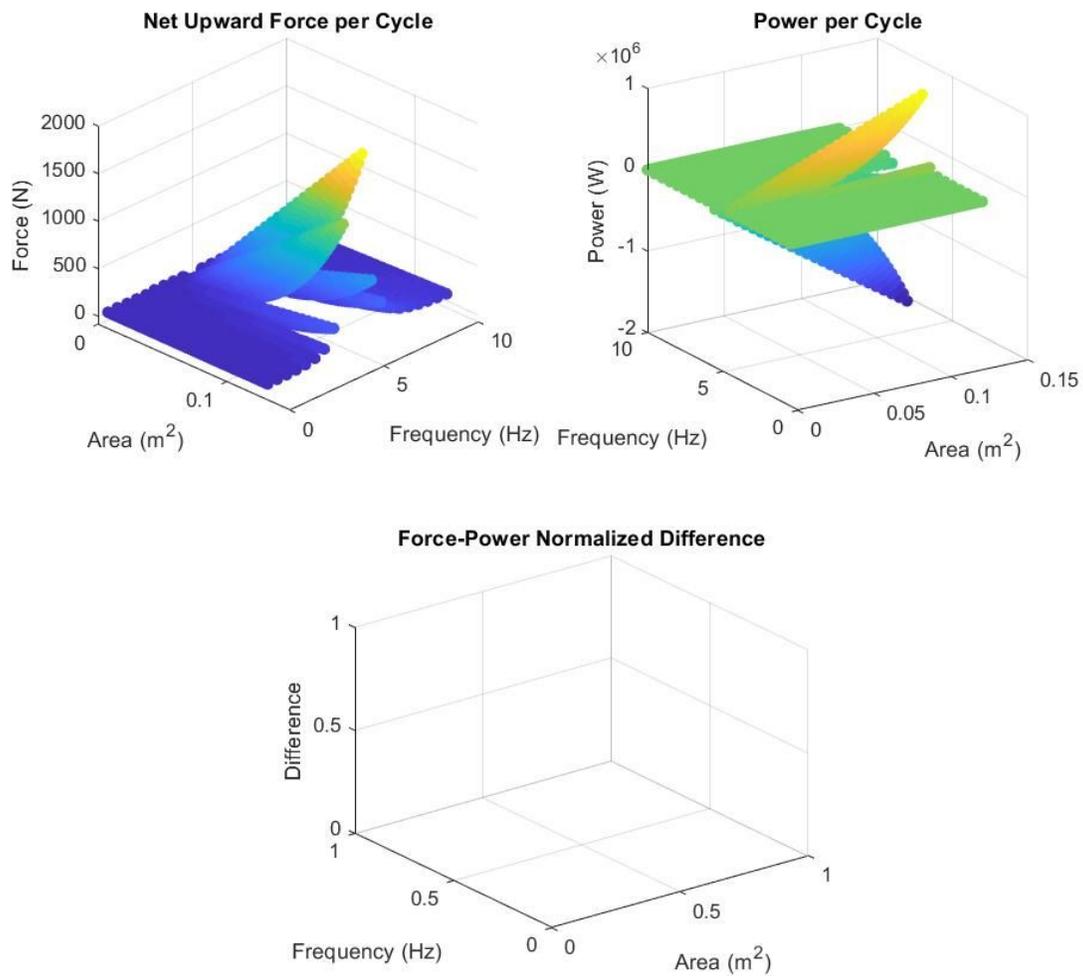


Figure 5: Error In Graphing

I have found an unusual error when the time it takes to rotate the wing for pronation and supination is set to a higher value. This produces a large spike around a specific frequency, which is unnatural when considering the nature of force and power relation to flapping frequency. Error handling could be implemented to prevent acceptance of the large turn time values.

Conclusion

Both the wing width and the flapping frequency appear to have a point of diminishing returns where the cost of the power becomes greater in proportion to the force produced than lower values. Strangely, however, the wing length did not appear to follow this pattern, so this suggests that an infinitely long wing would be optimal. Using logical reasoning, it is obvious that this is untrue, so there must be a factor I have not accounted for that limits the length of optimal wing shape in the real world.

