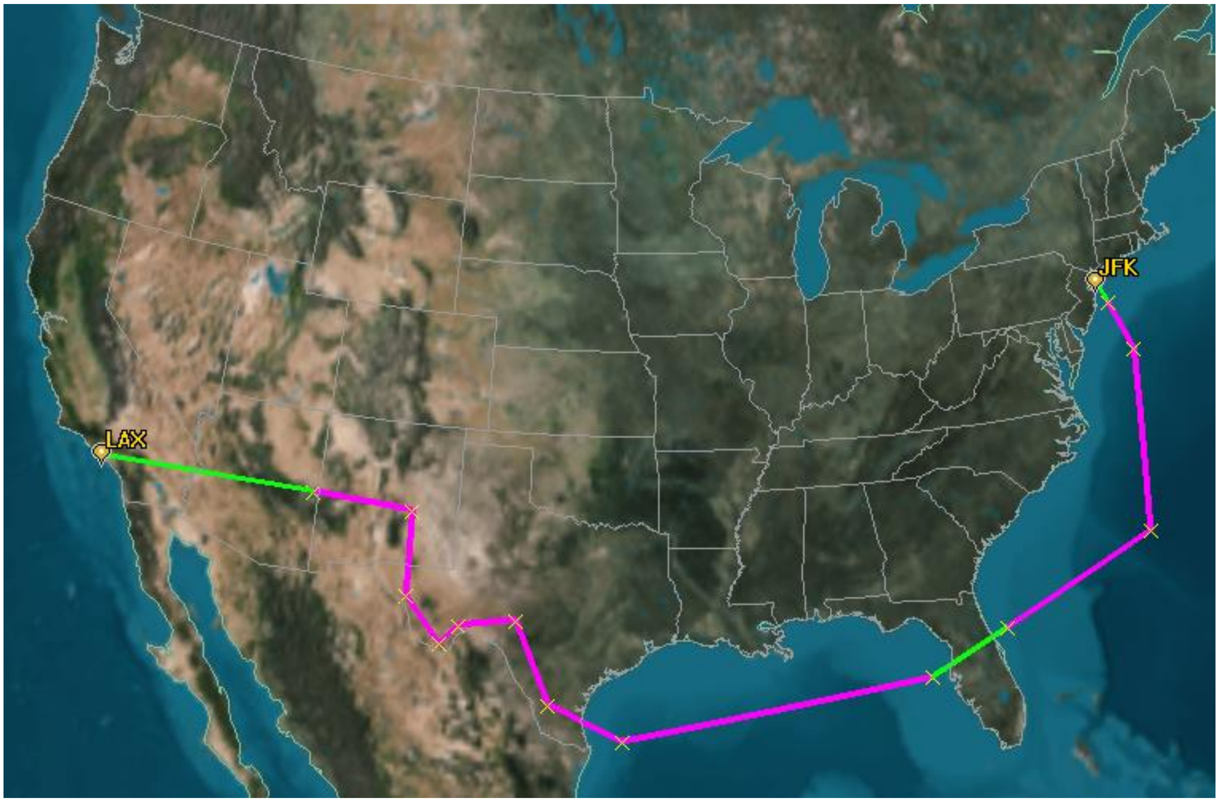
Optimization of the LAX – JFK Flight Route to Minimize Population Exposure to Sonic Booms



Prepared for:

Capstone Library
Engineering Honors Program
University of Michigan – Ann Arbor

Prepared by:

George William Adamson

December 16, 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

BACKGROUND

For those unfamiliar with supersonic aircraft, sonic booms, and/or the Concorde, this section will provide valuable background information. I highly recommend that people with a non-aerospace background read this section. It may be challenging to understand my research project without it.

Supersonic Aircraft

Supersonic aircraft, or aircraft that can travel faster than the speed of sound, have been around since the mid-1900s. The bulk of supersonic aircraft are military aircraft, but there have been a few commercial aircraft including the Tupolev Tu-144 and Concorde. Both of these aircraft have been retired, and today, all commercial aircraft fly below the speed of sound. The reasons behind this will be discussed in later sections.

Supersonic aircraft are typically defined by their Mach number, which is a ratio of the aircraft's airspeed relative to the speed of sound. This means that Mach 1 would refer to an aircraft travelling at exactly the speed of sound, and Mach 2 would mean it was travelling at twice the speed of sound.

Sonic Booms and Regulations

Sonic booms are one of the defining characteristics of supersonic aircraft. They are described as "a thunder-like noise a person on the ground hears when an aircraft or other type of aerospace vehicle flies overhead faster than the speed of sound, or supersonic" [9]. They are caused by a "cone of pressurized or built-up air molecules, which move outward and rearward in all directions and extend all the way to the ground. As this cone spreads across the landscape along the flight path, it creates a continuous sonic boom along the full width of the cone's base. The sharp release of pressure, after the buildup by the shock wave, is heard as the sonic boom" [9]. This loud, thunderous noise that accompanies a sonic boom is one of the primary reasons for the lack of commercial supersonic aircraft today.

During the 1960s and 1970s, there were several tests to document the effects of sonic booms on people and property. The research revealed that "three aspects of sonic booms were found to be most disturbing – (1) people being startled, (2) structural component vibrations and rattles, and (3) rising concerns over the possibility of sonic boom induced structural damage" [16, p. 518]. This research ultimately caused the Federal Aviation Administration (FAA) to impose an outright ban on supersonic flight over the United States in 1973. The text of this regulation (14 CFR 91.817) is shown below [6]:

(a) No person may operate a civil aircraft in the United States at a true flight Mach number greater than 1 except in compliance with conditions and limitations in an authorization to exceed Mach 1 issued to the operator under appendix B of this part.

(b) In addition, no person may operate a civil aircraft for which the maximum operating limit speed exceeds a Mach number of 1, to or from an airport in the United States, unless -

(1) Information available to the flight crew includes flight limitations that ensure that flights entering or leaving the United States will not cause a sonic boom to reach the surface within the United States; and

(2) The operator complies with the flight limitations prescribed in paragraph (b)(1) of this section or complies with conditions and limitations in an authorization to exceed Mach 1 issued under appendix B of this part.

As of today, this FAA regulation is still in place. However, on March 30, 2020, the FAA released a statement that said, "the FAA is assessing the current state of supersonic aircraft technology in terms of mitigating the noise impacts associated with supersonic overland flight. To this end, Section 181 [of the FAA Reauthorization Act of 2018] also requires a biennial review of aircraft noise and performance data beginning on December 31, 2020 to determine whether to amend the current ban on supersonic flight by civil aircraft over land in the United States" [19]. Thus, while discussions are taking place within the FAA, it remains to be seen if new commercial supersonic aircraft will ever fly over the United States.

History of the Concorde

The Concorde is arguably the most famous commercial supersonic aircraft to have existed. It flew from 1976–2003 and had a max speed of Mach 2.04 or 1354 mph [1]. For reference, a relatively new commercial aircraft in use today, the Boeing 787, has a max speed of Mach 0.9 or 667 mph [22, p. 15].

Figure 1: Concorde Landing at Washington Dulles International Airport [15]

One may ask, why did the Concorde stop flying in 2003 and why are newer aircraft regressing in cruise speed? There were many reasons for the Concorde's retirement, including a drop in air travel after 9/11, rising maintenance costs, inefficient fuel consumption, and an outdated cockpit

[1]. In addition, the Concorde required three pilots: the captain, co-pilot, and flight engineer. Most other aircraft had retired the flight engineer role due to increases in automation. This meant that the Concorde had a much higher operational cost than other aircraft. There was also a major crash involving the Concorde on July 25, 2000. Air France Flight 4590 crashed shortly after takeoff, resulting in 113 fatalities [2]. Based on these factors and the Concorde's limitation to flights only over the ocean, it became increasingly difficult for the Concorde to compete with subsonic aircraft. Thus, the Concorde was eventually retired and last flew on November 26, 2003.

INTRODUCTION

In this section, I will discuss how the research capstone began and provide a high-level overview of the project.

Project Conception

I began brainstorming this project during the summer of 2019. I had always enjoyed reading about the Concorde, and I began to wonder what it would look like if the Concorde flew across the United States. Specifically, how disturbing would the sonic booms be to the people living below the flight path?

As I conceptualized this idea, I realized the need to focus on a single flight route. I immediately thought of LAX - JFK. It was an ideal route for several reasons. It connected two cities on opposite coasts, it is widely traveled, and it would benefit from a reduced flight time. I did some quick research and found that the typical flight time is four to five hours (wheels up to wheels down). If the Concorde was used, this flight time could theoretically be reduced to less than two hours. There were certainly many other routes I could have chosen, and with more time, it would have been interesting to look at some of them. For now, the LAX - JFK route became the focus of my project.

At this point, I formulated a basic research question: how many people would be impacted by sonic booms if the Concorde flew from LAX - JFK? However, this question was a bit too trivial for a senior capstone; I needed to go a step further. I concluded that an optimization-style project made sense. Instead of quantifying the population impaction on a single LAX - JFK route, I could optimize the route such that the population impaction from sonic booms was minimized.

Project Approval

In January of 2020, I met with Amy Cohn, a professor in the Industrial and Operations Engineering Department. I had been referred to Professor Cohn by Rachel Armstrong-Ceron, the Engineering Honors Academic Advisor. We discussed my initial idea for the project, and she agreed to be my capstone supervisor.

Research Questions

In any optimization problem, a baseline must be defined. Otherwise, there is nothing to compare optimal results against. My project contains two optimization variables: flight time and the number of people impacted by sonic booms. Thus, I need a baseline metric for both of these variables.

The first baseline is quite simple; I need to determine the flight time for a typical LAX - JFK flight. However, the baseline for the number of people impacted by sonic booms is less clear. The Concorde never flew from LAX - JFK, so there is no historical precedent from which a baseline metric could be derived. To solve this, I modeled a Concorde flight from LAX - JFK using the same route as the first baseline. As you will read later on, I can use this model to estimate the number of people impacted by sonic booms. Because the second baseline is merely the first baseline’s route with a supersonic aircraft instead of a subsonic aircraft, I named these the subsonic and supersonic baselines.

With the two baselines defined, I had successfully refined my optimization problem into three concrete research questions:

1) Subsonic Baseline: What is the flight time for a typical LAX - JFK flight today?

2) Supersonic Baseline: If the Concorde flew on the subsonic baseline’s route, how many people would be impacted by the sonic booms?

3) Optimal Route: If the LAX - JFK route was optimized such that the population impaction from sonic booms was minimized, what would be the resulting flight time and population impaction? How does this compare to the subsonic baseline’s flight time and supersonic baseline’s population impaction?

This research project has two major components. The first goal is to analyze the baseline route in both the subsonic and supersonic case. Once that is completed, I can begin optimizing the route with the goal of generating a route that impacts fewer people than the supersonic baseline whilst flying faster than the subsonic baseline. This it outlined in Table 1 below.

Table 1: Baseline and Optimal Route Characteristics

	Aircraft	**LAX-JFK Route**
Subsonic Baseline	Boeing 767-300	Typical Route In-Use Today
Supersonic Baseline	Concorde	Typical Route In-Use Today
Optimal Route(s)	Concorde	Optimized

Why is the Project Important?

This project is important because supersonic aircraft have the ability to reduce flight times significantly, which could benefit both passenger and cargo transport. While these aircraft have already been proven to be feasible when flown over the ocean, they have struggled to make inroads into overland routes.

A flight from LAX – JFK is lengthy, requiring the better part of a day. The flight takes approximately five hours, and three hours are lost from crossing time zones. Decreasing the flight time would surely be met with appreciation by passengers and cargo transport companies.

Also, I should note that my project is not the only form of research into reviving supersonic aircraft for overland routes. In fact, the predominant research for tackling this problem is largely fixated on making sonic booms quieter. Look no further than NASA and Lockheed Martin which are developing the X-59 QueSST, "an experimental piloted aircraft designed to fly faster than sound without producing the annoying – if not sometimes alarming – sonic booms of previous aircraft" [10]. While the X-59 QueSST is certainly an exciting development for the future of commercial supersonic aircraft, there is certainly a question of whether the sonic booms will be quiet *enough*. It is for this reason that research like mine, examining potential overland routes for these aircraft, will likely be important. Even if the sonic booms themselves are made quieter, but not silent, would you rather deploy them on routes with 40,000,000 or 300,000 people hearing them? Unless supersonic aircraft reach a point where they are just as loud as subsonic aircraft, it is not unthinkable to foresee a future where aircraft routes have to be redesigned such that the population exposure to sonic booms is minimized.

OUTLINE OF PROBLEMS ADDRESSED

This section contains an outline of the problems that needed to be addressed in order to solve this optimization problem. They are presented in chronological order.

Develop a Baseline Route

1) What is (one of) the most common routes an aircraft takes when flying between LAX and JFK?

Characterize the Baseline Route

1) How long does a typical subsonic aircraft take to fly the baseline route?
2) If the baseline route was flown by a supersonic aircraft (Concorde), how many people would be impacted by sonic booms?
3) What is the flight time of the supersonic baseline and how does it compare to the subsonic baseline?

