NumMeth MAT-410 Lab 3 Report

Moritz M. Konarski

December 10, 2020

Contents

1 Introduction

The third laboratory project of this course deals with two–dimensional Poisson equation (also called Dirichlet equations). We consider these equations on the unit square Ω, defined as follows

$$\Omega = \{(x,y)|0 \leq x \leq 1, 0 \leq y \leq 1\}.$$

The borders of this unit square are defined as $\partial\Omega$

$$\partial\Omega = \{x = 0, 0 \leq y \leq 1\} \cup \{x = 1, 0 \leq y \leq 1\} \cup$$
$$\{0 \leq x \leq 1, y = 0\} \cup \{0 \leq x \leq 1, y = 1\}.$$

The Poisson equations describes a thermal field whose source is the function $f(x,y)$, and the operator Δ is the Laplace Operator, as seen below.

$$-\Delta u(x,y) \equiv -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x,y), \quad (x,y) \in \overset{0}{\Omega} \equiv \Omega \backslash \partial\Omega$$
$$u(x,y) = 0, \quad (x,y) \in \partial\Omega$$

To solve this equation we consider a uniform two–dimensional grid called Ω^h. On this grid we have the function

$$\bar{u}^h_{i,j},$$

which represents the numerical solution on Ω^h. The variables (i, j) here are the indices of the grid, where $x_i = i \cdot h$ and $y_j = j \cdot h$ (h is the step size). The function $f(x,y)$ is now written as $f_{i,j}$. This report uses a point–wise form of the finite–difference solution. By iterating this method multiple times, a desired accuracy of the approximation can be reached. Here k is the iteration parameter. For each point $u^{(k)}_{i,j}$ we have

$$u^{(k)}_{i,j} = \frac{\theta}{4}\left(u^{(k)}_{i-1,j} + u^{(k)}_{i,j-1}\right) + \frac{\tau-\theta}{4}\left(u^{(k-1)}_{i-1,j} + u^{(k-1)}_{i,j-1}\right)$$
$$+\frac{\tau}{4}\left(u^{(k-1)}_{i+1,j} + u^{(k-1)}_{i,j+1}\right) + (1-\tau)u^{(k-1)}_{i,j} + \frac{\tau h^2}{4} \cdot f^h; \quad (i,j) \in \Omega^{h,0} \tag{1}$$
$$u^{(k)}_{i,j} = 0, \quad (i,j) \in \partial\Omega^h$$

The parameters τ and θ as well as the function $f_{i,j}$ are determined by the methods or the task, respectively. Equation (1) is used in an iteration over all steps in x for each of the levels in y. Thus equation (1) approximates the Poisson equations using repeated iteration for each point in $\Omega^{h,0}$, the inner part of the uniform grid.

1.1 The Task

This report specifically covers the eigenfunctions of the grid Laplace operator Δ^h. According to my readme file, I should choose the parameters l and m to be 2. Accordingly, the analytic solution in the unit–square Ω is defined as

$$u^h_{i,j} = \sin(2\pi x_i) \cdot \sin(2\pi y_j). \tag{2}$$

The function $f_{i,j}$ is defined as

$$f^h_{i,j} = \lambda_{2,2} - \sin(2\pi x_i) \cdot \sin(2\pi y_j)$$
$$\lambda_{2,2} = \frac{4}{h^2} \cdot \left[\sin^2\left(\frac{2\pi h}{2}\right) + \sin^2\left(\frac{2\pi h}{2}\right)\right]$$

for the values $l = 2$ and $m = 2$. It has to be noted that because $l = m$ the error of the approximation for the x and y slices is identical and thus they do not need to be differentiated.

1.2 The Methods

The methods used to solve the Poisson equations through iteration are Jacobi's method and Seidel's method. For Jacobi's method, the parameters τ and θ are chosen as $\tau = 1$ and $\theta = 0$. For Seidel's method, $\tau = 1$ and $\theta = 1$. These parameters influence the behavior of the point–wise iterative function in (1) and lead to different processes of approximation.

