

Modeling and Simulation of Traffic on I-485 via Linear Systems and Iterative Methods

Dominic Kealoha, Fabiola Rojas, Xingjie Li

Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC

Students: dkealoha@charlotte.edu, frojas4@charlotte.edu

Mentor: xli47@charlotte.edu

ABSTRACT

Iterative methods such as Jacobi, Gauss-Seidel, and Successive Over-Relaxation (SOR) are fundamental tools in solving large systems of linear equations across various scientific fields, particularly in the field of data science which has become increasingly relevant in the past decade. Iterative methods' use of matrix multiplication rather than matrix inverses makes them ideal for solving large systems quickly. Our research explores the factors of each method that define their respective strengths, limitations, and convergence behaviors to understand how these methods address drawbacks encountered when performing matrix operations by hand, as well as how they can be used in real world applications. After implementing each method by hand to understand how the algorithms work, we developed a Python program that assesses a user-given matrix based on each method's specific convergence criteria. The program compares the spectral radii of all three methods and chooses to execute whichever will yield the fastest convergence rate. Our research revealed the importance of mathematical modeling and understanding specific properties of the coefficient matrix. We observed that Gauss-Seidel is usually the most efficient method because it is faster than Jacobi and doesn't have as strict requirements as SOR, however SOR is ideal in terms of computation speed. We applied the knowledge we gained to create a traffic flow model of the I-485 highway in Charlotte. After creating a program that generates the matrix for this model, we were able to iteratively approximate the flow of cars through neighboring exits using data from the N.C. Department of Transportation. This information identifies which areas are the most congested and can be used to inform future infrastructure development.

KEYWORDS

Iterative Methods; Convergence; Spectral Radius; Traffic Flow Network

1 INTRODUCTION

There are various ways to solve a system of linear equations, notably direct methods and iterative methods. During our ten weeks of undergraduate research, we closely examined three iterative methods intended for solving systems of linear equations, those being the Jacobi Method, Gauss-Seidel Method, and Successive Over-Relaxation Method. From our time studying these models we were able to determine their differences in general efficiency and when they might be most effectively applied. From this point we developed a program based in Python in which a user can input a given square matrix and desired accuracy and find its solution using the iterative method that will return quickest results. We developed a generalized process using flow networks for the translation of real world data into a matrix form that guarantees the effective use of the three iterative methods we studied. Using the average annual daily traffic (AADT) data values on each exit of the I-485 highway in the city of Charlotte, NC, we were able to successfully approximate the amount of traffic on every segment of road within the inner loop. We have opted to use iterative methods for this modeling because of their utilization of matrix multiplication which allows for significantly higher computational speed, making them ideal for exceptionally large data sets.

Charlotte ranks 35th in the nation for traffic congestion. In 2017, the average American commuter spent seven full working days in traffic over the span of the year, resulting in \$1,000 in personal costs⁷. These values are approximately seven years old and there are over 500,000 more commuters in the Charlotte Metropolitan area in 2024. The population of the Charlotte Metropolitan area now measures at 2,321,000 inhabitants according to the United Nations population projections and is expected to increase by almost 13% in the next decade⁸. The traffic issue will only worsen. Present infrastructure is not efficient enough to handle current residents, and we can only hope new construction to house and support future residents without significantly disrupting traffic patterns. Furthermore, given that Charlotte is the 6th least walkable city in the country, most residents are forced to drive to access most services⁹. It is important to analyze traffic patterns to address current issues and take action to prevent them from worsening. This may require a different approach to designing roadways.

The primary purpose of our mathematical modeling was to utilize learned concepts from our linear algebra textbooks in combinations with new concepts beyond its scope to formulate a tool in which we can study real world applications. We believe we have been able to provide some valuable insight on city infrastructure using these simple tools. Several models have been developed by scholars to investigate implications of traffic congestion with a rapidly growing population, including statistical models, travel time models¹⁰, and other complex mathematical models that rely on multiple variables¹¹. To the best of our knowledge, we are the first to apply a simple traffic flow linear system combined with iterative solvers to determine the AADT of I-485, and our findings have proven to be insightful. Our data highlights neglected infrastructure in regards to traffic flow in exits near the UNC Charlotte campus, indicating a rapidly increasing student population. This claim is supported by reduced AADT values in June and July where the majority of students return home for Summer break. The Charlotte Department of Transportation (CDOT) and North Carolina Department of Transportation (NCDOT), who in collaboration are responsible for all construction projects on I-485, have shown a focus on road construction in the southern regions of I-485. Our findings show that exits on I-485 near the UNC Charlotte campus had the highest AADT values, which we did not expect given the high density of Uptown as well as it being the nation's second largest banking center. These findings reflect the increasing population of UNC Charlotte as well as a need for more efficient infrastructure in the general area. The modeling of the average traffic along I-485 can play a significant role in determining which regions of I-485 are receiving less or more support in terms of infrastructure from the CDOT and NCDOT, and we offer potential solutions to improving traffic flow. The model we created is also flexible, and in the future we hope to address more issues in Charlotte infrastructure such as the effectiveness of Duke Energy's current power supply grid in Charlotte.