Search for Data Sources

1) Find population data for the United States, Canada, Mexico, Cuba, and the Bahamas at a fine scale (relative to the size of the sonic boom carpet).
2) Assemble a database of waypoints across the United States, Gulf of Mexico, and Atlantic Ocean. Ensure that the number of waypoints is sufficient such that an aircraft has the ability to maneuver around population centers.

Optimize the LAX – JFK Route

1) Reduce the computational expense of the optimization problem by only considering distant neighbors for each waypoint.
2) Develop a “cost” matrix for flying between each pair of waypoints, where “cost” signifies the approximate number of people impacted by sonic booms for a supersonic aircraft traveling between the pair of waypoints at hand.
3) Implement a Dijkstra algorithm to solve the cost matrix and find an optimal route between LAX and JFK.

Post-Process the Route

1) Introduce speed limits along the baseline and optimal routes. These speed limits should be placed near the Los Angeles and New York City metropolitan areas (and other areas as needed) where it is impossible to avoid large population centers.

Characterize the Optimal Routes

1) What are the flight times of the optimal routes and how do they compare to the subsonic and supersonic baselines?
2) How many people are impacted by sonic booms on the optimal routes and how do these figures compare to the subsonic and supersonic baselines?

METHODS

In this section, I will discuss my methods for solving each of the problems listed in the outline.

Baseline Route and Aircraft

The baseline is what commercial aviation looks like today; in other words, what route would an aircraft take to fly between LAX and JFK? Also, how long would that flight take? This may seem like a simple task, but there are a few complexities. The route that an aircraft flies from LAX - JFK may change based on weather and other factors. Also, different airlines may use different aircraft. To mitigate this, I utilized the *flightaware.com* website. One feature of this website is its ability to determine the most common LAX - JFK route being flown in a two-week period [13]. At the time of conducting this research, the most common route was the following, shown below in Figure 2.

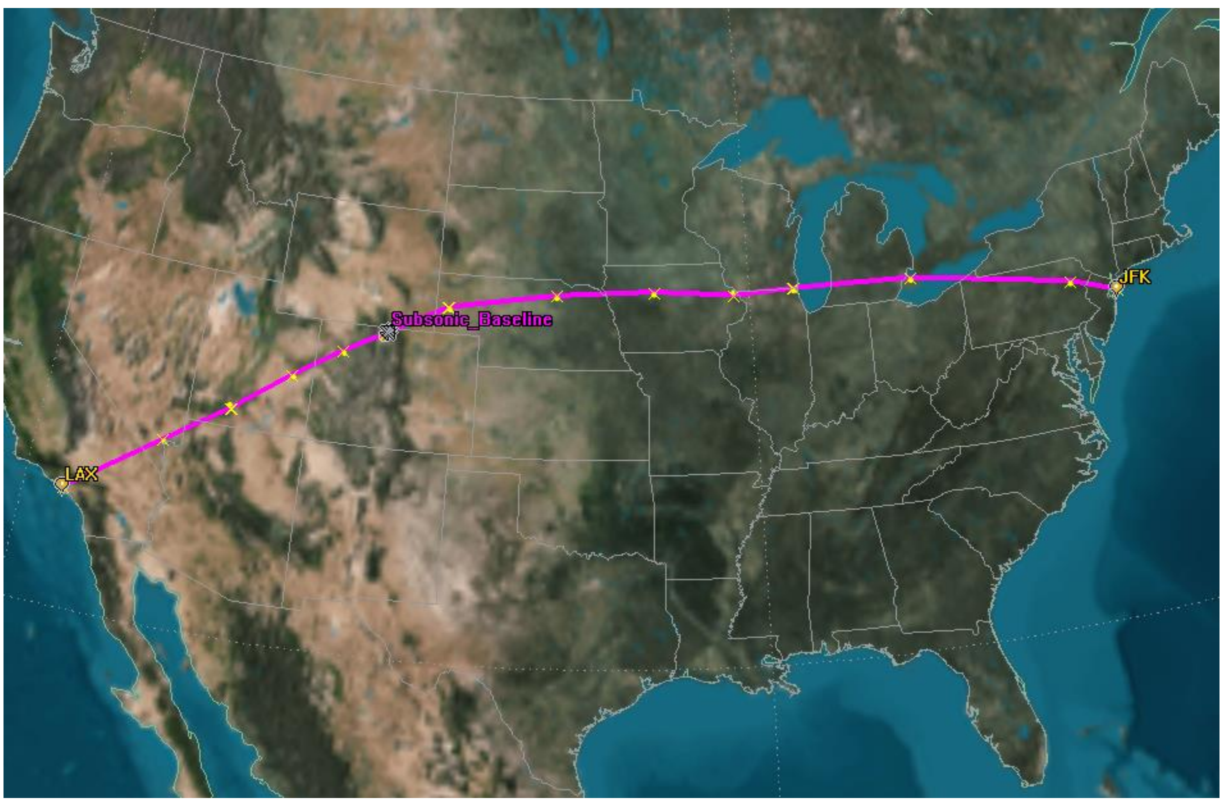


Figure 2: Most Common LAX - JFK Route Flown in a Two-Week Period (As of Late Jan. 2020)

The waypoints that make up this route are: `LAX BEALE BAWER LARVE EKR BFF ONL FOD DBQ KG75M DAFLU LVZ JFK`. These waypoints are marked as yellow X's on the route. They act as intermediate points to help the aircraft navigate from LAX to JFK.

For the baseline aircraft, I used the Boeing 767-300. It should be noted that the Airbus a321 is another common aircraft that flies between LAX and JFK. However, both aircraft have very similar cruise speeds of 489 knots and 487 knots, respectively [14]. Thus, choosing one over the other will not have a pronounced effect on the flight time of the subsonic baseline.

Data Sources

This project required two major datasets: population data and aircraft waypoint data.

Population Data: When I began this project, I only considered population data for the United States. However, upon analyzing several optimized routes, I began to notice aircraft routes that entered Canadian and Mexican airspace. Without population data in these regions, my results would be severely erroneous. To combat this, I expanded my population dataset to include the nearby nations of Canada, Mexico, Cuba, and the Bahamas.

I obtained my United States population data from the U.S. Census Bureau, specifically the 2010 Census (the 2020 Census had not yet been completed). The U.S. Census Bureau reports population data on several different scales, and I decided to use the finest scale possible: the block group [5]. The dataset was composed of 220,334 block groups, and the average population for a block group is 1,418 people. It is important to use a population dataset with a fine scale because the entire project revolves around choosing pairs of waypoints with the fewest people under the flight path between them. If the dataset is too coarse, datapoints will be grouped together (away from their actual locations), and the optimization algorithm may fail to generate a well-optimized route.

The Canadian population data is from Statistics Canada (StatCan), a Canadian government agency [4]. This data only included city names and their corresponding population. To convert city names into latitude/longitude pairs, I utilized the Google Map's Geocoding API [8]. The Canadian dataset is only composed of the 1,005 largest cities in Canada. This means that the dataset's resolution is quite coarse compared to the U.S. However, this is not a major concern as the generated routes do not actually enter Canada. The purpose of the Canadian data is to "discourage" the algorithm from thinking that Canada is uninhabited.

The population data for Mexico, the Bahamas, and Cuba was a bit more difficult to find. Eventually, I settled on the website *worldpopulationreview.com* which reports the population of the largest cities in each of these countries [23]. I would have liked to use a government source as I did for the U.S. and Canada, but I was unable to find one. Fortunately, the accuracy of the data for these countries is less important because the routes only fly near these countries, not over them. The population data for each country is outlined in Table 2 below.

Table 2: Population Dataset Breakdown by Country

Country	Resolution	# of Entries in Dataset
United States	Fine (Block Group)	220,334
Canada	Coarse (City)	1,005
Mexico	Coarse (City)	400
The Bahamas	Coarse (City)	19
Cuba	Coarse (City)	147

Each entry of the population dataset has three values separated by commas: latitude (in deg N), longitude (in deg W), and population. An example entry is shown below:

38.327457,122.492224,962

In the end, my population dataset had a total of 221,738 entries. A whopping 99.4% of these entries lie in the United States. While this may seem concerning, the point of the non-U.S. data is to dissuade the optimization algorithm from thinking that nearby countries are uninhabited (and thus a good place to fly supersonically). With a small number of international datapoints, the algorithm realized that these countries were indeed inhabited and picked a more realistic route accordingly. A visualization of the population dataset is shown below in Figure 3.

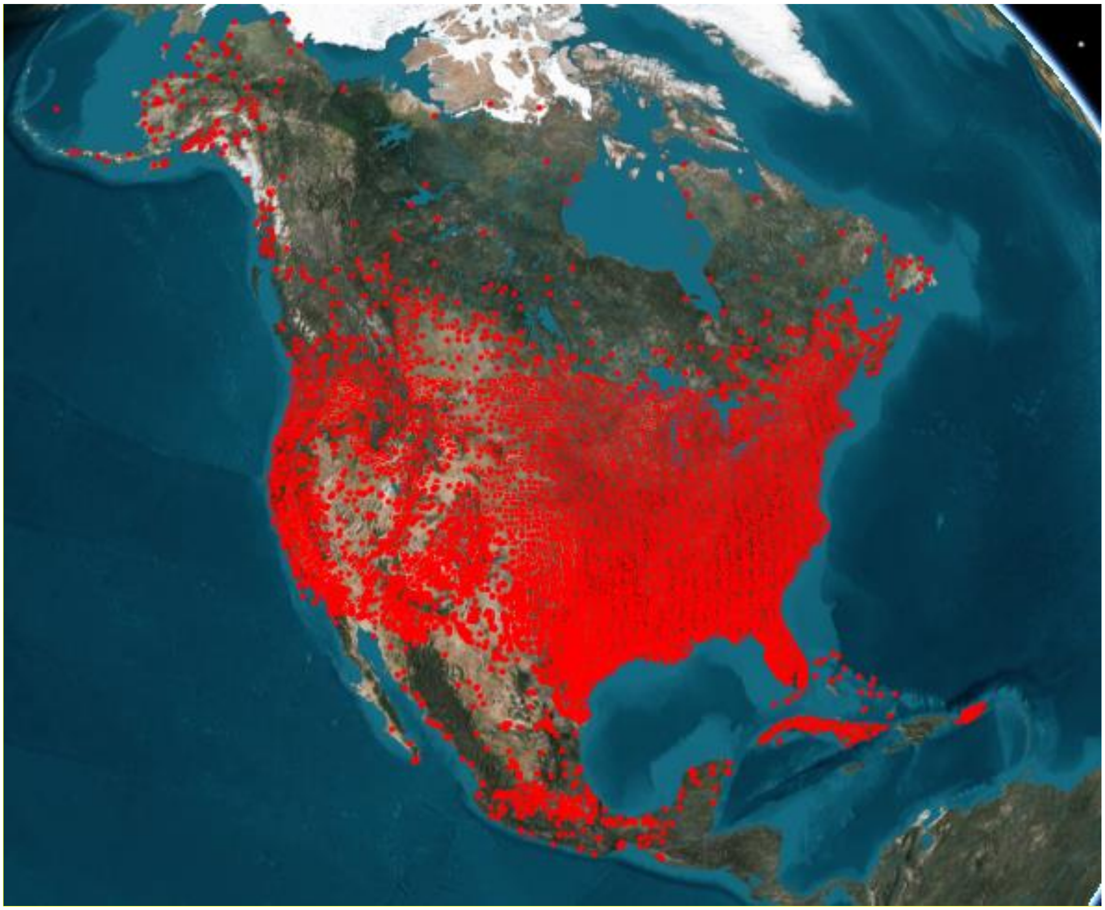


Figure 3: STK Visualization of Population Dataset

Aircraft Waypoint Data: Before I could optimize an aircraft route between LAX and JFK, I needed to compile waypoint data across the contiguous United States (CONUS) and nearby waters. These waypoints act as intermediate points between LAX and JFK that are preferentially chosen such that the supersonic aircraft avoids flying over large population centers.

Aircraft waypoints have evolved over the years from radio beacons to more modern GPS points. With radio beacons being phased out by the FAA in the coming years, I chose to limit my dataset to entirely GPS waypoints. These waypoints follow a naming convention of five-characters, with an example waypoint being:

42.336997,83.893683,DEXER,MI,US

Those acquainted with geography around the University of Michigan, may recognize that DEXER is referring to Dexter, MI, a city just northwest of Ann Arbor. The five-character waypoint name can often be a reference to its physical location on the ground. The latitude (in deg N) and longitude (in deg W) of the waypoint can also be seen alongside the name.

I used the GPX Aviation Waypoint Generator, a subset of the *navaid.com* website, to generate waypoints for my project [11]. I limited my waypoints to CONUS and its surrounding oceans because waypoint data for other countries can be outdated and/or erroneous. With these constraints, I generated 61,011 waypoints across CONUS. The distribution of these waypoints is shown below in Figure 4.