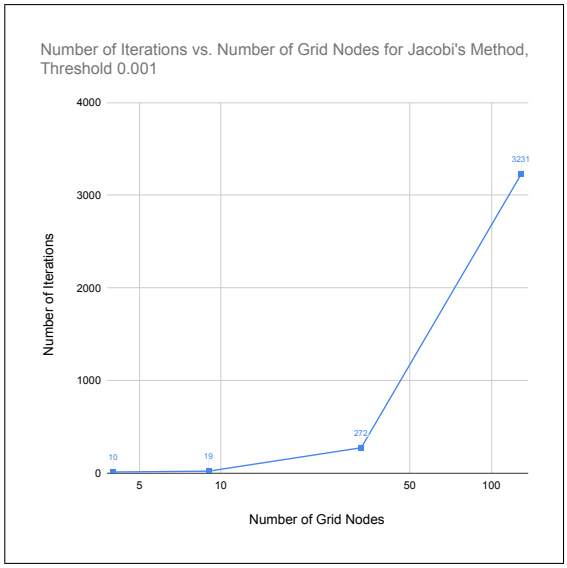
In my program the user chooses an error threshold σ that, when the error is lower than it, stops the iterative process. This will halt the program at the minimum required number of iterations to satisfy the error threshold.

In the following sections I will compare the speed of convergence for both methods based on the number of grid nodes, their convergence based on the values of the parameters l and m (different from the ones I was given), and the effectiveness of the methods as measured by the number of required iterations to achieve a predetermined accuracy σ, the errors of each method, and the visual proximity of the graph of the numerical solution to the analytic solution.

2 Speed of Convergence

The speed of convergence for both methods strongly depends on the parameter n, the number of grid nodes. The higher the number of grid nodes is, the more iterations are required to reach the error threshold σ. I think the reason for this is that for each new grid node, additional computation (and iteration) is required to approximate its correct position. Increased numbers of grid nodes will also increase the overall accuracy of the approximation because more nodes allow a more fine–grained fit of the numerical solution to the analytical solution.

In Figure 1 (left graph) we can see that the number of required iterations to reach an error threshold of $\sigma = 0.001$ drastically increases as the number of nodes increases. This is not just caused by this particular error threshold, the graph on the right shows the same behavior for a much smaller error threshold of $\sigma = 0.00001$. Through Figure 1 we can also note that the number of required iterations increases as the error threshold decreases – this will be covered in more detail later.



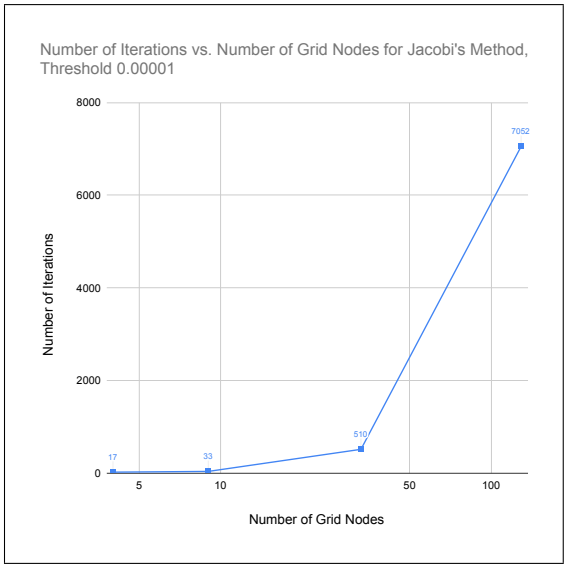
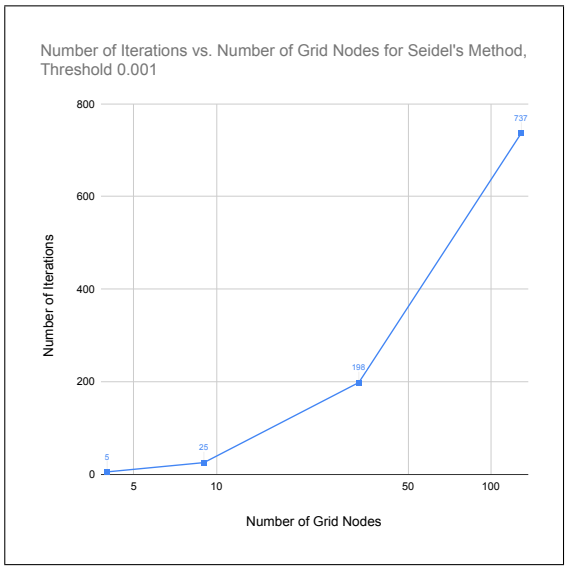