In the following we will provide an overview of our research and provide our findings. First, we provide an introduction and some important prerequisite information involving our general methodology. We also review the three iterative methods that we studied and observe their unique analytical properties. Next, we elaborate on our Python program and how we implemented the algorithms to promote efficiency, and discuss applying what we've learned to approximate traffic values on I-485 using iterative methods and discussing the AADT approximations. We also explain our method for making any similar matrix fit the convergence criteria for the three iterative methods. Finally, we have a general conclusion discussing our findings, the importance of their application, and possible applications that we would like to consider in the future.

2 METHOD

In this section, we will review three iterative methods for solving linear systems according to the tutorials^{2, 5, 6}, as well as the details on converting a traffic network to a linear system by following the textbook³.

2.1 *Linear Systems of Equations and Direct Methods*

Iterative methods have proven to be particularly effective in solving large systems of equations with impressive memory efficiency and solution accuracy. To help us understand why this is, the following is a brief general overview

of concepts involving linear systems of equations, and defining iterative methods as a whole while acknowledging their strengths and weaknesses.

Linear systems share the following general format where we define A as the coefficient matrix, \mathbf{x} as the vector of unknown variables that we intend to solve for, and \mathbf{b} as our vector of constants.

$$A\mathbf{x} = \mathbf{b}. \quad \text{Equation 1.}$$

This system of equations is able to be translated into a matrix representation. The general steps for this process involve pulling all of the equation coefficients into matrix A and a right hand side vector \mathbf{b} , and the unknown solution \mathbf{x} could be solved by sequentially eliminating unknown variables. For instance, the following system of equations:

$$\begin{aligned} 5x_1 - 2x_2 + 3x_3 &= -1, \\ -3x_1 + 9x_2 + 1x_3 &= 2, \\ -2x_1 - 1x_2 - 7x_3 &= 3 \end{aligned}$$

share a matrix representation of:

$$\begin{bmatrix} 5 & -2 & 3 \\ -3 & 9 & 1 \\ -2 & -1 & -7 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix}.$$

By Gaussian elimination, we convert the above system into

$$\begin{bmatrix} 5 & -2 & 3 \\ 0 & 39 & 14 \\ 0 & 0 & \frac{-335}{13} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 7 \\ \frac{190}{13} \end{bmatrix}$$

and hence we can get the solution.

To perform the direct method, the matrix A in **Equation 1.** needs to be invertible.

2.2 Iterative Methods

Other than the direct methods, we can also solve $A\mathbf{x} = \mathbf{b}$ by iterative methods. In this subsection, we review three common iterative methods for solving the linear system in **Equation 1.** The core of our research revolved around the study of iterative methods for their potential use in solving linear systems arising from various real world applications.

Let A_{ij} denote an entry in the matrix A at row i and column j and let k denote the current iteration number. We are able to approximate a solution by introducing an initial guess for the solution \mathbf{x} that we then repeatedly refine through multiple iterations using one of the iterative algorithms. For the sake of consistency, during our study we opted to always set our initial guess $\mathbf{x}^{(0)}$ to be a zero vector for more definitive results in comparing the performances of various algorithms. At the k -th iteration, the current iterative value is denoted by $\mathbf{x}^{(k)}$. The program we created utilizes the norm of the residual to track how close the iterations are getting to the true solution. Recall that for a linear system $A\mathbf{x} = \mathbf{b}$, the residual for the k -th iterative solution is defined as $\mathbf{r}^{(k)} := \mathbf{b} - A\mathbf{x}^{(k)}$. We know we are approaching the convergence point once our estimated quantity satisfies the pre-determined error threshold η , that is $\|\mathbf{r}^{(k)}\|_2 < \eta$. After each iteration the difference between \mathbf{b} and $A\mathbf{x}^{(k)}$ gets smaller and smaller, meaning the k -th approximation is getting closer and closer to the exact solution \mathbf{x} .

