SOLVING THE RESISTOR LADDER PROBLEM USING MATRIX MATH

Wissam Razouki
CSE 3802

INTRODUCTION

The Problem

The equations for solving for the voltages at each node for nodes 1 through N in the resistor ladder are given as:

$$\begin{aligned}
&Node\,1: \quad I_s = \frac{V_1 - V_2}{R_{s1}} \quad or\ I_s = (V_1 - V_2)\,G_{s1} \\
&Node\,2: \quad \frac{V_1 - V_2}{R_{s1}} = \frac{V_2 - V_3}{R_{s2}} + \frac{V_2}{R_{g1}} \quad or\,(V_1 - V_2)\,G_{s1} = (V_2 - V_3)\,G_{s2} + V_2 G_{g1} \\
&Node\,3: \quad \frac{V_2 - V_3}{R_{s2}} = \frac{V_3 - V_4}{R_{s3}} + \frac{V_3}{R_{g2}} \quad or\ \ (V_2 - V_3)\,G_{s2} = (V_3 - V_4)\,G_{s3} + V_3\,G_{g2} \\
&\ldots \\
&Node\,N: \quad \frac{V_{N-1} - V_N}{R_{s\,N-1}} = \frac{V_N}{R_{g\,N-1}} \quad or\,(V_{N-1} - V_N)\,G_{s\,N-1} = V_N\,G_{g\,N-1}
\end{aligned}$$

Where the current source I_s and the resistor values R_s and R_g are known, and $G = \frac{1}{R}$.

In order to convert this linear system to matrix form, we must first rewrite the equations in the form $GV = I$.

Rewritten Equations

The above system can be rewritten as:

$$G_{s1}V_1 - G_{s1}V_2 = I_s$$
$$G_{s1}V_1 - \left(G_{s1} + G_{s2} + G_{g1}\right)V_2 + G_{s2}V_3 = 0$$
$$G_{s2}V_2 - \left(G_{s2} + G_{s3} + G_{g2}\right)V_3 + G_{s3}V_4 = 0$$
$$\vdots$$
$$G_{sN-1}V_{N-1} - \left(G_{sN-1} + G_{gN-1}\right)V_N = 0$$

This system can easily be converted to the matrix form $\boldsymbol{Ax} = \boldsymbol{b}$ (or $\boldsymbol{GV} = \boldsymbol{I}$).

Resultant Matrix Form

$$N\left\{\overbrace{\begin{pmatrix} G_{s1} & -G_{s1} & 0 & 0 & \cdots & 0 \\ G_{s1} & -(G_{s1}+G_{s2}+G_{g1}) & G_{s2} & 0 & \cdots & 0 \\ 0 & G_{s2} & -(G_{s2}+G_{s3}+G_{g2}) & G_{s3} & \cdots & 0 \\ \vdots & \ddots & & & & \vdots \\ 0 & \cdots & 0 & 0 & G_{sN-1} & -(G_{sN-1}+G_{gN-1}) \end{pmatrix}}^{N}\right.$$

$$\times \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{pmatrix} = \begin{pmatrix} I_s \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Now that we have matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, we can implement the solution methods for $\boldsymbol{x}$.

Methods Used

Reference method:

- Exact solution using MATLAB

Experimental method:

- Jacobi iteration

METHODS

The Exact Solution

MATLAB provides a direct solution to $\boldsymbol{x}$ given $\boldsymbol{A}$ and $\boldsymbol{b}$. The command for this is:
`x = A\b.`

We can use this to compare to our experimental method.

Jacobi Method

- Solves for $\boldsymbol{x}$ iteratively
- Requires computation of the diagonal $\boldsymbol{D}$ and remainder $\boldsymbol{R}$ of $\boldsymbol{A}$
- Only works on square matrices that are strictly diagonally dominant
- The spectral radius of the iteration matrix must be less than 1 to guarantee convergence. This is defined as

$$\rho(D^{-1}R) = \max(|\lambda_1|, \ldots, |\lambda_N|)$$

I chose this method because MATLAB provides built-in functions for obtaining these parameters. Also, the system satisfies all the necessary conditions.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

$$A = D + R \qquad \text{where} \qquad D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{ and } R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}.$$