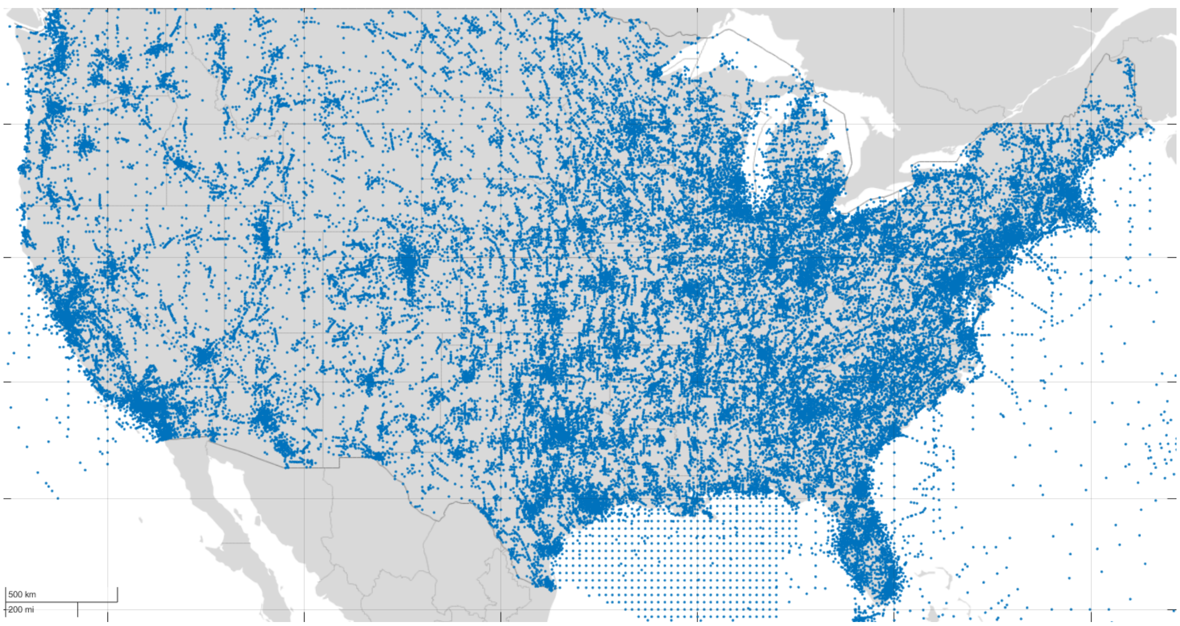


Figure 4: GPS Waypoint Distribution Across CONUS

It should be noted that there is a decrease in waypoint density near the Rocky Mountains, and this must be kept in mind when generating aircraft routes. In a region with decreased waypoint density, it becomes more difficult to find a path of waypoints that avoids population centers. Fortunately, the population also happens to be lower over this region, so it is not a major concern.

Model of Sonic Boom Carpet

Next, I created a model for the area at the surface in which people would hear the sonic booms produced by a supersonic aircraft flying overhead. This area is commonly referred to as the sonic boom "carpet."

Figure 5 shows the shock wave ray paths that propagate from an aircraft flying out of the page at 60,000 feet. Of most importance are the ray paths that travel directly to the ground from the aircraft, shown on the bottom left of the figure. These ray paths make up the primary carpet which extends about 25 nautical miles (nmi) from each wing. Thus, the entire primary carpet is approximately 50 nmi in width. The sonic booms in the primary carpet are "on the order of 1 lb/ft^2 to 3 lbs/ft^3 in intensity" [16].

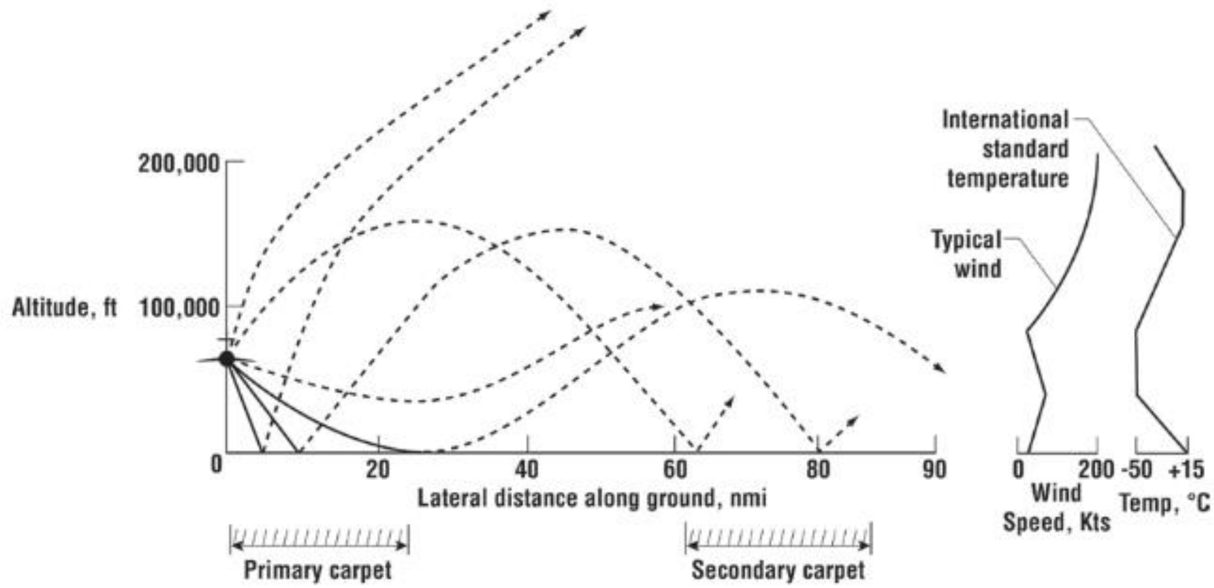


Figure 5: Shock Wave Ray Paths for Supersonic Aircraft at 60,000 ft [16]

The secondary carpet is also notable; it is formed by ray paths that travel upwards initially, bend in the upper atmosphere, and eventually hit the ground. I chose to neglect the secondary carpet in my sonic boom model. I made this decision because "the [secondary carpet] disturbances tend to be very weak in intensity (on the order of 0.02 to 0.20 lb/ft^2)" [16]. The intensity is important because it relates to my optimization variable: the number of people *disturbed* by sonic booms. By modeling two carpets with different intensities, the disturbance would become a variable of its own, and it would no longer be sufficient to simply count the number of disturbed people. Thus, for the purposes of my capstone, I will only be modeling the primary carpet. In a more intense research study, the secondary carpet could be included alongside a ranking system where people impacted by the primary carpet are ranked as being more disturbed than those in the secondary carpet.

To model the primary carpet, I used an STK sensor object which is defined using a cone half angle. Using basic trigonometry, I could relate my cruise altitude (60,000 ft) and primary carpet lateral distance (25 nmi or 151,902.89 ft) to the cone half angle. This is shown in Figure 6.

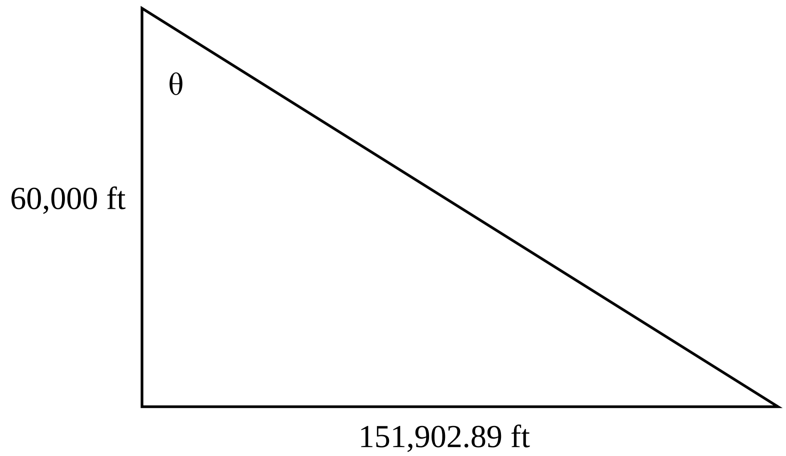


Figure 6: Trigonometric Relation to Determine Sensor Cone Half Angle

Using inverse tangent, it can be shown that the cone half angle is 68.4465 deg. A screenshot of the sensor object with this cone half-angle is shown below in Figure 7.

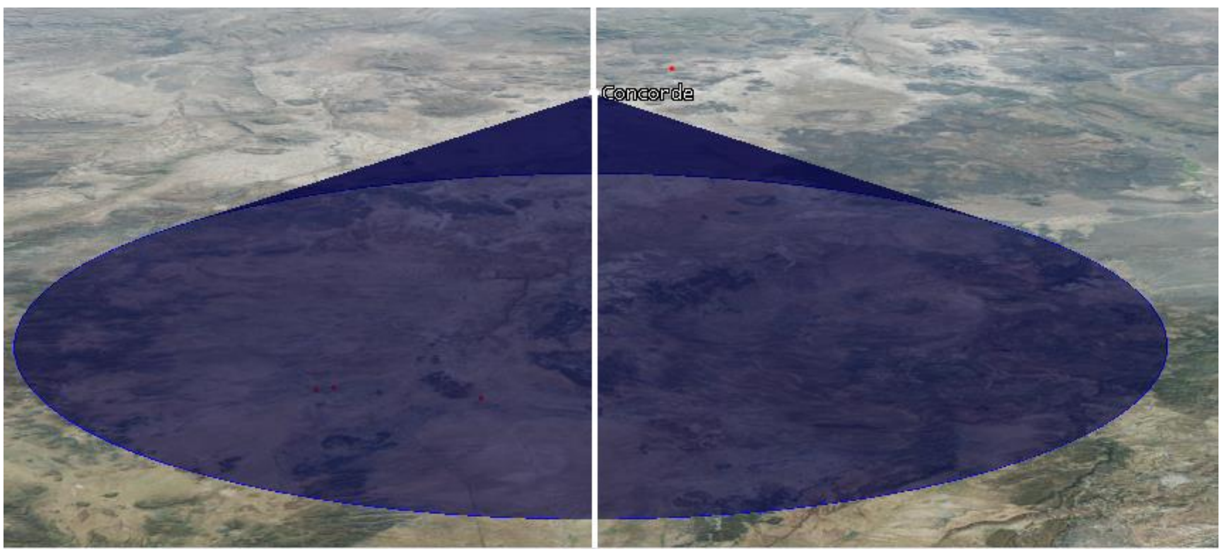


Figure 7: STK Sensor Object Used to Model Primary Carpet at Surface

Since the sensor object is attached to the supersonic aircraft, it will be projected onto the surface along the flight path. The net result is that after flying the entire route, the projections will be piled on top of each other, creating the primary carpet. This is shown on the following page in Figure 8, with the primary carpet shown as the red area, the flight path as a white line, and the sensor object as a blue circle.

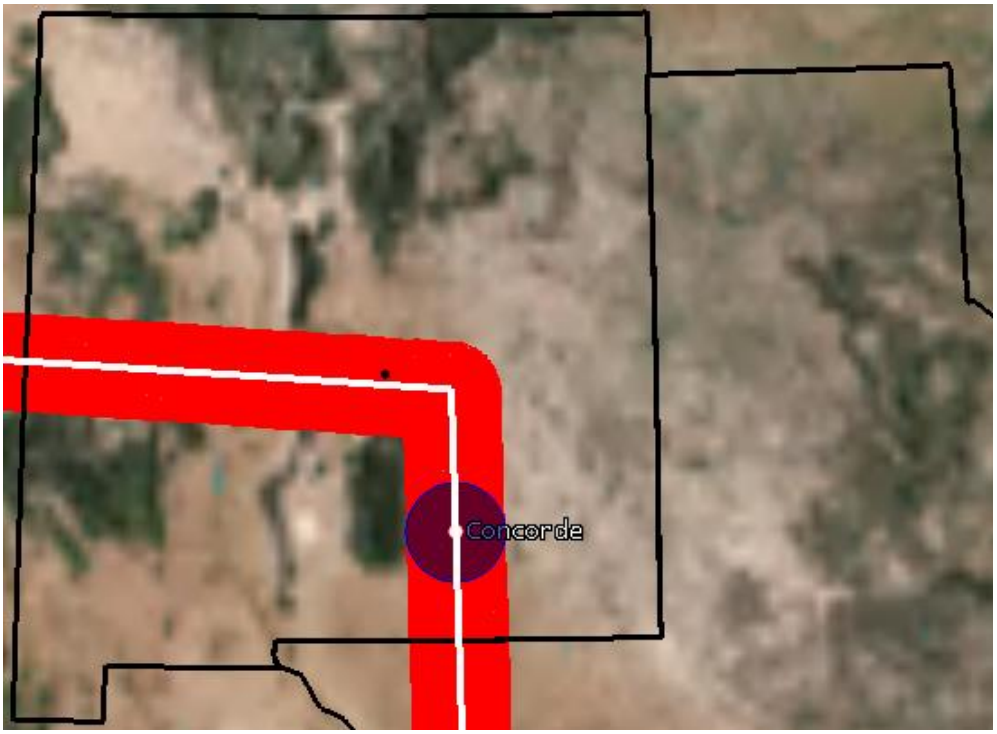


Figure 8: Primary Carpet Modeled Along the Flight Route

Python

In this section, I will provide a high-level overview of some of the Python scripts I developed. For more information on any particular script, please consult my GitHub (GeorgeAdamson23).

Data Consolidation and Formatting: One of my first tasks was to use Python to consolidate and format all of my data.

