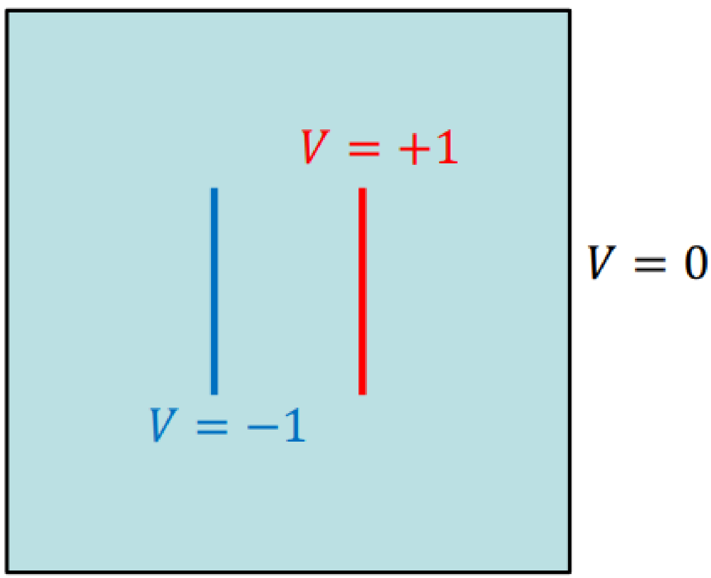
**Project 2**

Analysis of the Electric Field of a Parallel-Plate Capacitor using Relaxation Methods

Douglas Nyberg and Haleigh Brown




Friday, April 14, 2023, 11:59 PM

# 1 Abstract

This project contains the computation of the electric potential within a two-dimensional grounded square enclosure containing parallel plates maintained at antipodal potentials utilizing methodologies such as the Jacobi and simultaneous over-relaxation (SOR). The primary objective of this investigation is to optimize the weighting and scrutinize the convergence scaling for both relaxation methodologies, thus facilitating a comprehensive understanding of the electric field behavior within the capacitor and the effectiveness of relaxation methods in addressing partial differential equations through the solution of a set of linear simultaneous equations.

# 2 Introduction

To determine the electric potential of a parallel-plate capacitor within a two-dimensional grounded square enclosure containing parallel plates held at antipodal potentials ($\pm 1$). The plates in question are assumed to be half as tall as the side of the box. The positioning of the plates allows for 1/3 of the box to be situated to the left of the negative plate, 1/3 in the middle, and 1/3 to the right of the positive plate (as can be seen on the cover of this project). It is imperative to solve the system, described by partial differential equations, to discern clear distinctions in the overall efficacy of relaxation methods and achieve optimization of the weighting factors.

One approach to resolving partial differential equations, the finite differences method, involves segmenting a grid space into discrete points. Computations must be performed on each row/column of the grid space, necessitating second-order partial derivatives for every discrete point with respect to each dimension. The finite difference method of breaking down PDE's can be employed to approximate solutions across the entire grid space. Using this method as a basis, we can employ more optimized methods such as Jacobi, Gaussian elimination, and SOR to solve our PDE's as a collection of linear simultaneous equations.

When utilizing relaxation methods, an initial guess can be made for all discrete points. Subsequently, each discrete point is iteratively calculated based on the average of its adjacent points within the entire grid space. This process continues until convergence to an acceptable solution, adhering to a predetermined target accuracy, is achieved. In the Jacobi method calculations for all grid points are performed and then stored and compared to previous calculations until a minimal difference is detected and thus convergence is found. This method takes up a lot of computational power and memory space as it looks at each point individually during each iteration and requires storing past and current solutions. This method may not be suitable for large grid spaces or when the initial guess is considerably distant from the acceptable solution. In such cases, the Gauss-Seidel implementation with Simultaneous Over Relaxation (SOR) should be considered as an alternative for an unsatisfactory initial guess or for extensive grids. The rationale behind employing Gauss-Seidel SOR is its capacity to update alternating discrete points by adopting a checkerboard design and weight each iterative solution to over-estimate values, and potentially decrease the number of iterations needed to reach convergence. In this method, one set is updated using only the ”black squares” of the checkerboard, followed by updating the other set with the new values from the ”black squares.” Finally, the entire solution is weighted by a value usually ranging from $1.0 < w < 2.0$ in order to overestimate our step towards convergence. Subsequent iterations involve performing the inverse and continuing this pattern until an acceptable solution is reached.