The Jacobi method. First, we studied the Jacobi iterative method. The algorithm is as follows:

$$\mathbf{x}_i^{(k)} = \frac{1}{A_{ii}} \left(\sum_{j=1, j \neq i}^n (-A_{ij} \mathbf{x}_j^{(k-1)}) + \mathbf{b}_i \right), \quad i = 1, 2, \dots, n. \quad \text{Equation 2.}$$

The Gauss-Seidel method. Next, we studied the Gauss-Seidel algorithm:

$$\mathbf{x}_i^{(k)} = \frac{1}{A_{ii}} \left(-\sum_{j=1}^{i-1} A_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n (A_{ij} \mathbf{x}_j^{(k-1)}) + \mathbf{b}_i \right), \quad i = 1, 2, \dots, n. \quad \text{Equation 3.}$$

The successive overrelaxation method (SOR). The SOR method is similar to the Gauss-Seidel method (GS) and the update of $\mathbf{x}^{(k)}$ can be viewed as the combination of its previous value and GS value through a new weight parameter ω :

$$\mathbf{x}_i^{(k)} = (1 - \omega) \mathbf{x}_i^{(k-1)} + \frac{\omega}{A_{ii}} \left(\mathbf{b}_i - \sum_{j=1}^{i-1} A_{ij} \mathbf{x}_j^{(k)} - \sum_{j=i+1}^n A_{ij} \mathbf{x}_j^{(k-1)} \right), \quad i = 1, 2, \dots, n. \quad \text{Equation 4.}$$

Suppose that $A = D - L - U$, where D is a diagonal matrix that contains the diagonal part of A , $-L$ and $-U$ are the lower and upper triangular parts of A , respectively. Each algorithm then can be translated into a simpler matrix form:

$$\mathbf{x}^{(k)} = T \mathbf{x}^{(k-1)} + \mathbf{c},$$

where T is the iterative matrix and \mathbf{c} is a constant vector:

- The Jacobi method:

$$T_j := D^{-1} (L + U), \quad \mathbf{c}_j := D^{-1} \mathbf{b}.$$

- The Gauss-Seidel method:

$$T_g := (D - L)^{-1} U, \quad \mathbf{c}_g := (D - L)^{-1} \mathbf{b}.$$

- The SOR method:

$$T_\omega := (D - \omega L)^{-1} ((1 - \omega)D + \omega U), \quad \mathbf{c}_\omega := \omega (D - \omega L)^{-1} \mathbf{b}.$$

2.3 Convergence criteria

We began our research by studying the algorithms and convergence criteria for each iterative method. Each method shares the same sufficient and necessary criteria about the working matrix in order for the method to converge to a solution, that is the matrix T must have a spectral radius less than 1:

$$\rho(T) < 1.$$

Here, $\rho(T)$ is defined as the radius of the smallest circle in the complex plane that contains all eigenvalues of matrix T . We know the matrix will converge if and only if it meets these conditions. For the SOR method, another necessary condition for convergence is that $0 < \omega < 2$. Each method also has some more conditions regarding the structure of the matrix that are sufficient to determine convergence. For Jacobi, if A is diagonally dominant, we know that T_j will converge. For Gauss-Seidel, the method is convergent if A is symmetric and positive-definite. For

SOR, T_ω is convergent if A is symmetric, positive-definite, and tridiagonal. For each method, the smaller $\rho(T)$ is, the faster the convergence speed is.

The program that we developed first checks the norm of residual after the estimated number of iterations are completed, and if the current residual norm is still above the threshold, i.e., $\|\mathbf{r}^{(k)}\| \geq \eta$, then the iterations will be continued. When the residual falls under the threshold, the program displays the final solution. It is important to note that this is an approximation, and we can never get the exact solution using iterative methods. In theory, iterations can continue infinitely. The purpose of introducing an error threshold is to stop iterations at a point where the residuals are sufficiently small.