From Wikipedia

Method Verification

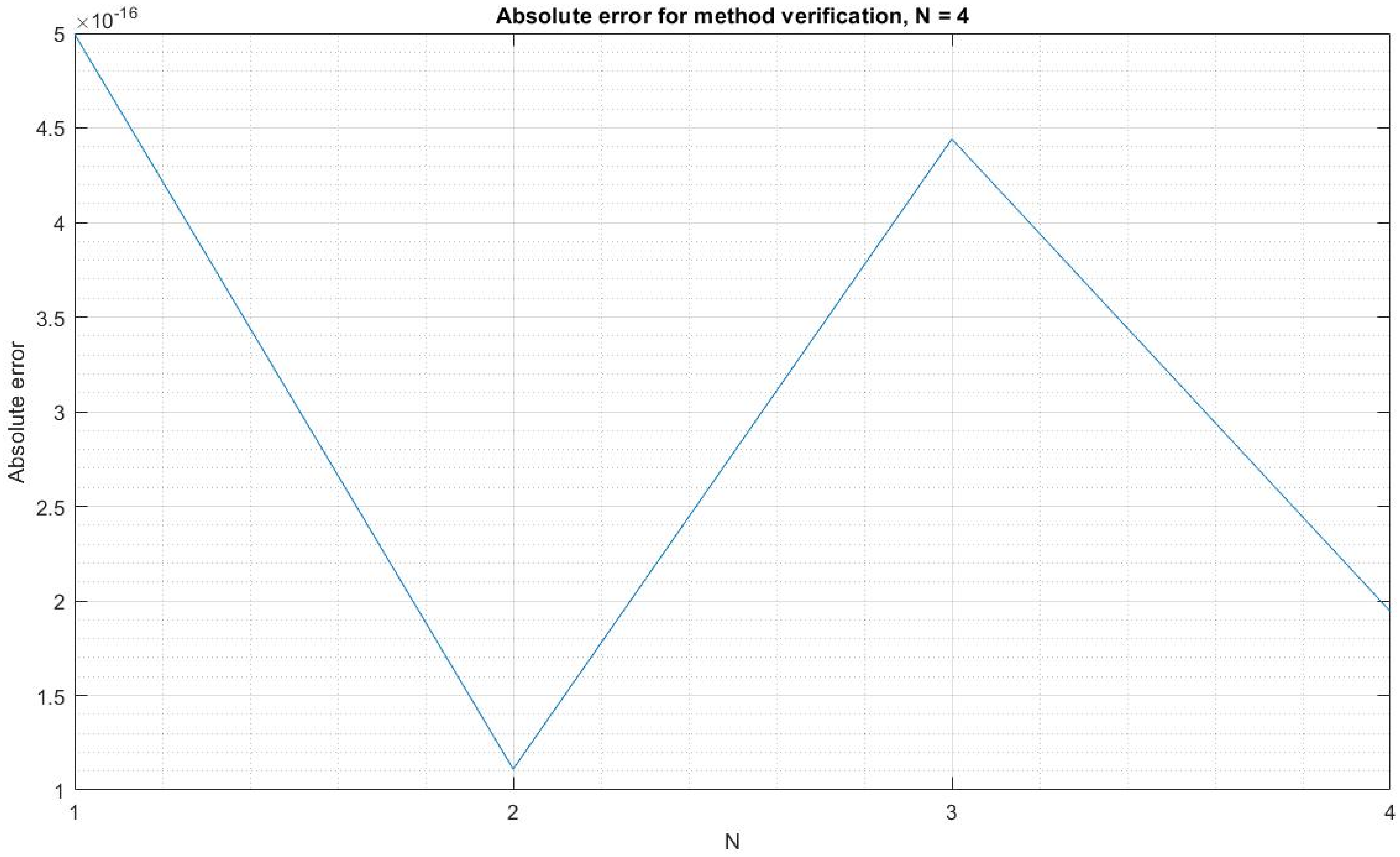
MATLAB provides tools for generating random values for matrices and vectors.

Setting $\boldsymbol{A}$ to a random $N \times N$ matrix and $\boldsymbol{b}$ to a random column vector of size N, we can solve for $\boldsymbol{x}$ using the Jacobi method and then compare this result to the exact solution.

This can be run continuously until a system that meets the spectral radius requirements is found.

I set up the Jacobi method to iterate 500 times for high accuracy.

The absolute error ($|reference - experimental|$) for a particular system with $N = 4$ is plotted below. Since the error is on a magnitude of 10^{-16}, I can conclude that my implementation is correct.