For calculating the electric potential of the described problem the solution will be acceptable when the values converge to,

$$Max|V_{new} - V| < \varepsilon$$

In the context of the plate behavior, envisioning the negative plate as a trough and the positive plate as a hill, one can imagine a positive test charge as water to analyze various scenarios. When the water is located in the first 1/3 region to the left of the trough, it flows directly into the trough. If the water is situated in the middle 1/3 region, it still flows into the trough; however, the vertical distance it gains from the hill’s incline while in that middle region results in an increased ”velocity” toward the trough. Finally, when the water is in the furthest right 1/3 region, it gradually flows around the hill to ultimately reach the trough. If the distance between the plates is reduced then the velocity of the water should only increase towards the trough. These predictions will be confirmed by utilizing the following general finite difference equations and the equation for the Electric field in the code.

$$f_x(x,y) = \frac{f(x+h,y) - f(x-h,y)}{2h}, \quad f_y(x,y) = \frac{f(x,y+k) - f(x,y-k)}{2k}$$

$$\text{Where,} \quad h = k = \frac{1}{\text{Number of Grid Points}}$$

$$E = -\left[i\frac{\partial}{\partial x} + j\frac{\partial}{\partial y} + k\frac{\partial}{\partial z}\right]V$$

# 3 Results

Firstly, the code will utilize a grid space of $10^4$, where $N = 100 =$ grid points per side. With an established N the code will provide some initial information optimal weighting value. The output for the optimal weighting value is $\boxed{\text{W = 1.5}}$.To compare the scaling of the convergence of some, $N = 50, 100, 200, 400$ for basic Jacobi and an SOR, the previously calculated optimal weighting value of 1.5 can be used for SOR.

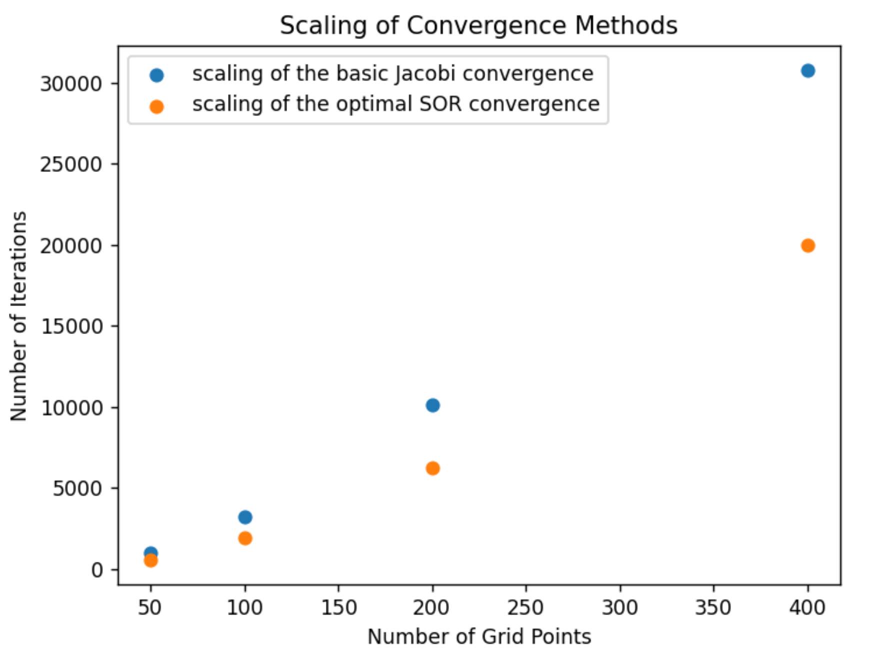
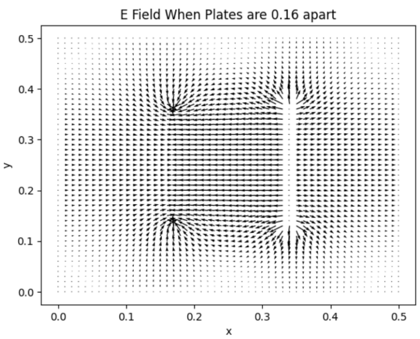
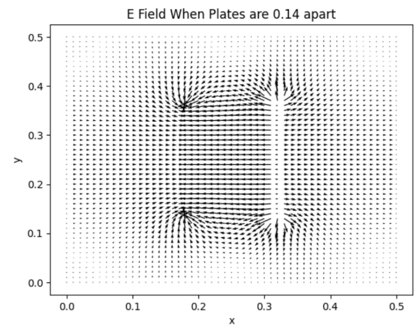