To save the computational cost and avoid checking the residual norm every iteration, we can approximate the total number of iterations according to a given threshold η . For each method, we have the following error estimate of the iterative solution

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|,$$

therefore, for residual $\mathbf{r}^{(k)}$ we have

$$\begin{aligned} \|\mathbf{r}^{(k)}\| &= \|\mathbf{b} - A\mathbf{x}^{(k)}\| = \|A(\mathbf{x} - \mathbf{x}^{(k)})\| \\ &\leq \|A\| \times \|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|A\| \frac{\|T\|^k}{1 - \|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|. \end{aligned}$$

Consequently, for a given threshold η , the pre-determined total number of iterations is given by

$$\hat{k} \geq \frac{\log\left(\frac{\eta(1-\|T\|)}{\|A\| \times \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|}\right)}{\log(\|T\|)} = \frac{\log\left(\frac{\eta(1-\rho(T))}{\|A\| \times \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|}\right)}{\log(\rho(T))}, \quad \text{Equation 5.}$$

where $\rho(T)$ is the spectral radius of the matrix T .

For SOR method, the convergence speed depends on the optimal weight: $\omega^* = \operatorname{argmin}_{0 < \omega < 2} \rho(T_\omega)$, which is usually difficult to compute. However, if A is a positive definite and tridiagonal matrix, then the optimal ω^* for SOR is

$$\omega^* = \frac{2}{1 + \sqrt{1 - \rho(T_j)^2}}, \quad \text{Equation 6.}$$

where $\rho(T_j)$ is the spectral radius of the corresponding Jacobi method, and the optimal spectral radius for SOR is $\rho(T_\omega) = \omega^* - 1$.

2.4 Comparing Methods

Typically, we solve these matrices using direct methods via Gaussian elimination and row reduction. There are two main issues from using direct methods. Firstly, they are quite inefficient in general, and require an abundance of unnecessary computational memory resources and require far more time to reach a solution. When it comes to larger matrices containing data that can not be represented by just a few variables, or a matrix that is too large and sparse to find a solution in a reasonable amount of time, this issue becomes especially apparent. Secondly, there are cases in which direct methods are unable to reach a solution entirely, like in cases where a matrix might be non-invertible. These are two issues that iterative methods do not share.

Iterative methods really shine when it comes to real-world application as they can be applied very easily to large or sparse matrices. These methods are very time efficient, reaching solutions far faster than direct methods. They are far less taxing on memory storage and computational resources. This is since we only need specific information from large matrices or sparse matrices. These methods also have the ability to solve non-invertible matrices.

One of the key differences that allow for iterative methods to be computation- and memory-efficient is the use of matrix multiplication, which negates the reliance that direct methods have on finding an inverse matrix to find a solution.

2.5 Flow Networks and Traffic Modeling

In our search for real world applications of iterative methods, we came across flow networks. Flow network diagrams are a great way to visualize the data in linear systems. These diagrams are composed of two primary components. The first component being junctions which are labeled with capital letters in Figure 1 .

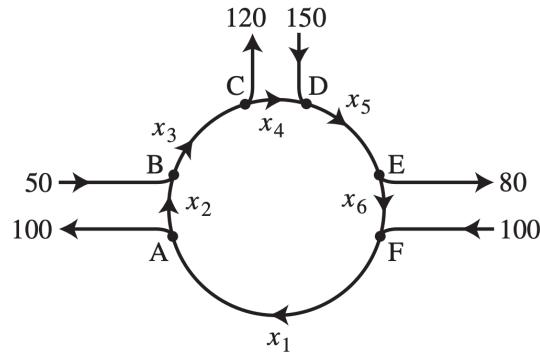


Figure 1. A graphical illustration of a simple traffic flow network.

Note that we can define a junction as any point or *node* with values flowing in or out. The second main component are *branches*, which are described by the directional flow (indicated by directional arrows) and flow amount (indicated by the number values, or placeholder variables). We can define branches as the connections between nodes.

We are able to use each junction's input and output values (defined by the branches entering and exiting each node) to create a linear equation. The input values belong to the left hand side of the equation, and the output is represented by the right. This can be done for every node in a diagram. For the illustration network in Figure 1, the associated linear system reads:

$$\begin{aligned}
 A : \quad x_1 &= x_2 + 100, \\
 B : \quad x_2 + 50 &= x_3, \\
 C : \quad x_3 &= 120 + x_4, \\
 D : \quad x_4 + 150 &= x_5, \\
 E : \quad x_5 &= 80 + x_6, \\
 F : \quad x_6 + 100 &= x_1.
 \end{aligned}$$

3 NUMERICAL SIMULATION

3.1 Modeling Highway I-485

The circular flow network discussed in Subsection 2.5 inspired us to think about modeling highway I-485 in Charlotte, as demonstrated in Figure 2.