Figure 1: Iterations for Jacobi's Method, $\sigma = 0.001$ and $\sigma = 0.00001$

When considering Seidel's method, much the same behavior from Jacobi's method is evident. Figure 2 (left graph) shows the required iterations of Seidel's method for a threshold of $\sigma = 0.001$ as the number of grid nodes increases. The right graph shows the required iterations for a threshold of $\sigma = 0.00001$.



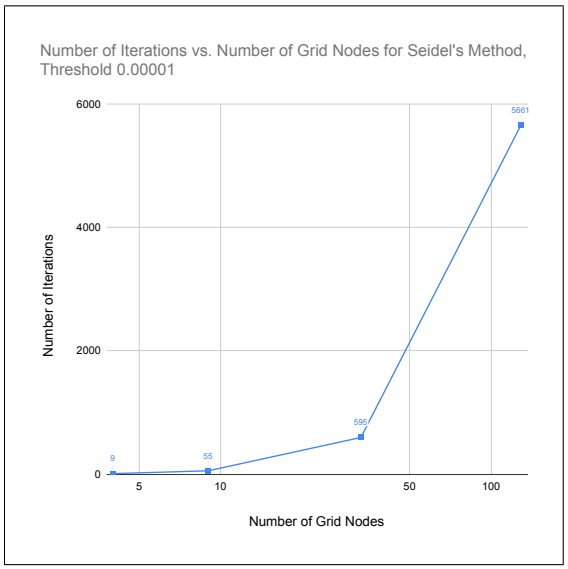


Figure 2: Iterations for Seidel's Method, $\sigma = 0.001$ and $\sigma = 0.00001$

3 Complexity of the Solutions

The parameters l and m determine how many periods of the sine functions in equation (2) exist on the interval $[0, 1]$. The parameter l influences the periods in the x dimension and m the periods in the y dimensions. For multiples of 2 one full period exists. For my values of

$l = m = 2$, one full period of the sine wave exists on the interval $[0, 1]$ in both the x and y dimensions.

The values of these parameters influence the number of iterations required to reach a certain error threshold. Unfortunately, I was not able to determine exactly in what way. In one case, decreasing l to 1 and increasing m to 3 increased the number of required iterations. This can be seen in Figure 3 looking at the red graph. In another case, increasing l to 10 and increasing m to 4 dramatically decreased the number of required iterations. This behavior can be see in the yellow graph in Figure 3. The blue graph shows the required iterations for $l = m = 2$ for reference.

I could not find a correlation between the values of l and m and the number of required iterations to achieve an error below $\sigma = 0.0001$. The only thing this shows is that the required iterations change dependent on the values of l and m.