The aircraft waypoints from *navaid.com* were stored in a (.xml) format with a lot of unnecessary data. I used the script *readWaypoints.py* to reformat the waypoint data into a (.txt) file with one waypoint on each line. The format for the output file is:

WP_NAME,LATITUDE,LONGITUDE,STATE_ABBREVIATION,COUNTRY

I also had population data from a variety of sources that were stored in different formats. I used several scripts to consolidate all of the data into a single (.txt) file with one population datapoint on each line. The format for the output file is:

LATITUDE,LONGITUDE,POPULATION

Finally, I used the script *createPointFile.py* to export the population datapoints into a point file (.pnt), a format readable by STK. This will be discussed further in a later section.

Distant Neighbors: As detailed in the next section, the optimization problem can be solved by assembling a cost matrix which contains the "cost" of flying between each and every waypoint in the dataset. Here, "cost" refers to the number of people that would be impacted by sonic booms if a supersonic aircraft flew between the two waypoints.

The computational expense of assembling the cost matrix is proportional to the number of waypoints squared (N^2), and this can lead to very long runtimes for the Python script. To address this, I implemented a distant neighbors algorithm. In effect, this limits the number of waypoints that can be connected to each other, reducing the computational expense of the problem.

I coined the term distant neighbors as a spinoff on the popular nearest neighbor search. For those unfamiliar, a nearest neighbor search "locates the k-nearest neighbors or all neighbors within a specified distance to query data points" [18]. For my project, I needed "neighbors" that were at least X distance away from the waypoint at hand. Thus, my distant neighbor search has two variables:

k = # of distant neighbors
N = minimum separation distance for neighbors

Both k and X can be manipulated to change the computational expense of the problem. I found that k = 50 and X = 50 km is a good choice, i.e. locate the k = 50 distant neighbors that are at least X = 50 km away from the waypoint at hand. An example of this is shown in Figure 9.

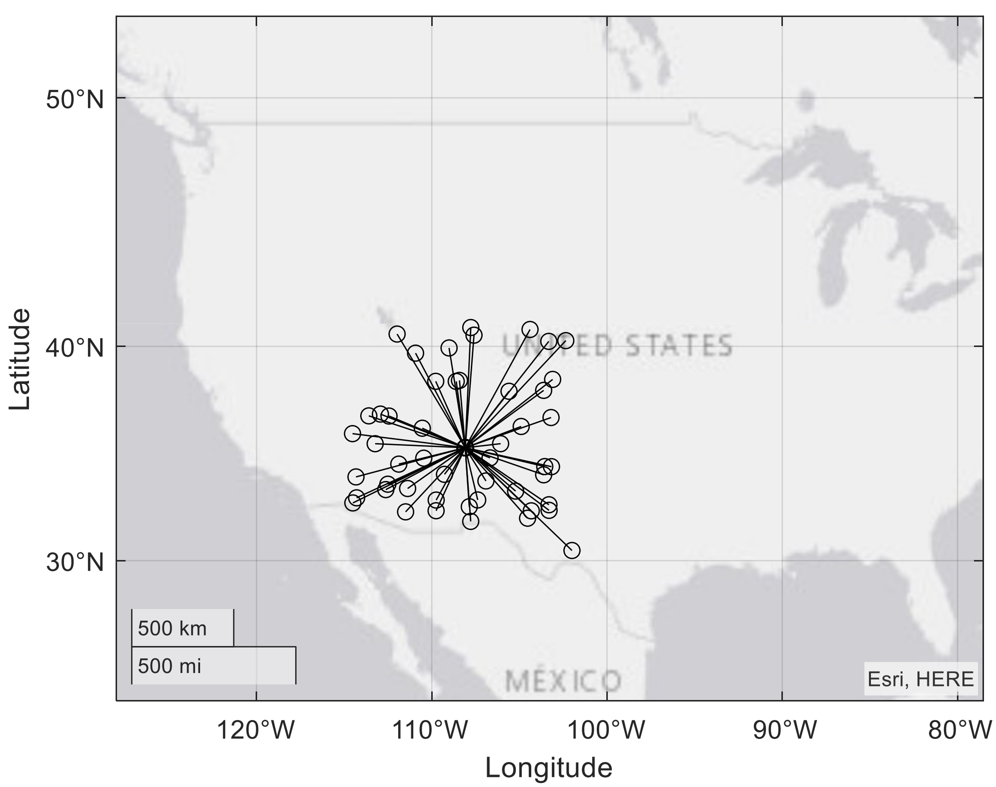


Figure 9: Distant Neighbors (k = 50, X = 50 km) for Waypoint ACELI in New Mexico, USA

My reason for using distant neighbors instead of nearest neighbors was to minimize the number of times the aircraft turns along the route. Using nearest neighbors, it would not be uncommon to have hundreds of waypoints on the LAX – JFK route because each waypoint is connected to its nearest neighbor (which may be just a few km away). A high number of waypoints along a route means that the aircraft is constantly turning to redirect itself to the next waypoint, and turning is problematic for two reasons. The first is that my simplified sonic boom model neglected the effects of aircraft turning; thus, each turn introduces a small amount of error in the route generation process. This is discussed in the limitations section at the end of the report. Second, excessive turning makes the overall route quite jagged, and this is undesirable from a pilot/passenger perspective. Smooth turns are relatively unnoticeable, but wide, jagged turns every few minutes would make the flight experience unenjoyable.

To emphasize the importance of using a distant neighbors algorithm, I have provided an image of an optimized route generated with nearest neighbors. It is shown below in Figure 10. The route has a total of 421 waypoints (yellow dots). This makes the route extremely jagged with lots of excessive turns. Thus, nearest neighbors should not be used to generate aircraft routes.

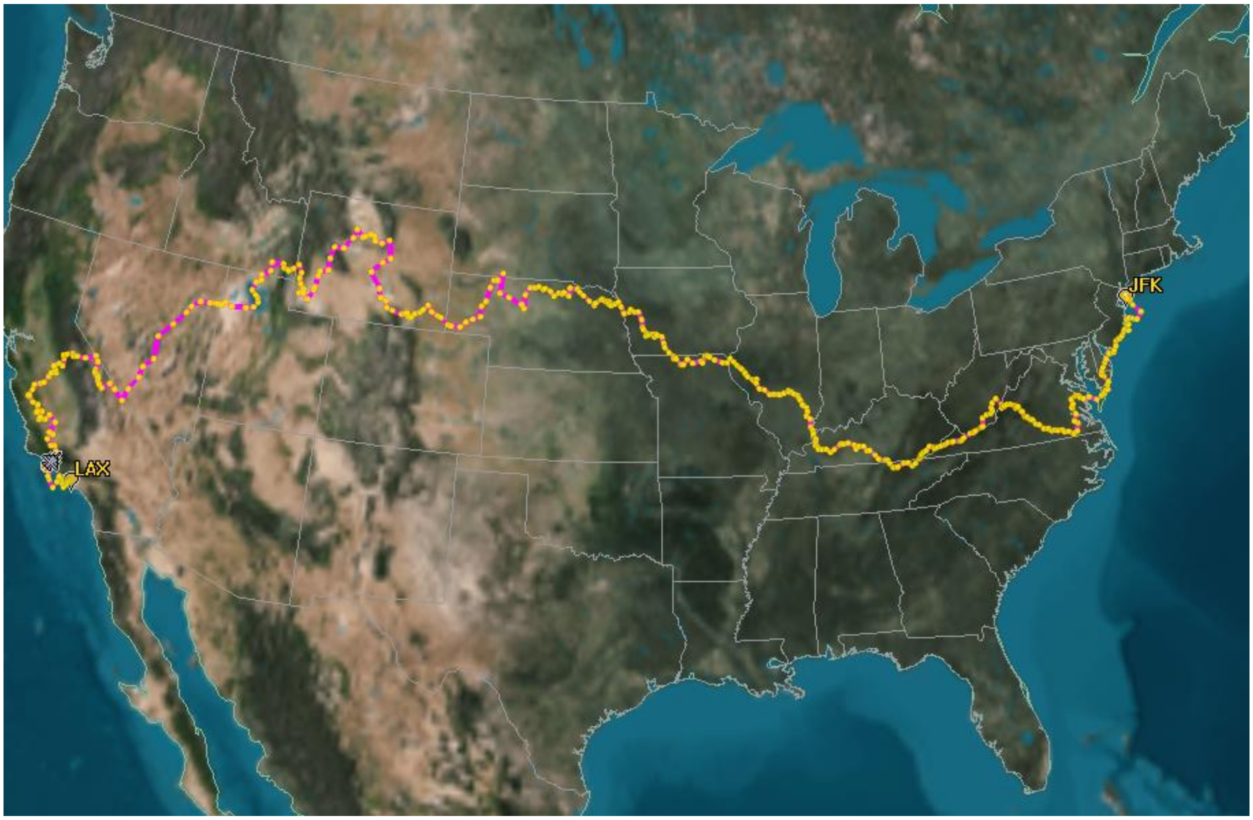


Figure 10: Nearest Neighbors Leads to Routes with Too Many Waypoints

Cost Matrix: As described previously, the cost matrix contains the "cost" of flying between waypoints in the dataset. It is an N x N matrix, with N being the number of waypoints in the dataset.

Before discussing the cost matrix, I will give an overview of how the "cost," or population impaction, is determined when flying between two waypoints. In the section on the model of the sonic boom carpet, it was shown that the primary carpet extends laterally about 25 nmi from each wing of the aircraft. Thus, a simple approximation of the primary carpet between two waypoints is a rectangle with a width of 50 nmi, with the waypoints located at the midpoints of the rectangle. This sounds simple enough, but a distance in nautical miles is not directly compatible with latitude/longitude coordinates.

A simple approximation is to convert the distance into a separation in degrees latitude/longitude. However, this approximation is far from perfect because the separation in degrees latitude/longitude is not constant across Earth. Specifically, the distance between degrees of longitude shrinks from ~69 miles at the equator to zero at the poles (where the lines of longitude converge) [3]. Fortunately, this is not a problem for degrees of latitude. Since degrees of latitude are parallel to each other, the distance between them remains approximately constant at ~69 miles. Thus, the approximation holds up well for perfectly westward/eastward flight. And because the flight from LAX to JFK is largely eastward, I felt the approximation was adequate. Using the haversine formula, it can be shown that in the westward/eastward case, 50 nmi is approximately 0.844° degrees of separation in latitude. This is shown below in Figure 11.

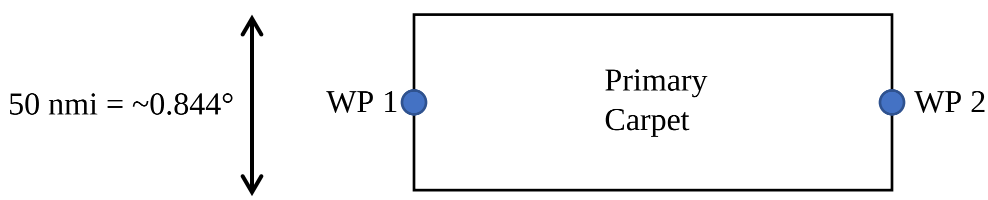


Figure 11: Primary Carpet Rectangle Approximation

Knowing the location of the two waypoints and the height of the rectangle, it is now possible to find the GPS coordinates of the corners of the rectangle. I accomplished this by writing a Python function named *recCorners()*. Please see the reference Python code for more information. With the GPS coordinates of the rectangle's corners in hand, the primary carpet rectangle is now well-defined. From here, I used the Matplotlib *contains_points()* function to determine which population datapoints lie in the primary carpet. Added together, this represents the number of people impacted by a supersonic aircraft flying between two waypoints (the "cost"). It should be emphasized that this is the most computationally expensive problem in the entire project. The *contains_points()* function must determine which population datapoints, out of 200,000+, lie in the primary carpet rectangle, for each waypoint and all of its distant neighbors.

An example of this technique is shown on the following page in Figure 12. There are two waypoints located at the midpoints of the primary carpet rectangle. The green dots represent population centers that will be impacted by sonic booms, while the red dots are unimpacted. A closer look shows Lake Michigan and the states of Wisconsin and Michigan. It should be noted that Canadian data was purposely excluded from the figure to make the geographic locations more noticeable. They are included in the actual calculations.