Figure 1: Scaling of convergence methods (number of iterations vs number of grid points)

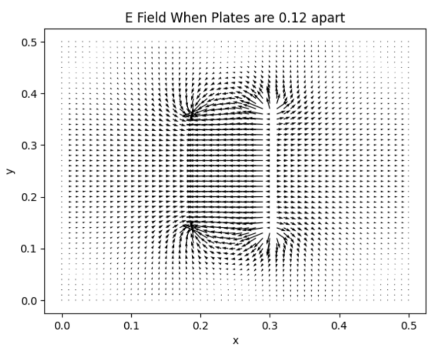
Lastly, to confirm the qualitative expectations of the electric field along with separation expectations previously stated in the form of the water example, a quiver plot of the electric field will be shown below, along with plots demonstrating how the electric field is affected by changing the separation distance.



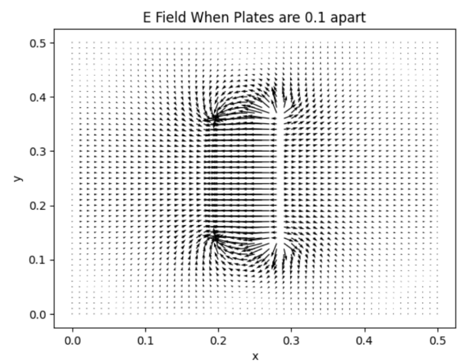

(a) $\Delta x = 0.16$




(b) $\Delta x = 0.14$




(c) $\Delta x = 0.12$




(d) $\Delta x = 0.1$

Figure 2: How the electric field lines change with distance

# 4 Conclusions

The optimal weighting value was calculated to be 1.5. Following this, an estimation of the average time complexity was conducted by scaling all values of N, excluding N = 400. The reasoning for excluding N=400 is simply because the difference in completions between N=400 and N =200 was too high and we wanted to obtain a concrete estimation for not only the time complexity but for the estimated time N=400 should take. The calculations employed the doubling hypothesis, and time measurements were taken using a non-code-implemented stopwatch. As a result, there is potential for significant deviations. An example of the equations employed is as follows:

$$t = aN^b$$

Where t is the time, a is a constant, N is the size of the dataset, and b is the algorithmic complexity. If we have two times where N doubles, then we have the following to calculate the time complexity.

$$t_1 = aN^b, \quad t_2 = a(2N)^b$$

$$\frac{t_2}{t_1} = \frac{a(2N)^b}{aN^b} = 2^b, \quad \log_2(2^b) = b$$

so,

$$\log_2(671.4/47.5) = 3.82$$

From this analysis, an estimation of the time required for N=400 on our machine is derived. However, it is crucial to emphasize that the estimated time for N=400 is not the main focus. Instead, the calculated average time complexity holds greater significance.

$$\left(\frac{400}{200}\right)^{3.8} = 2^{3.8} = 14.1,$$

$$\text{Time}(N = 400) = (671.4)(14.1) = 9,467.94s = 2.62\,\text{hours}$$

| Runtimes | | |
|---|---|---|
| N | Time(seconds) | $\Theta(AverageTimeComplexity(N, N-1))$ |
| 50 | 3.64 | 3.71 |
| 100 | 47.5 | 3.82 |
| 200 | 671.4 | NA |
| 400 | NA | - |

An examination of Figure 1 reveals the significant impact of weight on the total number of iterations. Although both methods exhibit exponential growth, the optimal weighting SOR method displays a markedly lower factor compared to the Jacobi method (SOR Gaussian sidel method tends to take roughly 2/3 the amount of time basic Jacobi takes). The expected efficiency of SOR over Jacobi can be attributed to the increasing grid sizes and the initial guess of zero for the system. Nevertheless, Jacobi's performance is commendable, given that it did not benefit from the optimal weight.

Incorporating a stopwatch into the code for a more precise comparison of the methods would have been insightful; however, the observed run times were not anticipated, and the code has been running for hours. The time required for more complex problems using only the Jacobi method for relaxation is difficult to imagine. Despite the impressive nature of these methods, which provide approximations for partial differential equations at each grid point, the run time remains considerable. The degree to which an inadequate initial guess affects the extended run time is unclear; however, the prospect of using data structures that do not compile to machine language, resulting in even longer run times, is daunting. The use of NumPy arrays is highly valued, as running the N=400 case for this relatively simple project might have taken a week otherwise.

The electric field lines provided in Figure 2 represent what was previously anticipated. The strength of the electric field lines correlates with the velocity of the water for the trough/hill example.