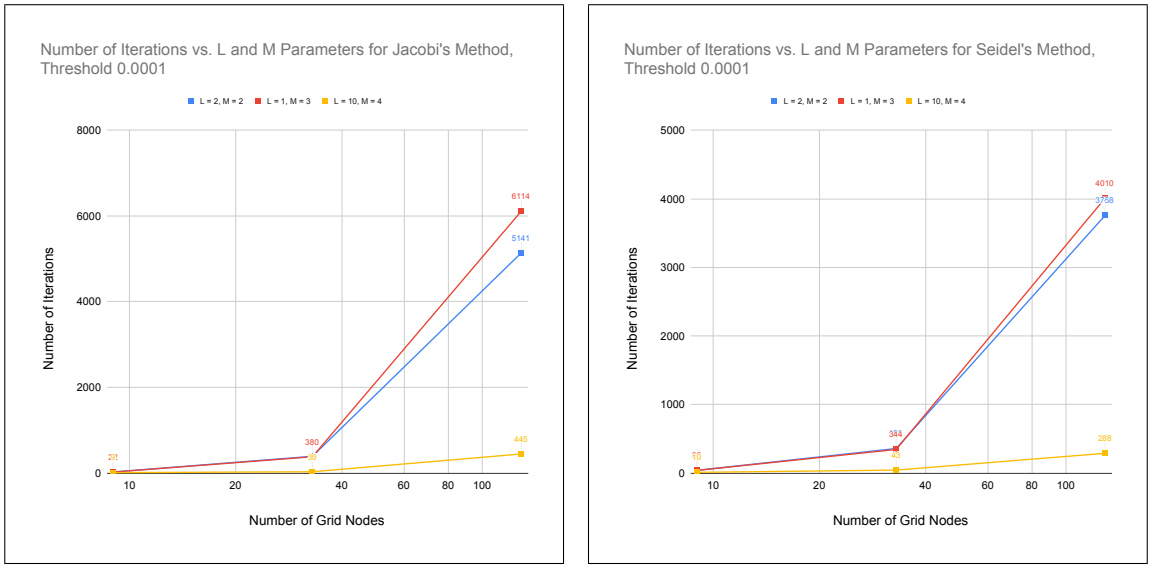


Figure 3: Iterations for Jacobi's and Seidel's Method, $\sigma = 0.0001$, varying l and m

4 Effectiveness of the Methods

This section will evaluate the effectiveness of the methods as measured by the number of required iterations to achieve a predetermined accuracy σ, the errors of each method, and the visual proximity of the graph of the numerical solution to the analytic solution.

4.1 Number of Required Iterations

My analysis of the number of required iterations to reach a certain error threshold σ was inconclusive. For certain combinations of σ and n Jacobi's method requires fewer iterations than Seidel's method while for other combinations it is the other way around.

Nonetheless, for the majority of combinations Seidel's method required fewer iterations to reach the error threshold. In Figure 3 we can see that the highest numbers of iterations for

$\sigma = 0.0001$ and $n = 129$ for Jacobi's method are $5141, 6114, 445$, while for Seidel's method the errors for the same combination are $3758, 4010, 288$. All errors for Seidel's method here are smaller than the ones for Jacobi's method.

Furthermore, comparing two of the previous figures for $\sigma = 0.00001$ in Figure 4 for Jacobi's method and Seidel's method shows that for $n = 4, 9, 33$ Jacobi's method requires fewer iterations than Seidel's method. When $n = 129$, Seidel's method requires fewer iterations than Jacobi's method. This is one example of how one of the methods is better than the other and then that changes when n increases.