Visually, it is reminiscent of a circular highway with multiple exits along the circumference. We also observed a notable pattern that encouraged us to explore using data from I-485. This example diagram alternates input and

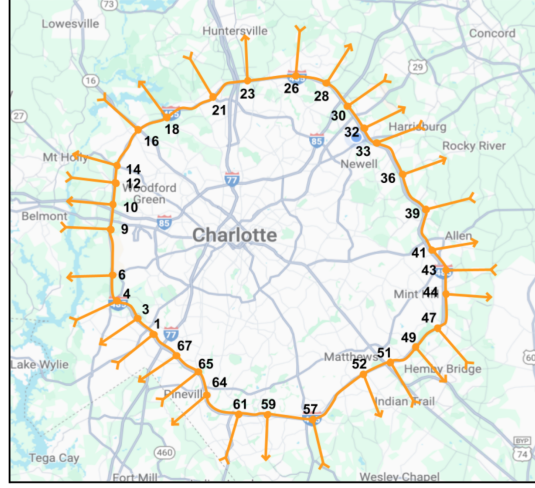


Figure 2. A traffic Network-Flow of highway I-485 in the city of Charlotte.

output data at each branch, resulting in the 1, -1, 0 pattern in the matrix, which can be recast as 32-by-32 matrix

$$A = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & -1 \\ -1 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}_{32 \times 32} \quad \text{Equation 7.}$$

However, the 32×32 matrix A is a circular matrix, hence A has a simple eigenvalue of zero. Therefore, any iterative matrix T resulting from this flow network has a spectral radius of 1, which makes it ineligible to solve with one of the three iterative methods. To solve this problem, we utilized some properties regarding the eigenvalues and eigenvectors of the matrix. Meanwhile, because that A had a simple eigenvalue $\lambda = 0$ and a corresponding eigenvector of $e = [1 \ 1 \dots 1 \ 1]^T$, by definition, $A \times e = \lambda \times e$, so $A [1 \ 1 \dots 1 \ 1]^T = \mathbf{0}$. From here we remove the last column of A in order to remove the simple zero eigenvalue, resulting in a new 32×31 matrix \tilde{A} . To modify the linear system accordingly, we must then also remove the last row of solution \mathbf{x} as it corresponds to that last column. Now we have defined a new linear system $\tilde{A}\tilde{\mathbf{x}} = \mathbf{b}$, which is not a square linear system. The next step is to make the equation into a normal equation

$$\tilde{A}^T \tilde{A} \tilde{\mathbf{x}} = \tilde{A}^T \mathbf{b},$$

resulting in a 31×31 matrix $\tilde{A}^T \tilde{A}$ and 31×1 vectors $\tilde{A}^T \mathbf{b}$ as well as the solution $\tilde{\mathbf{x}}$.

Now, we have a square linear system, we can enter it into our code¹ to calculate the values for $\tilde{\mathbf{x}}$. The calculations were made without the 32nd and final entry of the original \mathbf{x} vector, denoted by c so we must approximate a value for c and adjust $\tilde{\mathbf{x}}$ to compensate for the missing element. We decided that the *median* of the $\tilde{\mathbf{x}}$ values was the most reasonable approximation for c . The final step is to append the 32nd entry c back into $\tilde{\mathbf{x}}$, and then increment each element of $\tilde{\mathbf{x}}$ by the value of c . For the right hand side vector \mathbf{b} , we use the real world data from the North Carolina Department of Transportation website⁴. Consequently, the result of the linear system is the final approximation of the annual average daily traffic volume on each exit of I-485.

3.2 Results

The goal of our application of iterative methods to real world data was to accurately and rapidly approximate the values of the unknown \mathbf{x} vector for average traffic volume on each exit of I-485. After the manipulation of the traffic matrix to meet convergence criteria, we were able to upload the data to our program to be solved using iterative methods. SOR returned the smallest spectral radius, indicating that it would converge the fastest, providing us with the quickest results just as we were hoping for. We tested out each method with an error threshold of $\eta = 0.001$ to see just how different the convergence rates were, and found that SOR was nearly 40 times faster than Jacobi, a staggering result. The approximations were computed remarkably fast, and since we were able to adjust the matrix to fit the criteria for SOR (having the fastest convergence rate of the three methods) our calculations are executed in a fraction of a second. The results are summarized in Table 1.