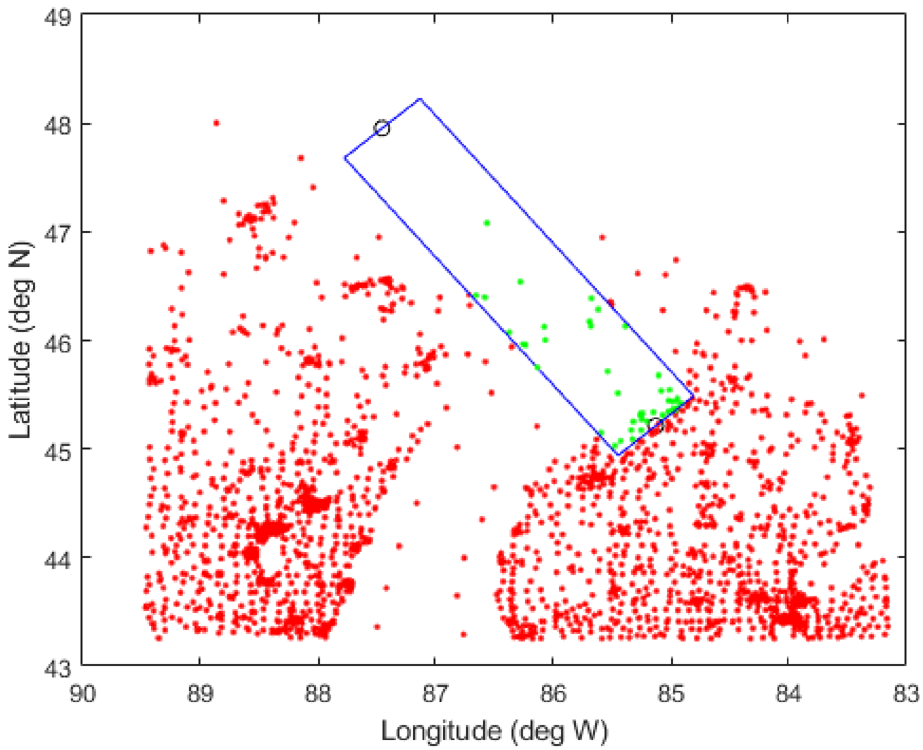


Figure 12: Estimating the Cost of Flying Between Two Waypoints

This function will then iterate over all of the waypoints and their N distant neighbors. The population impaction ("cost") will then be stored in the cost matrix. Since the code is only iterating over the N distant neighbors for each waypoint, many entries of the cost matrix will be left empty. The basic premise of the cost matrix is shown below in Table 3.

Table 3: Cost Matrix Design

	WP 1	WP 2	WP 3	WP 4	…	WP N
WP 1		104,228		629,623	…	
WP 2	104,228		5,735,251		…	
WP 3		5,735,251			…	
WP 4	629,623				…	
…	…	…	…	…	…	
WP N						

Color Coding Legend:

(gray)	Matrix Diagonal – WP to Itself – N/A
(white)	Distant Neighbors – Calculated
(orange)	Non-Distant Neighbors – Not Calculated

There are a few unique features of the cost matrix. The first is that the diagonal of the matrix is empty (the gray entries). This is because flying between a waypoint and itself is not realistic. The second feature to note is the orange entries. These are waypoints that are not considered distant neighbors; thus, the cost was not calculated in order to reduce the computational expense. Finally, the matrix is symmetric because the direction in which a supersonic aircraft travels between a pair of waypoints does not change the number of people impacted.

Dijkstra's Algorithm: Dijkstra's algorithm is a famous algorithm that is used to find the shortest path between two points given a map of points. For my purposes, I was concerned with finding the path with the lowest cost (population impaction). To do this, I simply replaced any reference of distance with values from the cost matrix. The rest of the algorithm is basically unchanged.

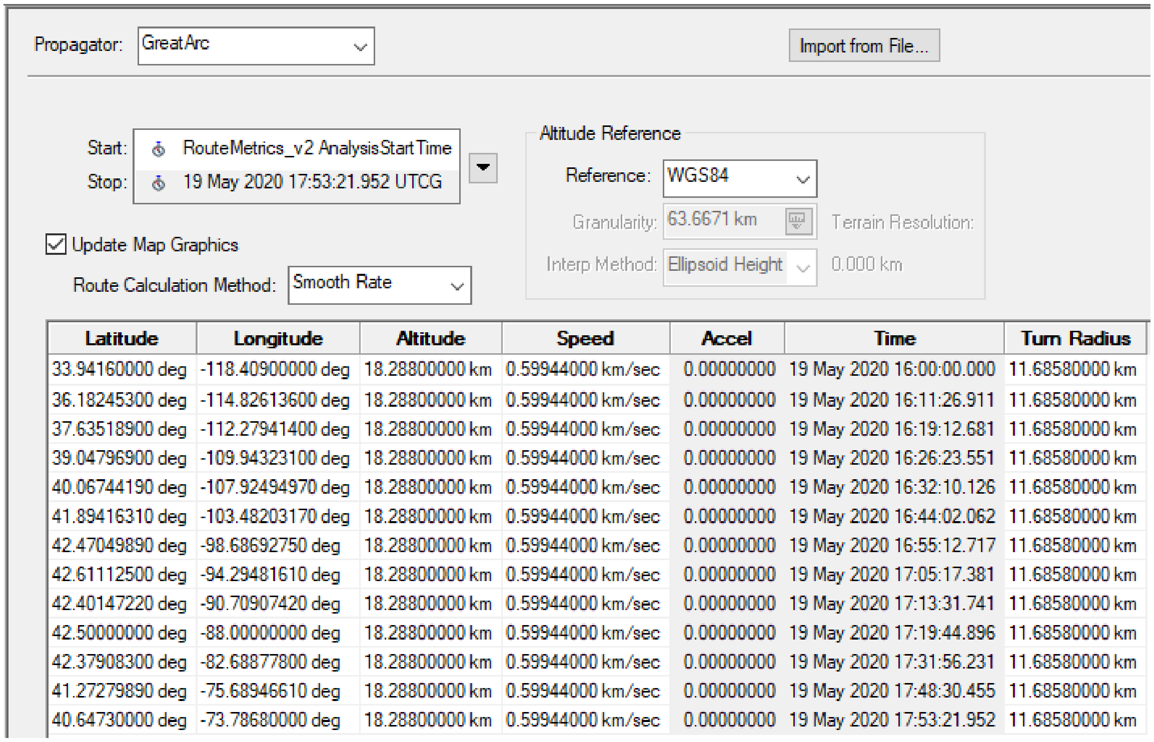
I will not spend much time discussing the algorithm itself, as it is well-documented online and not the focus of this paper. However, Dijkstra's algorithm does require an initial and final point from which to search for an optimal route. The trivial solution would be to provide LAX and JFK as the initial and final points, respectively. However, both of these points are located in major metropolitan areas where the Concorde would likely have to fly subsonically. Thus, it makes more sense to set the initial and final points somewhere outside the metropolitan areas, where the Concorde would actually be flying supersonically. The question of where is up to debate, but I found that placing the start point in the desert just east of LAX and the end point over the Atlantic Ocean just south of JFK as reasonable choices. To fly between these initial and final points to the airports, the Concorde is assumed to decelerate to a subsonic speed where it will not produce sonic booms.

Export Route into STK-Readable Format: After the Dijkstra algorithm generates the route, it must be converted into a format that is readable by STK: the great arc propagator file (.pg). This filetype always contains a header with STK-specific information, and the remaining portion of the file contains the latitude, longitude, altitude, and speed of each waypoint. I wrote a python script *greatArcProp.py* to create this file. More info about this specific filetype can be found on the *help.agi.com* website [12].

AGI's Systems Tool Kit (STK)

Systems Tool Kit (STK) is a software created by Analytical Graphics, Inc. (AGI). It is used to model complex systems (typically aerospace systems) and also offers great visualization tools. For my capstone, I used STK for modeling the Concorde aircraft as it flies from LAX to JFK. The software allowed me to quantify several metrics for the route, including the population impacted by sonic booms as well as the flight time.

Scenario Overview: Modeling a supersonic aircraft route in STK is quite simple. First, I created an aircraft object and gave it some waypoints. An easy way to load a large set of waypoints is to generate a great arc propagator file (.pg) in Python. This was described in the previous section. After setting the waypoints, the altitude and airspeed of the aircraft should be configured. Since I am modeling the Concorde, the altitude was set to 18.29 km (60,000 ft), and the airspeed was set to 0.59944 km/s (~Mach 2.02). This configuration is shown on the following page in Figure 13.



Latitude	Longitude	Altitude	Speed	Accel	Time	Turn Radius
33.94160000 deg	-118.40900000 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:00:00.000	11.68580000 km
36.18245300 deg	-114.82613600 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:11:26.911	11.68580000 km
37.63518900 deg	-112.27941400 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:19:12.681	11.68580000 km
39.04796900 deg	-109.94323100 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:26:23.551	11.68580000 km
40.06744190 deg	-107.92494970 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:32:10.126	11.68580000 km
41.89416310 deg	-103.48203170 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:44:02.062	11.68580000 km
42.47049890 deg	-98.68692750 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 16:55:12.717	11.68580000 km
42.61112500 deg	-94.29481610 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:05:17.381	11.68580000 km
42.40147220 deg	-90.70907420 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:13:31.741	11.68580000 km
42.50000000 deg	-88.00000000 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:19:44.896	11.68580000 km
42.37908300 deg	-82.68877800 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:31:56.231	11.68580000 km
41.27279890 deg	-75.68946610 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:48:30.455	11.68580000 km
40.64730000 deg	-73.78680000 deg	18.28800000 km	0.59944000 km/sec	0.00000000	19 May 2020 17:53:21.952	11.68580000 km

Figure 13: Aircraft Route Properties in STK

Next, a sensor object is attached to the aircraft. The sensor's geometry will be modified such that its projection on the surface matches that of the sonic boom primary carpet. For more information, please see the section on the model of the sonic boom carpet.

I created a coverage definition to model the population datapoints. I recommend using a global grid area of interest and a 0.01 lat/lon point granularity. A finer granularity will better detect population centers at the edge of the primary carpet, but it will also slow down the STK computation time (perhaps significantly). For that reason, I found 0.01 to be a good middle ground. The point locations should be set to "Custom Locations." This allows the user to import the point file (.pnt) with the population datapoints, as described in the data consolidation and formatting section. I then navigated to the "Assets" tab and assigned the primary carpet sensor object to the coverage definition.

Finally, a figure of merit should be attached to the coverage definition. At this point, the STK scenario is basically complete. The user can then right click on the coverage definition and compute accesses. When the accesses have finished computing, right click on the figure of merit and click on the "Report and Graph Manager." The user can then generate a "Value by Grid Point" report to determine which population datapoints were impacted by the sonic booms. After this report generates, save it as a (.csv) file. The Python script *impactionMetric.py* can then analyze the file and sum up the impacted population datapoints.

Route Metrics: The flight time can be determined by looking at the simulation time for the first and final waypoints. Figure 13 on the previous page shows a route that takes approximately 1 hour and 53 minutes.

To determine the population impaction, the python script *impactionMetric.py* analyzes the (.csv) file generated from the coverage definition in STK. The script itself is quite simple; it takes the sum of all the population datapoints that lay in the primary carpet. It prints the result in the command line.

RESULTS

In presenting my results for the baseline and optimal routes, it should be emphasized that the flight times do not account for taxi, takeoff, or landing. Instead, they are calculated as the time it takes to fly from the origin to the destination at a constant cruise altitude. Thus, the flight times are likely shorter than they would be in reality. However, this does not hinder any comparisons of the individual flight times because they are all shortened by a similar amount of time.

Subsonic Baseline Route

The subsonic baseline route is what a flight from LAX – JFK looks like today. Obviously, there can be differences due to routing around weather as well as aircraft substitutions, but the general flight characteristics remain the same. I modeled the subsonic baseline route with a Boeing 767-300 cruising at 33,000 ft at 561 mph (10.06 km at 903 km/hr).

The subsonic baseline route is shown below in Figure 14. It consists of 11 intermediate waypoints which are shown as yellow X’s. As mentioned earlier in the report, this route was found using the *flightaware.com* website.

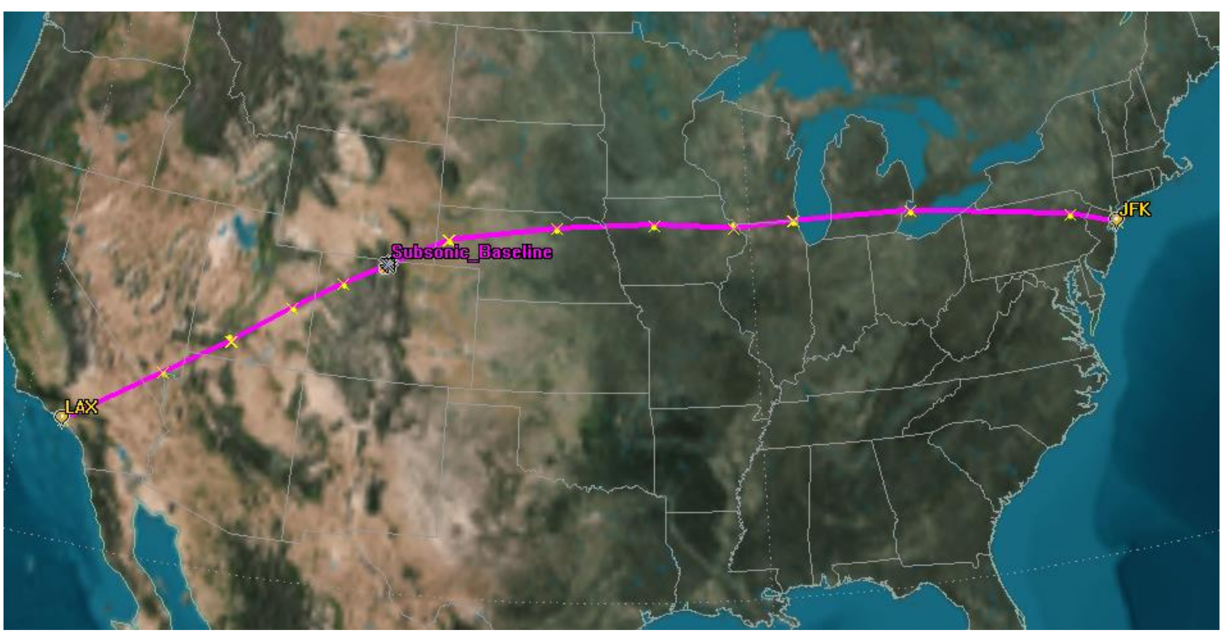


Figure 14: Subsonic Baseline Route

STK estimated the flight time of the subsonic baseline as 4 hours and 31 minutes. This is in the range of flight times reported by FlightAware, but it is on the shorter side due to the flight being modeled at cruise conditions for the entire duration. The population impaction for the subsonic baseline is zero because the route is entirely subsonic.