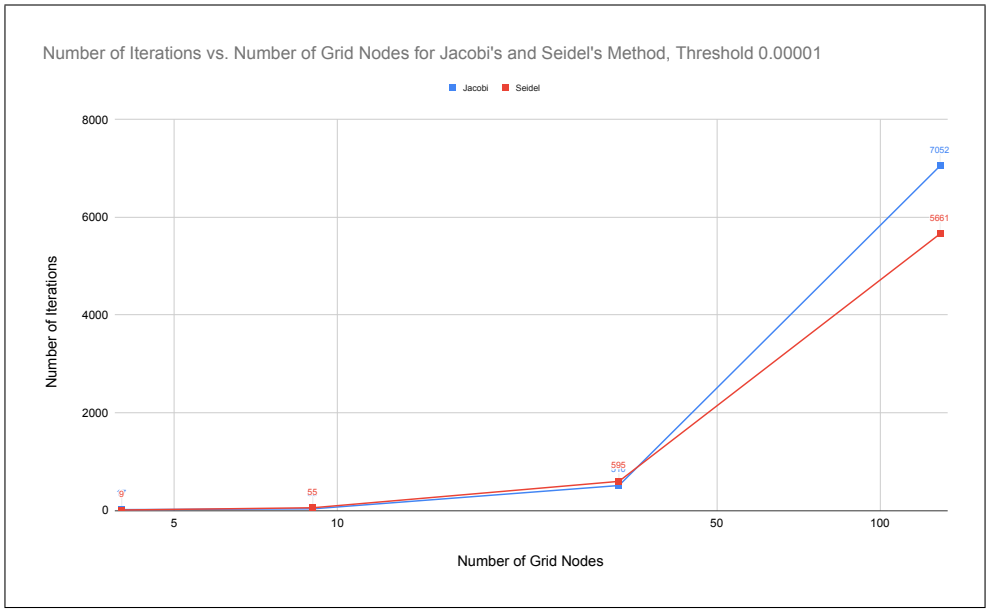


Figure 4: Iterations for Jacobi's and Seidel's Method, $\sigma = 0.00001$

4.2 Errors

Both methods converge to the analytic solution and thus the actual solution. Out of the two methods, Seidel's method converges much faster than Jacobi's method at first. After that, the error of its approximation increases again, before approaching the error threshold again. Jacobi's method has no such quirks and steadily decreases its error with each iteration. Figure 5 shows this behavior. Please note that the error is displayed on a logarithmic scale with base 10.

Figure 5: Error for Jacobi's and Seidel's Method, $\sigma = 0.00001$, $n = 129$

4.3 Visual Proximity of the graphs

In this comparison Seidel's method clearly is the winner. Because of the rapid convergence shown in Figure 5, the numerical solution quickly comes close enough to the analytic solution that they look identical. This does not necessarily mean that the error threshold is reached more quickly, but it looks like it. In Figure 6 we can see that for $n = 129$ at 250 iterations, Seidel's method (on the right) has already converged much close to the analytic solution than Jacobi's method (on the left).

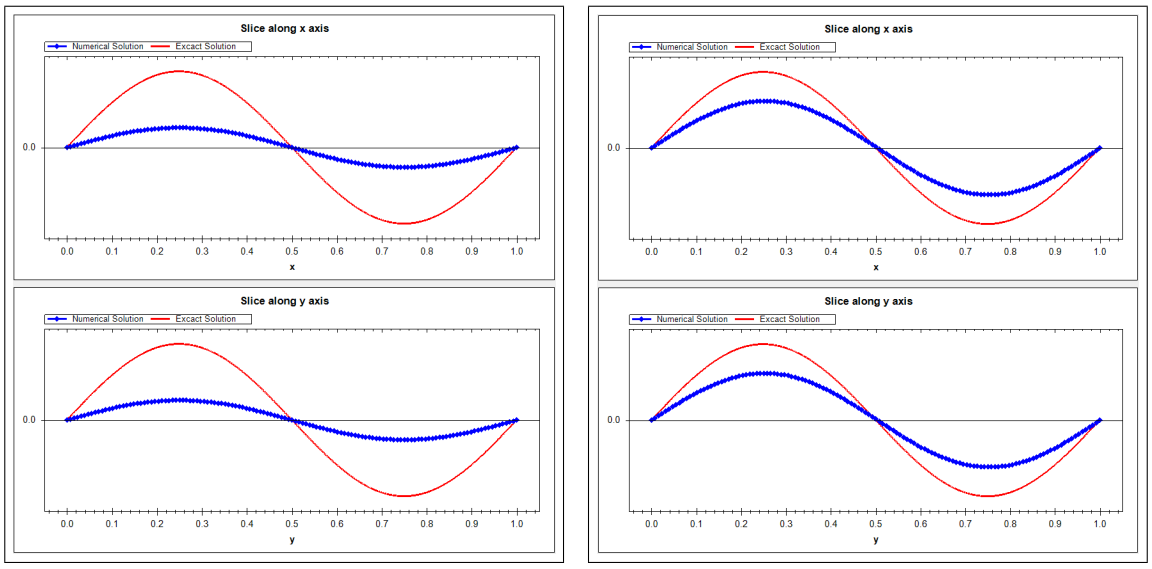


Figure 6: Visual comparison of Jacobi and Seidel method graphs for $n = 129$, 250th iteration