More important than the statistical data I retrieved from this program, this program has the potential to be used as a personalized optimizer calculator. It also has the potential to be incorporated into larger programs that could find trends in wing shape based on load mass, flapping angle, or desired speed. Although by itself this program just finds optimal wing shape and frequency, it can be easily incorporated into new code to accomplish so much more.

Recommendations

I wrote this program with the intent that it could be easily editable so that a multitude of changes can be made to it to increase the precision as well as the range of tests that it would go through and optimize. That being said, I do have some specifics in mind that could be added to this program to enhance the quality of the results.

Firstly, the program is made for direct induction and external method calls, so it is far from as user friendly as it could be. A user interface could be added that prompts

the user for the specific data that is needed such as the wing thickness and load mass.

After the data is processed, graphs could be used to show the dimensions of the optimal wing shape.

The precision of the model could also be improved. There are five actions flapping organisms utilize to increase lift. These are leading edge vortices, added mass, wing-wake interactions, rotational circulation, and clap and fling. (Chin & Lentink, 2016) I did not include these in this model because they are much harder to analyze due to their seemingly random behaviors. According to Anderson, 2011, “ if the MAV designer builds to the quasi-steady model, he can expect to be able to generate greater lift than expected, but will also experience greater drag, and thus, greater power requirements.” This would likely cause a shift in the point with the greatest lift to power, thus slightly changing the optimal point.

Other features could be added for cross analysis with variation in other parameters such as load mass. If a function were created for this purpose, an equational model could be created to predict the how load mass relates to wing shape and frequency.

Acknowledgments

I'd like to thank Celia Einhorn and Bill Blackler for helping link me to possible mentors and for making sure I always had what I needed.

I appreciated the assistance in understanding some of the concepts in this field provided by Alice Bischoff, Major Hanks, and Dr. Anderson.

I'd also like to thank Chris Morrison and Mohit Dubey for reading my papers and presentations throughout the challenge and providing useful feedback and resources I could use to further my research.

Sources

Anderson, M. L. (2011). Design And Control Of Flapping Wing Micro Air Vehicles. *Air Force Institute of technology*. Retrieved from

<https://apps.dtic.mil/dtic/tr/fulltext/u2/a549053.pdf>

Chin, D. D., & Lentink, D. (2016). Flapping wing aerodynamics. *Journal of Experimental Biology*. Retrieved from <http://jeb.biologists.org/content/219/7/920>

Compressibility and Incompressibility. (n.d.). *Centennial of Flight*. Retrieved from

<https://www.centennialofflight.net/essay/Dictionary/Compressibility/DI136.htm>

Hartnett, K. (2018). What Makes the Hardest Equations in Physics So Difficult? *Quanta Magazine*. Retrieved from [https://www.quantamagazine.org/what-makes-](https://www.quantamagazine.org/what-makes-the-hardest-equations-in-physics-so-difficult-20180116/)

[the-hardest-equations-in-physics-so-difficult-20180116/](https://www.quantamagazine.org/what-makes-the-hardest-equations-in-physics-so-difficult-20180116/)

Gold, L. (2009). To flap, or not to flap? *Cornell Chronicle*. Retrieved from <http://news.cornell.edu/stories/2009/09/birds-fly-more-efficiently-airplanes-study-shows>

Knight, K. (2010). How Birds Power Flight. *Journal of Experimental Biology*. Retrieved from <http://jeb.biologists.org/content/213/16/i>

Lehmann, F. O., & Pick, S. (2007). The aerodynamic benefit of wing–wing interaction. *Journal of Experimental Biology*. Retrieved from <http://jeb.biologists.org/content/210/8/1362>

Tobalske, B. W. (2007). Biomechanics of bird flight. *Journal of Experimental Biology*. Retrieved from <http://jeb.biologists.org/content/210/18/3135>