Supersonic Baseline Route

The supersonic baseline route is the same route as the subsonic baseline. The only difference is that the Boeing 767-300 has been swapped out for the Concorde which cruises at 60,000 ft at 1341 mph (18.29 km at 2,158 km/hr). STK estimated the flight time of the supersonic baseline as 1 hour and 53 minutes. 43,522,021 people are impacted by sonic booms. The impacted population centers (green dots) are shown below in Figure 15.

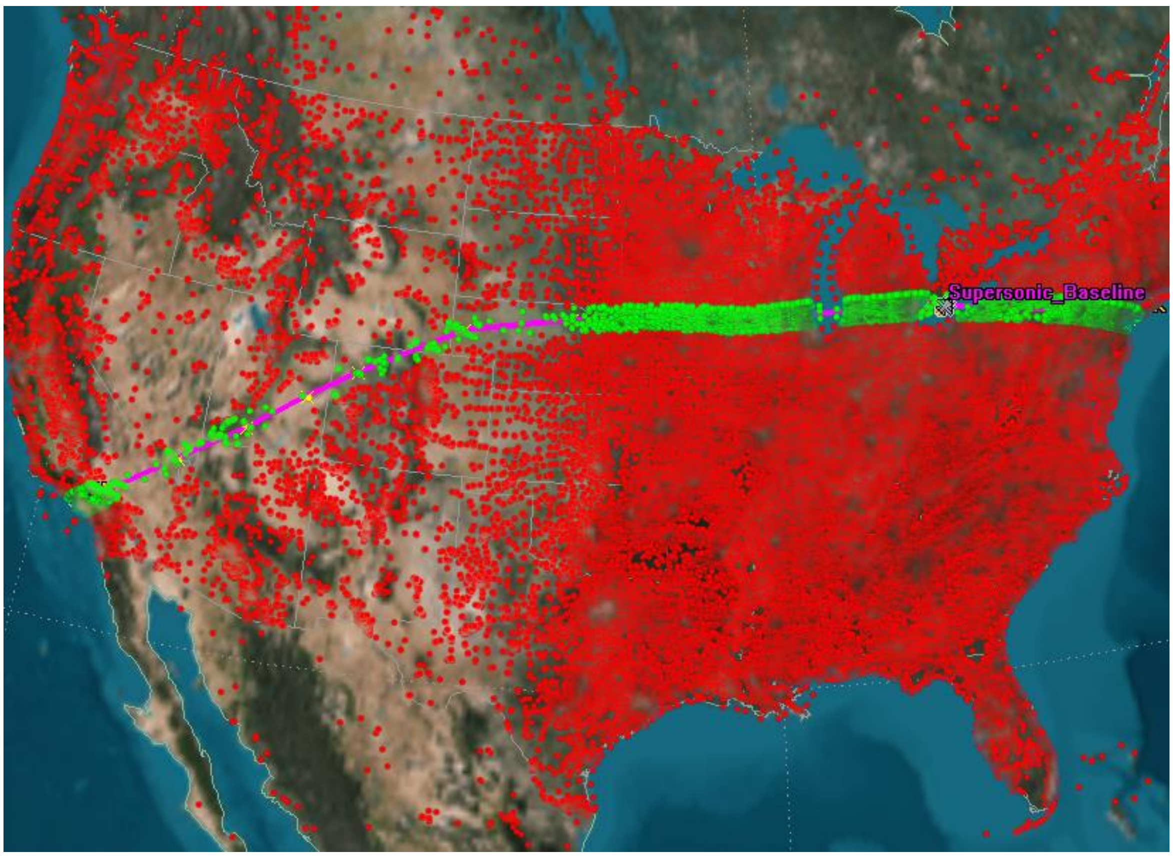


Figure 15: Supersonic Baseline – Population Impaction by Sonic Booms

It is evident that the population impaction is quite large. However, the LA and NYC population centers make up a large portion of the impaction metric. This is quite unrealistic because the Concorde would have to takeoff and land subsonically. Thus, adding speed restrictions around LAX and JFK would make the supersonic baseline more realistic. These speed restrictions also have the added benefit of limiting sonic booms over some of the largest population density regions in the country.

The supersonic baseline route with speed restrictions is shown below in Figure 16. The green segments represent the portion of the flight route where the aircraft travels subsonically.

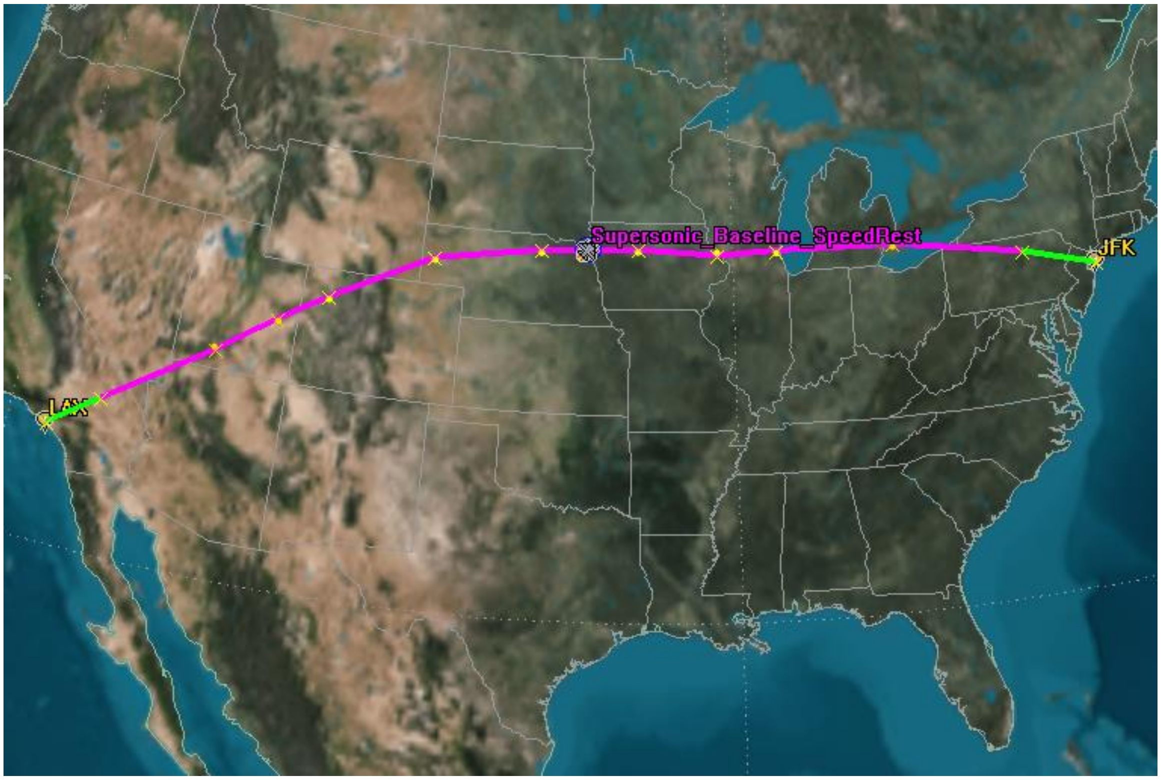


Figure 16: Supersonic Baseline with Speed Restrictions

With speed restrictions in place, the population impacted by sonic booms drops to 13,291,344. The flight time also increases slightly to 2 hours and 6 minutes. This is a substantial reduction in population impaction for only 13 minutes of extra flight time. However, the overall population impaction is still quite high, highlighting the need for optimization of the flight route itself.

Optimal Route

Having discussed the subsonic and supersonic baselines, I can now present the optimized routes that were generated with Dijkstra's algorithm. The route with the best results (before any post-processing) is shown on the following page in Figure 17. This route was generated with (k = 50 and N = 50 km) distant neighbors.

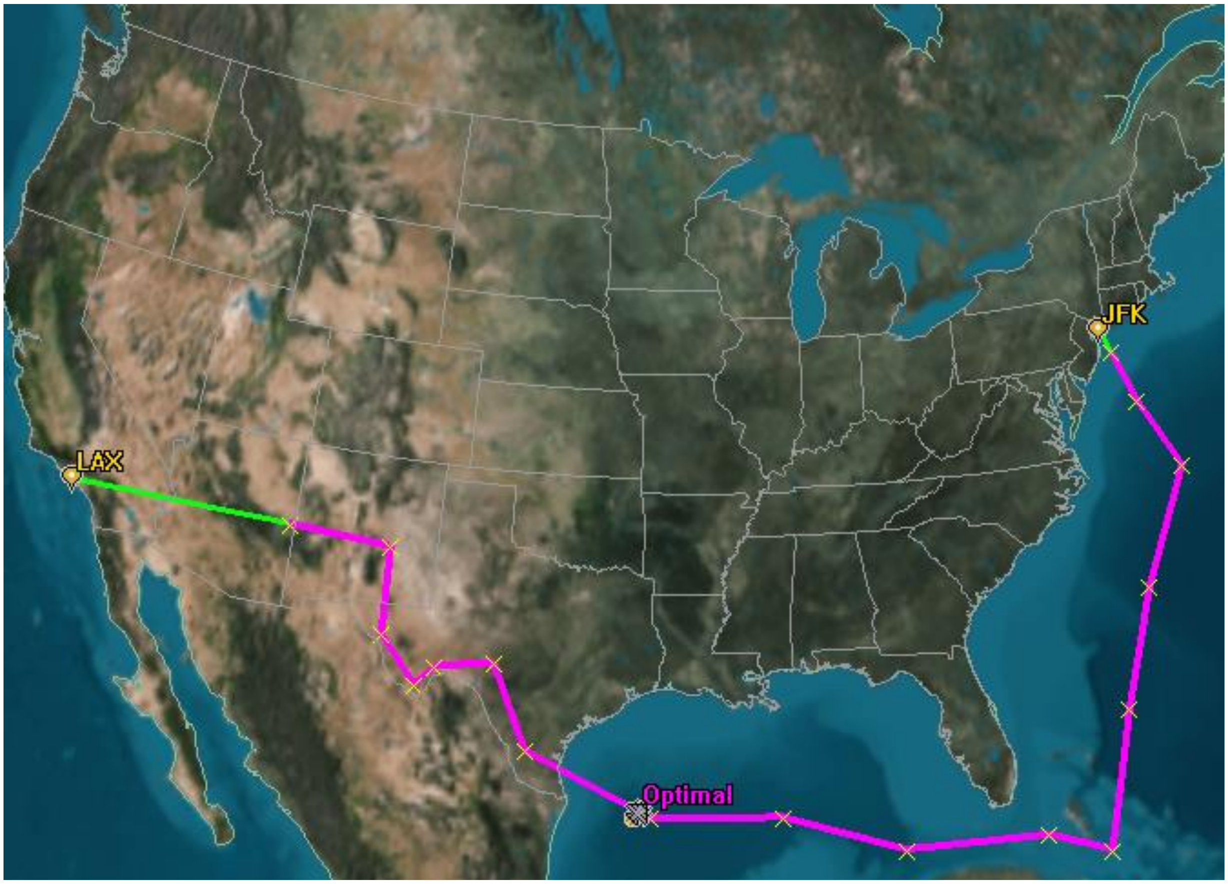


Figure 17: Optimal Route Generated by Dijkstra Algorithm

To help minimize the number of people impacted by sonic booms over large population centers, speed restrictions were introduced over the initial and final legs of the route. These are shown as green segments. The magenta portion of the route is where the Concorde flies supersonically. This route has a flight time of 3 hours and 34 minutes, and only 246,332 people are impacted by sonic booms. While this is a vast improvement in terms of population impaction (10,000,000+ down to ~250,000), the flight time is only a one-hour improvement over the subsonic baseline.

Optimal Route with Post-Processing

While the optimal route has a low population impaction, I believed that the flight time could be improved further. Specifically, a lot of time is "wasted" when the aircraft flies around the Florida peninsula. My initial thought was to try crossing the Florida peninsula subsonically, as this would offer a more direct route. There is some historical precedent for a maneuver like this. When the Concorde flew to Mexico City from either New York or Washington, the route "included a deceleration, from Mach 2.02 to Mach 0.95, to cross Florida subsonically and avoid unlawfully creating a sonic boom over the state" [7, p. 509].

With some manual tweaking of the waypoints, I was able to develop the route shown on the following page in Figure 18. Note that a green segment has been introduced across the Florida peninsula. This is one of three segments where the Concorde is “forced” to travel subsonically to avoid producing sonic booms over high-population areas. Overall, the route is much more direct.