Iterative Method	Spectral Radius	# of iterations
Jacobi	0.9952	3822
Gauss-Seidel	0.9904	1511
SOR	0.8215	99

Table 1. Performance summary of iterative methods solving traffic volume of each exit on highway I-485.

We summarize the traffic volume of each exit of I-485 in Figure 3. Each bar in the graph represents the approximated average number of cars found to be between each exit (i.e. column 1 represents the average number of cars from exit 1 to exit 3 and so forth).

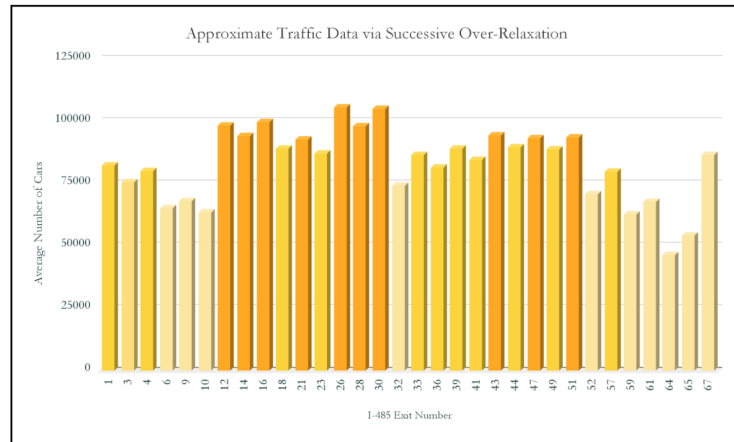


Figure 3. The annual average traffic volume on each exit of I-485 in the city of Charlotte.

3.3 Discussion

The information in Figure 3 provides powerful insight for potential Charlotte infrastructure. This data indicates areas of greater population density where lack of alternative routes are the cause of traffic congestion, like around exit 26 and exit 30. The framework for the data showcases flexibility, as if we were to hypothetically shut down an arbitrary number of exits for constructions or in the case of an accident, we would be able to approximate the traffic values of the segments with a new matrix that excludes those exits.

The actual numbers returned by the program were quite unexpected and opened up many questions about the current traffic situation in Charlotte. We found that the most dense areas in terms of car traffic are around exits 26 to 30, which are in the northernmost section of the highway and of the city. It is known that Charlotte is one of the nation's biggest banking hubs and most of these large employers are located in Uptown Charlotte, right in the center of the city and the highway loop. As Charlotte residents, we also know that a majority of these employees either live in the outskirts of the center city or further south in the more affluent neighborhoods. Although most

people who live in these areas likely would not need to take I-485 to commute Uptown, we did expect higher values around exits 52 through 65, which actually turned out to be some of the lowest values on the graph. We are not certain what causes this, but we can speculate a few things: (1) It could be that taking I-485 simply is not the most efficient route for most daily activities, including driving to work, school, grocery and other retail stores, and parks or other areas for leisure. (2) It is also possible that public transportation is more efficient in these areas, relieving some of the automobile congestion. (3) We also know that there has been more than five years of ongoing construction expanding this southern portion of the highway. It is possible that a few years ago our approximated AADT values would have been much higher due to closed lanes and detours. Now that the construction work is several years in, we are probably seeing the results of the highway expansion—that it seems to be effective in its goal to improve the flow of traffic in the area.

On the other hand, the northernmost area of the city is experiencing a lot more traffic on I-485—almost double. We did not expect this because this part of the city is not as densely populated as the southern region. The most dense areas of North Charlotte are UNC Charlotte and the area surrounding Concord Mills, a popular shopping center. These areas see the most traffic due to residents' daily commute to school, work, or to go shopping. UNC Charlotte falls closer to exit 33, however, which is not as packed as some of the previous exits. Furthermore, most students live either on campus or in apartments that are very close to the school. *This leads us to wonder if the level of traffic in these areas has not so much to do with the population but with the quality of infrastructure in this region.* As shown with our data and prior knowledge of South Charlotte, even though it holds a large percentage of the population, its AADT values are relatively low. We can speculate that residents of North Charlotte likely do not have as effective alternatives to the highway in order to reach the more popular destinations, like work or grocery stores.