The next graph in Figure 7 shows that when we have the same parameters as in Figure 6 but

iterate 750 times, Seidel's method has already converged to the analytic solution while Seidel's method is still noticeably further away.

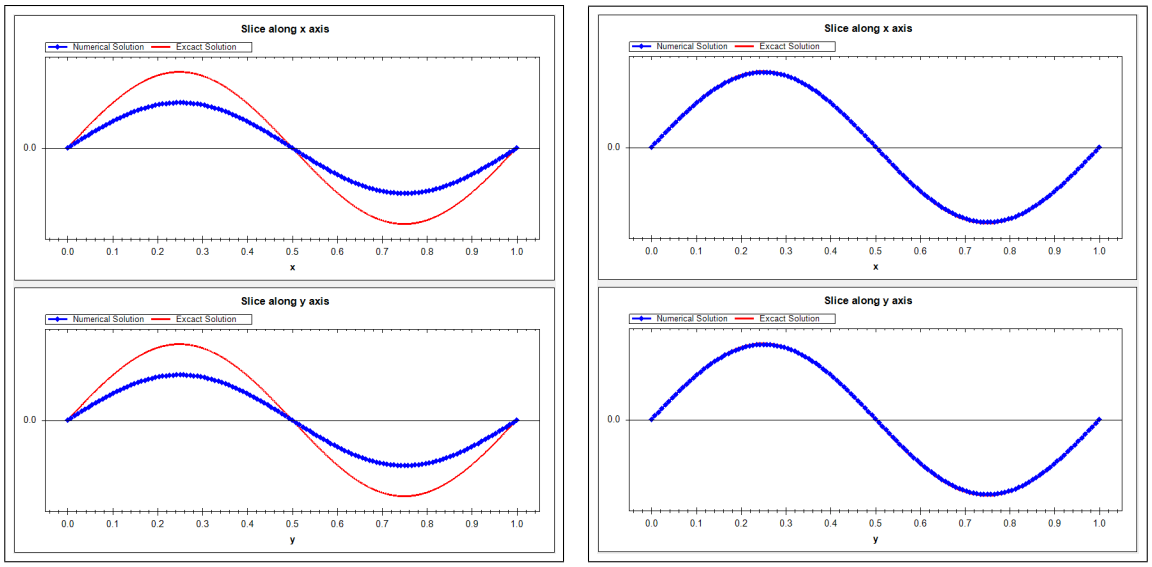


Figure 7: Visual comparison of Jacobi and Seidel method graphs for $n = 129$, 750th iteration

We can see that when it comes to visual proximity of the numerical solution to the analytic solution Seidel's method is a lot more effective than Jacobi's method.

5 Conclusion

Based on the features this report covered, I conclude that Seidel's method is superior to Jacobi's method. Seidel's method generally requires fewer iterations to reach a specific error threshold, it converges to the analytic solution faster, and its visual proximity to the analytic solution increases faster. Considering all of these factors, I think Seidel's method is better.

That does not mean, however, that Jacobi's method is bad. One big advantage of this method compared to Seidel's method is that it converges consistently. Seidel's method exhibits the "bounce" that can be seen in Figure 5 but Jacobi's method does not. This means that one can be sure that for a higher number of iterations the accuracy will increase, while this is uncertain for Seidel's method. In my opinion this does not make up for the disadvantages this method has compared to Seidel's method though.