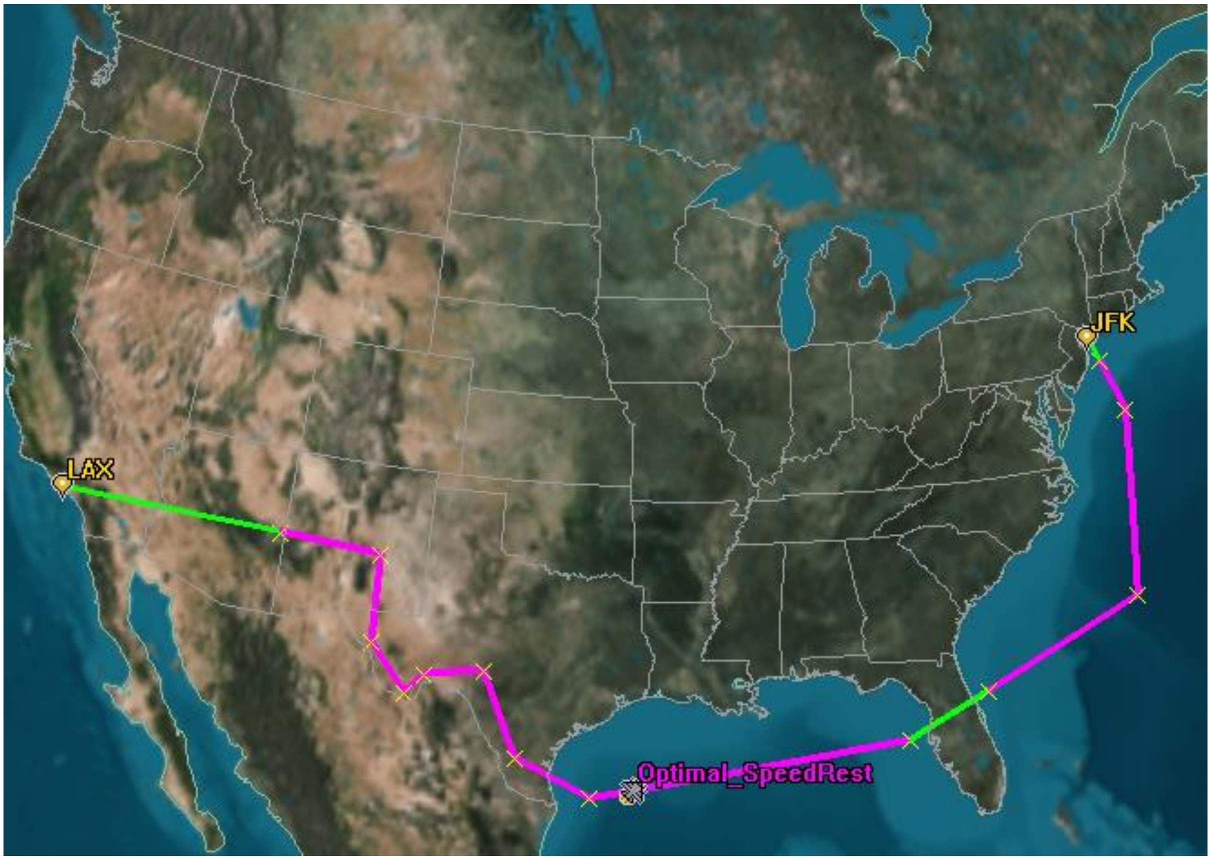


Figure 18: Optimal Route with Post-Processing

With the small amount of post-processing, the flight time drops to 3 hours and 20 minutes, a 14 minute reduction. Also, as expected, the population impaction stays the same. While the flight time improvement seems small, a 14 minute reduction does add up over time if the route were flown once a day for many years.

Semi-Optimal Route – Honorable Mention

I would like to give an honorable mention to one more route, which I am dubbing the semi-optimal route. It was generated very early-on in my research when I only had U.S. population data. Because of this, the Dijkstra algorithm found an “exploit” where it thought Canada was completely uninhabited. In a fantasy world where no-one lived in Canada, this route would actually be the optimal route. So why I am I presenting this route? After analyzing the route in STK with U.S. and Canadian population data, I found that the population impaction is not terrible. And interestingly, the flight time is even faster than the optimal route. It is for this reason that I am naming it the semi-optimal route. It offers a middle ground where slightly more people hear sonic booms, but it reaches JFK much quicker. The route is shown on the following page in Figure 19.

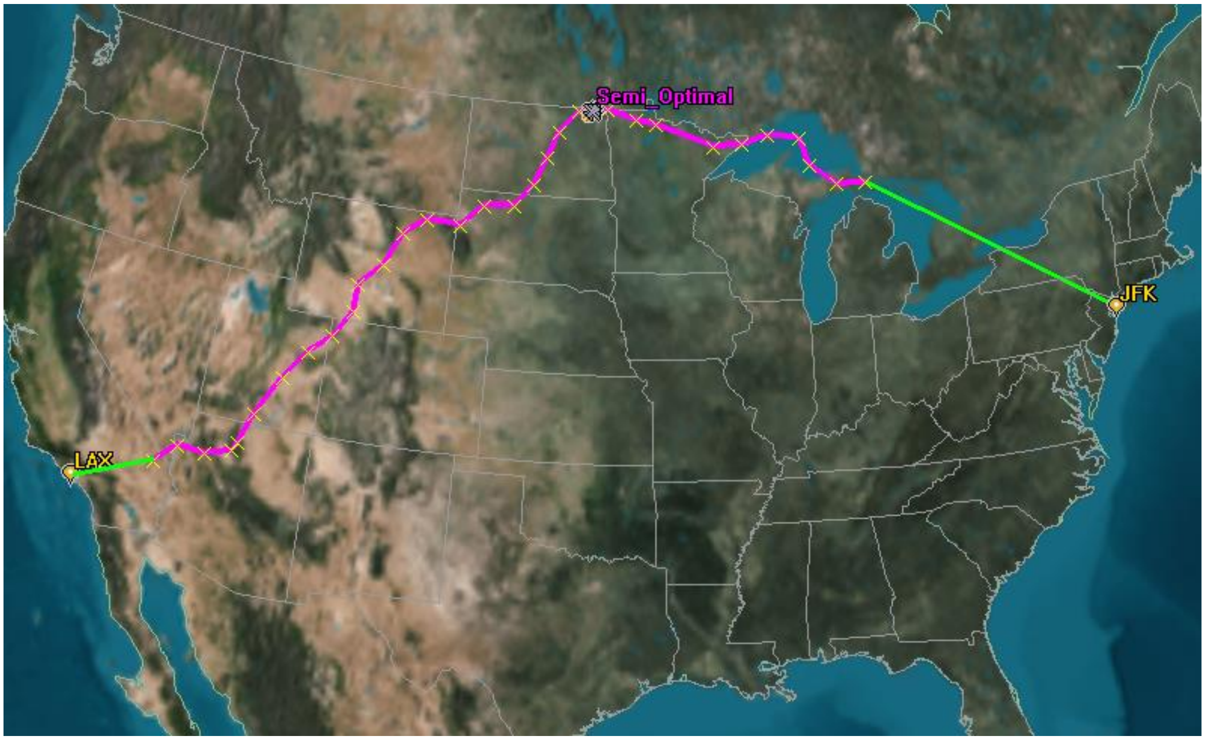


Figure 19: Semi-Optimal Route (Honorable Mention)

After analyzing this route in STK, I found that the flight time is an outstanding 2 hours and 49 minutes. However, 794,323 people are exposed to sonic booms. Thus, this route represents a middle ground between the two variables I wished to minimize. It is also unique in that it reaches JFK without flying over the Atlantic Ocean.

DISCUSSION OF RESULTS

I have now presented a total of six routes. There are three baseline cases: subsonic, supersonic, and supersonic with speed restrictions. There are also three optimized routes: optimal, optimal with post-processing, and semi-optimal. The performance metrics for these routes are shown below in Table 4.

Table 4: Baseline and Optimal Route Performance Metrics

Route	Flight Time	Population Impaction	Figure #
Subsonic Baseline	4 hr 31 min	0	14
Supersonic Baseline	1 hr 53 min	43,522,021	15
Supersonic Baseline w/ Post-Processing	2 hr 06 min	13,291,344	16
Optimal Route	3 hr 34 min	246,332	17
Optimal Route w/ Post-Processing	3 hr 20 min	246,332	18
Semi-Optimal Route	2 hr 49 min	794,323	19

The best route in terms of population impaction is the optimal route with post-processing, shown in Figure X. It has a flight time of 3 hours and 20 minutes, and 246,332 people are exposed to sonic booms. This is a flight time improvement of 1 hour and 11 minutes from the subsonic baseline, which is significant. However, the number of people impacted by sonic booms is likely too high for the FAA to consider repealing the ban on supersonic travel over the U.S.

One way to analyze these routes as a whole is to plot their population impaction as a function of flight time. This is shown below in Figure 20.

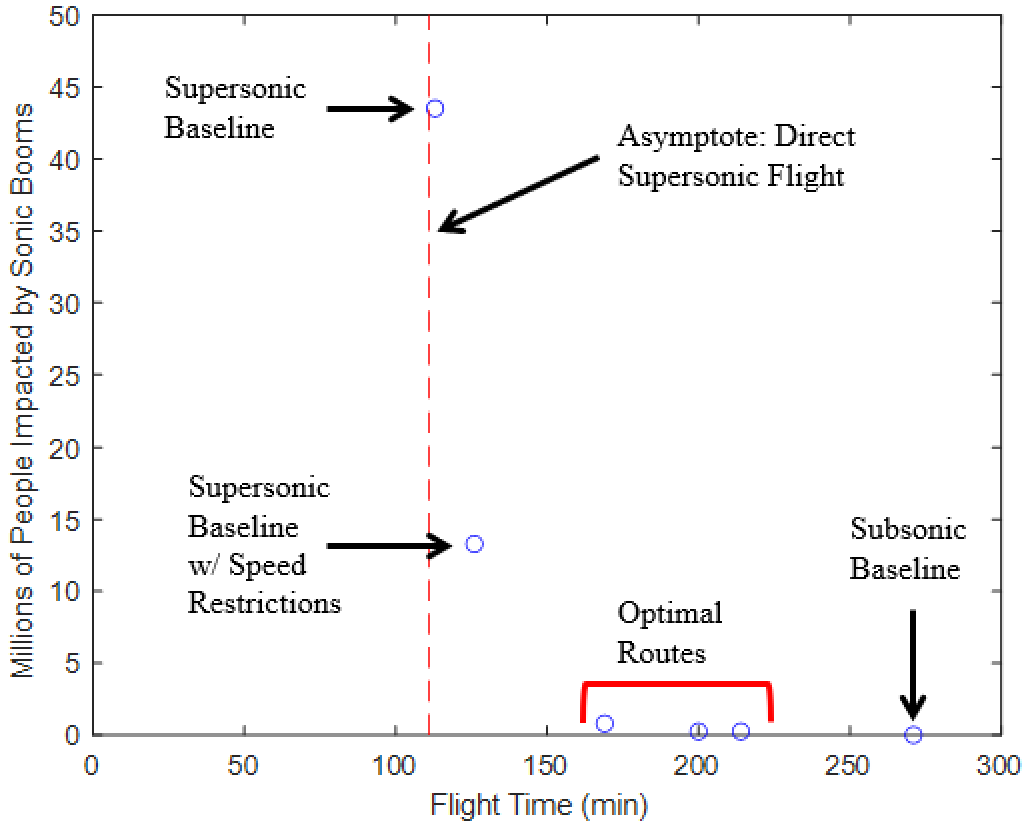


Figure 20: Relationship Between Population Impaction and Flight Time

While this plot only has six datapoints, important conclusions can still be drawn. The first is the presence of an asymptote (the dashed red line). At a flight time of approximately two hours (120 min), the population impaction starts to increase dramatically. This makes a lot of sense because, at some point, the aircraft will be flying as directly as possible towards JFK. A direct route shows no regard towards avoiding population centers; instead, it is entirely focused on decreasing the flight time. Once a perfectly direct route is established, it is impossible to decrease the flight time any further unless the aircraft itself travels faster. And, in a project where the cruise speed of the aircraft has a fixed maximum, the flight time must approach an asymptote.

Additional conclusions can be drawn from the four datapoints near the bottom of the plot: the three optimal routes and the subsonic baseline. Figure 21, shown below, is the same plot as Figure 20, just zoomed in on these four points.