There are some ways to address these issues, they just require years of planning and more money than the state is willing to spend. One solution is to promote denser urban growth instead of sprawling out further. This can be challenging because current residents may oppose further construction in their neighborhoods, living in crowded areas, or building on top of green spaces. To mitigate these concerns, it is crucial to pair denser development with strategic distribution of amenities throughout the city. Better access to services may eliminate the need to drive as often, opening up roadways and providing residents with enhanced quality of life. Another possibility is to create more connections to I-485 and surrounding roadways to provide alternative routes. There are some stretches of highway that go several miles without an exit as well as areas parallel to the highway that would benefit from new routes to divert traffic. Finally, the state can implement tolls and congestion pricing. This is another option that would result in some pushback, but the revenue would be directed to improving infrastructure. Congestion pricing has been implemented in many cities around the world and it has shown to improve travel times and quality of life. In New York, where drivers lose 117 hours a year and about \$2,000 in traffic, congestion pricing is estimated to reduce the number of cars on the road by 100,000¹². Congestion pricing is also meant as an incentive to opt for other methods of transportation, such as public transit, walking, or biking. This would require more resources to be put into improving public transportation and the walkability of the city. A multi-faceted approach like this, though challenging and requiring significant investment, could lead to long-term benefits in reducing traffic congestion, enhancing urban living, and promoting sustainable growth.

4 CONCLUSIONS

In the large majority of cases, iterative methods have shown themselves to be the optimal approach to solving linear systems, especially when it comes to large or sparse real world data sets. We have seen that out of the three methods that we studied, the Jacobi Method is the most widely applicable with its less strict convergence criteria, although taking a considerably higher amount of iterations to meet the given error threshold. The Gauss-Seidel Method showed higher convergence rates than Jacobi, while having stricter convergence criteria. In all cases, SOR proved to have the highest convergence rates of the three by a substantial amount, while also having the strictest

convergence parameters including the weight parameter ω which must be between zero and two. If able to meet the criteria for SOR, it is the most optimal iterative method of the three taking a fraction of the time and resources that the other methods need.

In general, iterative methods have a superior convergence rate and are overall more efficient in computation and memory costs than direct methods, showcasing superior practicality due to their use of matrix multiplication. These methods meet a demand that is ever-present in the new age of data collection and processing. We were able to compare the effectiveness of the three methods directly after creating a flow network diagram to translate traffic values into a matrix form for iterative method application, using the traffic AADT data on highway I-485 in the city of Charlotte, North Carolina. After inserting the data into our program, and manipulating the I-485 matrix to fit the criteria of SOR using properties of eigenvalues and eigenvectors, we were able to showcase SOR's efficiency by approximating the solution to the total 32 highway exits in less than 100 iterations, and within a fraction of a second.

The approximations we obtained highlight areas of high traffic congestion in the greater Charlotte area (northern and eastern hemispheres of I-485) opening up potential insights for infrastructural improvements across the city. With the scalability of the model we have developed utilizing flow networks and our Python program, we can also approximate traffic values if we were to close any amount of exits on I-485. This would prove useful in cases of road construction, or in the case of accidents obstructing the roadways. Scalability is a major advantage for many different potential applications and with the use of flow networks, translation of real data has become more efficient and we look forward to inserting larger matrices into our program, and increasing the scale of our research.

In the future, we would like to apply what we have learned to a much larger system to observe the flow of electricity for Duke Energy's power grid and highlight areas that might be lacking in sufficient power. We would also like to optimize our Python program to further streamline the process of reading new matrices for real world application with large datasets in future projects.

5 ACKNOWLEDGEMENTS

The authors thank Dr. Xingjie Li for her continuous support and mentorship all throughout and outside of the research process. The authors are incredibly grateful for the opportunities that she has provided and funding support from the NSF Award DMS-1847770.

REFERENCES

1. Our codes and values for all exits on I-485, as well as other information expanding approximations are available at:
<https://github.com/fabiola-rojas/iterative-methods-research2024>
2. Chapter 5 Iterative Methods for Solving Linear Systems. Courtesy of the University of Pennsylvania,
<https://www.cis.upenn.edu/~cis5150/cis515-12-sl5.pdf> (accessed Mar 2024).
3. Lay, D. C., Lay, S. R., and McDonald, J. (2016) *Linear Algebra and Its Applications*, Pearson, Boston.
4. NCDOT. N.C. Department of Transportation Annual Average Daily Traffic (AADT) Mapping Application, *ArcGIS web application*,
<https://ncdot.maps.arcgis.com/apps/webappviewer/index.html?id=964881960f0549de8c3583bf46ef5ed4> (accessed Mar 2024).
5. Section 7.3: the Jacobi and Gauss-Seidel Iterative Methods. Courtesy of the University of Notre Dame:
<https://www3.nd.edu/~zxu2/acms40390F12/Lec-7.3.pdf> (accessed Mar 2024).
6. Section 7.4: Relaxation Techniques for Solving Linear Systems. Courtesy of the University of Notre Dame:
<https://www3.nd.edu/~zxu2/acms40390F13/Lec-7.4-5.pdf> (accessed Mar 2024).
7. Schrank, D., Eisele, B., Lomax, T. (2019) Urban Mobility Report. Courtesy of the Texas A&M Transportation