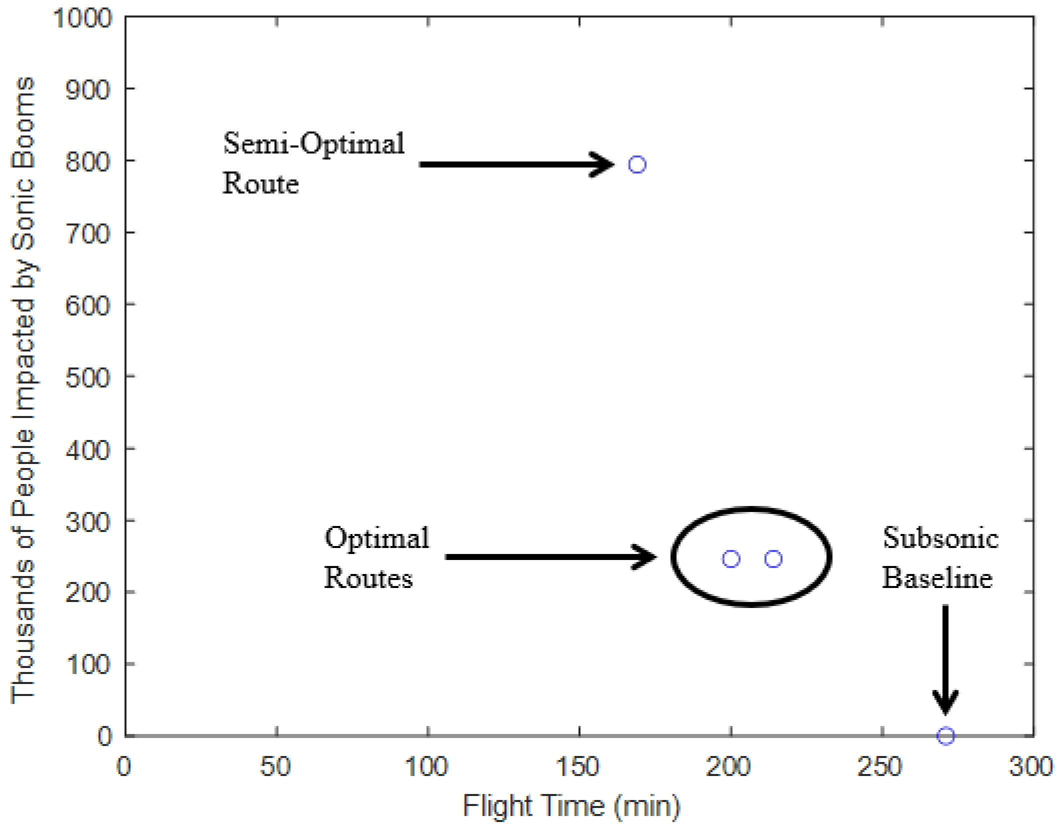


Figure 21: Routes with Population Impaction Less Than 1,000,000

It is extremely likely that other optimized routes exist between LAX and JFK, and this plot alludes to where they may exist in the flight time vs. population impaction domain. Specifically, for optimized routes with a population impaction less than one million, it is likely that the flight time will be somewhere between 150 - 250 minutes. In future studies, it would not be surprising to find other optimal routes in this vicinity.

LIMITATIONS

In this section, I will discuss some of the limitations of this research capstone. Should future work be done on this optimization problem (or a similar one), these limitations would likely need to be addressed.

Primary Carpet Approximation During Cost Matrix Construction

As I noted in the section on the cost matrix, converting the primary carpet's width of 25 nmi into latitude/longitude coordinates is not as simple as it may seem. Because the distance between lines of longitude is not constant with changing latitude, the approximation of 25 nmi = 0.844° only holds when flying directly eastward/westward.

It should also be noted that another error occurs when the waypoints are not colinear. Since I modeled the primary carpet as a rectangle between two waypoints, the rectangle fails to capture the full extent of the actual primary carpet during turns. This is illustrated in Figure 22 below.

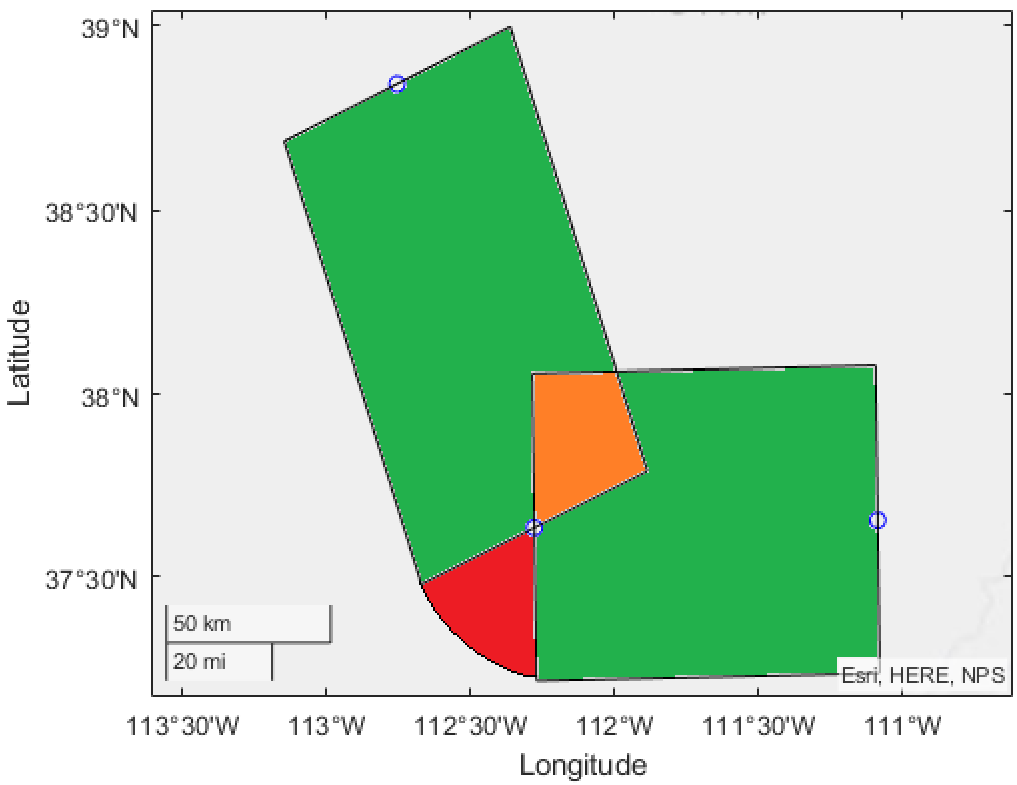


Figure 22: Non-Colinear Waypoints Create Errors in Primary Carpet Approximation

Any population centers in the green regions will be counted correctly. In the orange region, the sonic boom impaction area was double counted (i.e. while flying from WP1 to WP2 it counted it once, it then counted it again from WP2 to WP3). Population centers in the red region were not counted at all. Since the primary carpet is modeled as rectangles, it fails to account for the curvature introduced when the aircraft turns.

While these errors may seem a bit egregious, it is important to note that they only exist in the construction of the cost matrix. This means that the errors only affect the route generation process. When verifying the route in STK, this primary carpet approximation is not used in any capacity. Instead, a sensor object attached to the aircraft measures the population impaction to a very high degree of accuracy. In other words, while the route itself is generated based on somewhat poor assumptions, it can still be verified with a high fidelity to ensure it is an improvement over the baseline.

Modeling the Sonic Boom Carpet During Aircraft Turns

In the section on distant neighbors, I noted that routes with a high number of waypoints (and therefore aircraft turns) were undesirable. This is because my simplified model of the sonic boom primary carpet does not take turns into account. Thus, each individual turn along the route will introduce a bit of error.

The U.S. Air Force has researched the effect of aircraft turning on the sonic boom carpet. In one study, they found that a turn causes sonic booms to become focused in some regions, and "the measured focus boom from the turn amplified the boom by a factor of 2.5 . . . [but] the superfocus region on the ground was very small in area and challenging to capture" [16, pgs. 76-77]. In other words, while turning can cause sonic booms to become louder and perhaps more damaging, the region where this amplification occurs is so small that it is difficult to predict its location. For this reason, I did not make any changes to my primary carpet model during aircraft turns. In a more advanced study, the primary carpet could be modeled in different flight modes, such as turns.

Advanced Optimization Techniques

The optimization algorithm (Dijkstra) is quite simple in that it searches for waypoint connections that will expose the fewest number of people to sonic booms. There are a few ways in which this algorithm could be improved in future studies.

First, the optimization algorithm could modify the flight speed of the aircraft itself. In other words, the algorithm would be able to decide whether to fly between two waypoints subsonically or supersonically. This would be beneficial in regions with "unavoidable" population centers, such as the areas surrounding LAX and JFK. The algorithm could also modify the aircraft's altitude. Theoretically, the width of the primary carpet changes with altitude. This could be beneficial in situations with two cities located close together. Instead of flying around them, the aircraft could adjust its altitude until the primary carpet's width is small enough such that it is safe to fly between the cities.

Both of these techniques could be used to develop more advanced routes. However, it is important to note that speed and altitude changes may reduce the accuracy of the primary carpet model itself. This is because any deviation from steady flight can lead to sonic boom amplification.

CONCLUSION

This research capstone demonstrated that the routes on which supersonic aircraft fly can be optimized on the basis of the number of people exposed to sonic booms. Furthermore, these optimized routes still achieve flight times that are shorter than their subsonic counterparts, despite the elongation of the routes themselves (to avoid population centers). Despite these optimistic findings, there are still several hurdles to overcome, including the critical fact that the best-case optimal route still exposes ~250,000 people to sonic booms. This is a figure that is unlikely to change any minds at the FAA, an organization which placed a ban on supersonic travel over the United States in 1973. If commercial supersonic aircraft are to ever fly over the United States, it will likely take a joint-effort between researchers quieting the sonic boom itself and those optimizing overland routes.

ACKNOWLEDGEMENTS

Amy Cohn, Industrial Operations and Engineering Professor, Capstone Advisor

I would like to extend sincere thanks to Professor Amy Cohn for her support and guidance on this capstone. This project was quite multi-disciplinary, with aerospace, optimization, and data science components. Optimization was definitely one of my "weak links," but Professor Cohn was able to provide some great advice on the optimization side. I cannot thank Professor Cohn enough for supporting this capstone over the past eleven months.

Rachel Armstrong-Ceron, College of Engineering Honors Academic Advisor

I would like to thank Rachel for her help in the early phases of my capstone project. I met with her during the Fall 2019 semester to discuss my project idea, and she really helped get the ball rolling. During this meeting, she referred me to Professor Cohn, who ended up becoming my capstone advisor. Rachel told me that Professor Cohn had experience in the aviation industry as well as tackling optimization problems. I also want to thank Rachel for advising me throughout my undergraduate career and providing feedback on my application to the Sequential Undergraduate Graduate Studies (SUGS) program.

REFERENCES

[1] "BAC Concorde – Luxury at twice as fast as the speed of sound." *BAE Systems*, published online 09 Apr. 2019.

[2] Barry, B. "How Concorde Pushed the Limits – Then Pushed Them Too Far." *National Geographic*, published online 05 Sep. 2019.

[3] Briney, A. "Longitude." *ThoughtCo*, published online 27 Aug. 2020.

[4] "Census Profile, 2016 Census – Population Centres (POPCTRs)." *Statistics Canada*. Accessed Sep. 2020.
https://www12.statcan.gc.ca/

[5] "Centers of Population for the 2010 Census – Centers of Population by Block Group." *U.S. Census Bureau*. Accessed Feb. 2020.
https://www.census.gov/geographies/reference-files/2010/geo/2010-centers-population.html

[6] Code of Federal Regulations, Title 14, Vol. 2, Part 9 – General Operating and Flight Rules, Subpart 1 – Operating Noise Limits, Sec. 91.817, Civil Aircraft Sonic Boom, 1973.

[7] Cramoisi, G. "Air Crash Investigations – The End of the Concorde Era – The Crash of Air France Flight 4590." *Mabuhay Publishing*. 2011.

[8] "Geocoding API." *Google*. Accessed Sep. 2020.
https://developers.google.com/maps/documentation/geocoding/overview

[9] Gibbs, Y. "NASA Armstrong Fact Sheet: Sonic Booms." *NASA*, published online 15 Aug. 2017.

[10] Gipson, L. "NASA's Supersonic X-59 QueSST Coming Together at Famed Factory." *NASA*, published online 09 Oct. 2019.

[11] "GPX Aviation Waypoint Generator." *navaid.com*. Accessed Jul. 2020.
http://navaid.com/GPX/

[12] "Great Arc Propagator File Format (*.pg or *.ga)." *AGI*, published online Nov. 2020.
https://help.agi.com/stk/index.htm#stk/importfiles-05.htm

[13] "IFR Route Analyzer." *FlightAware, LLC*.
https://flightaware.com/statistics/ifr-route/

[14] Jenkinson, L, Simpkin, P, Rhodes D. "Civil Jet Aircraft Design – Aircraft Data File." *Elsevier*, published 1999.

[15] Linden, R. van der. “My Ride on the Concorde.” *Air and Space Magazine*, published online Mar. 2004.

[16] Maglieri, D, Bobbitt, P, Plotkin, K., Shepherd, K, Coen, P, Richwine, D. “Six Decades of Sonic Boom Research.” *NASA Langley Research Center*. 2014.

[17] “MATLAB r2019b.” *MathWorks*.
https://www.mathworks.com/products/matlab.html

[18] “Nearest Neighbors.” *MathWorks*.
https://www.mathworks.com/help/stats/nearest-neighbors-1.html

[19] Price, H. J. “FAA – Press Release – March 30, 2020 – Fact Sheet – Supersonic Flight.” *FAA*, published online 30 Mar. 2020.

[20] “Python Language Reference, version 3.7.3.” *Python Software Foundation*.
http://www.python.org

[21] “Systems Tool Kit v11.7.1.” *Analytical Graphics Inc*.
https://www.agi.com/products/stk

[22] “Type-Certificate Data Sheet No. EASA.IM.A.115 for Boeing 787.” *EASA*, published online 16 Nov. 2017.

[23] “World Population Review.” Accessed Sep. 2020.
https://worldpopulationreview.com/