- Institute,
<https://static.tti.tamu.edu/tti.tamu.edu/documents/umr/archive/mobility-report-2019.pdf>.
8. United Nations (2024) Current Metro Area Population of Charlotte. Data from Macrotrends,
<https://www.macrotrends.net/global-metrics/cities/22954/charlotte/population> (accessed Aug 2024).
 9. Carne, S. (2023) Charlotte Ranks Close to Last on Walkability. WFAE,
<https://www.wfae.org/charlotte-area/2023-05-07/charlotte-ranks-close-to-last-on-walkability>
(accessed Aug 2024).
 10. Kukkapalli, V. M., and Pulugurtha, S. S. (2019) *Effect of Road Construction on Travel Time*, Mineta Transportation Institute, San Jose State University,
<https://transweb.sjsu.edu/sites/default/files/1702B-Pulugurtha-Effect-Road-Construction-Travel-Time.pdf> (accessed Aug 2024).
 11. Budhu, G., and Grissom, D. (1985) *Freeway Traffic Management: The Chicago Area Expressway Surveillance Project*, Transportation Research Record, No. 1046, Transportation Research Board,
<https://onlinepubs.trb.org/Onlinepubs/trr/1985/1046/1046-007.pdf>.
 12. Martin, N., Kahn, D., and St. Juste, S. (2024) *The Benefits of Congestion Pricing*. Office of the New York City Advocate,
<https://advocate.nyc.gov/blog/the-benefits-of-congestion-pricing> (accessed Aug 2024).

ABOUT THE STUDENT AUTHORS

Dominic Kealoha is a senior in UNC Charlotte's College of Computing and Informatics concentrating in AI and is expected to graduate with a bachelor's degree in the Spring of 2025. Dominic has interests in applied mathematics in the field of data science and aspires to find potential implementations of mathematics to contribute to the field of artificial intelligence. Specifically, Dominic hopes to find ways to optimize the methods of the sorting and collection of training data for AI development.

Fabiola Rojas is a student at UNC Charlotte's School of Data Science, expected to graduate in the Spring of 2025. Her inclination towards mathematics and statistics as well as interests in computer science and social matters drew her to pursue this study of her community. She hopes to continue to combine her love for mathematics and passion for social causes in her future career as a data scientist.

This research project marks the authors' first venture into academic research. With only undergraduate-level coursework in mathematics and computer science, they successfully conducted a detailed study on the traffic flow I-485—a topic that emerged due to its personal relevance—while incorporating advanced mathematical concepts.

During the course of their study, the authors have developed valuable skills that they will continue to refine and apply throughout their academic careers. They have gained significant experience in independently conducting research, formulating relevant questions, and articulating their findings in a professional manner. Both Dominic and Fabiola are committed to seeking further STEM-related opportunities and making a broader impact through their work. They are eager to continue exploring and making contributions to the STEM field.

PRESS SUMMARY

This study focuses on iterative methods such as Jacobi, Gauss-Seidel, and Successive Over-Relaxation (SOR), and how they were utilized to model traffic flow on the I-485 highway in Charlotte, NC. These methods, known for their efficiency in solving large systems of linear equations, were implemented in a Python program developed by the students to assess traffic data retrieved from the N.C. Department of Transportation and model the daily average number of cars on any given segment of the highway I-485, which is the one of most important highways in Charlotte. The study revealed that the Gauss-Seidel and SOR methods, particularly the latter, offered faster convergence rates, providing rapid approximations of traffic flow. The results highlighted a significant congestion near

exits close to the UNC Charlotte campus, suggesting that city resources are not sufficiently focused on the busiest areas. This research not only advances the application of mathematical modeling in traffic management but also underscores the need for targeted infrastructure improvements in high-traffic regions.