Absolute error for method verification, N = 4
×10^-16
5
4.5
4
3.5
3
2.5
2
1.5
1
Absolute error
1
2
3
4
N

SOLUTIONS: CASE 1 AND CASE 2

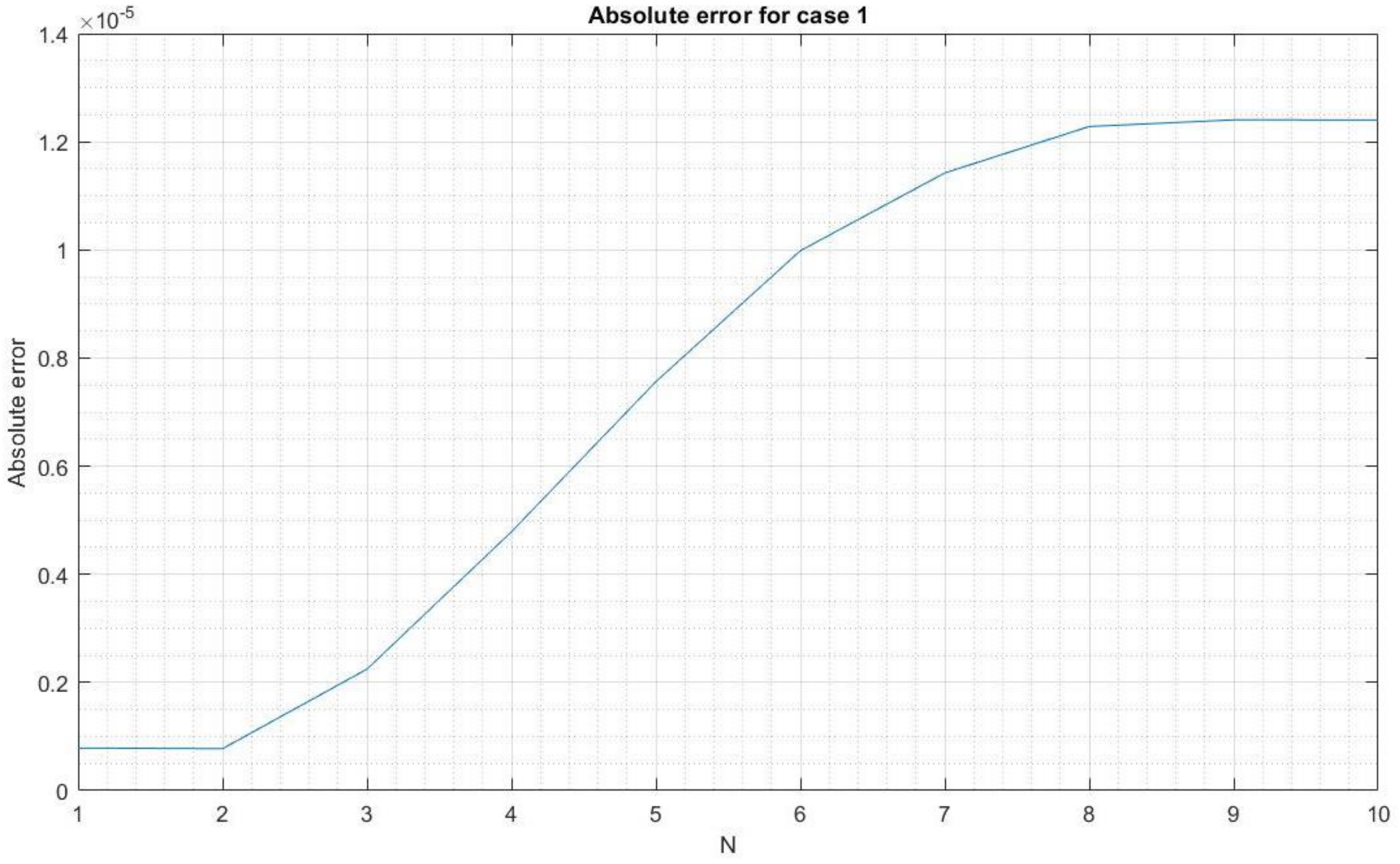
Case 1

The Jacobi method was set to iterate 500 times, which provided results that were almost exactly equal to the actual solutions.

The solutions for $V_1, \ldots, V_{10}$ are given here.

The absolute error is plotted below.

Voltage (V)
869.4368
69.4368
8.3104
1.3392
0.2659
0.0641
0.0217
0.0124
0.0105
0.0102



Absolute error for case 1
×10^-5
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Absolute error
1
2
3
4
5
6
7
8
9
10
N

Case 2

The golden ratio is approximately:

$$\frac{1+\sqrt{5}}{2} = 1.618$$

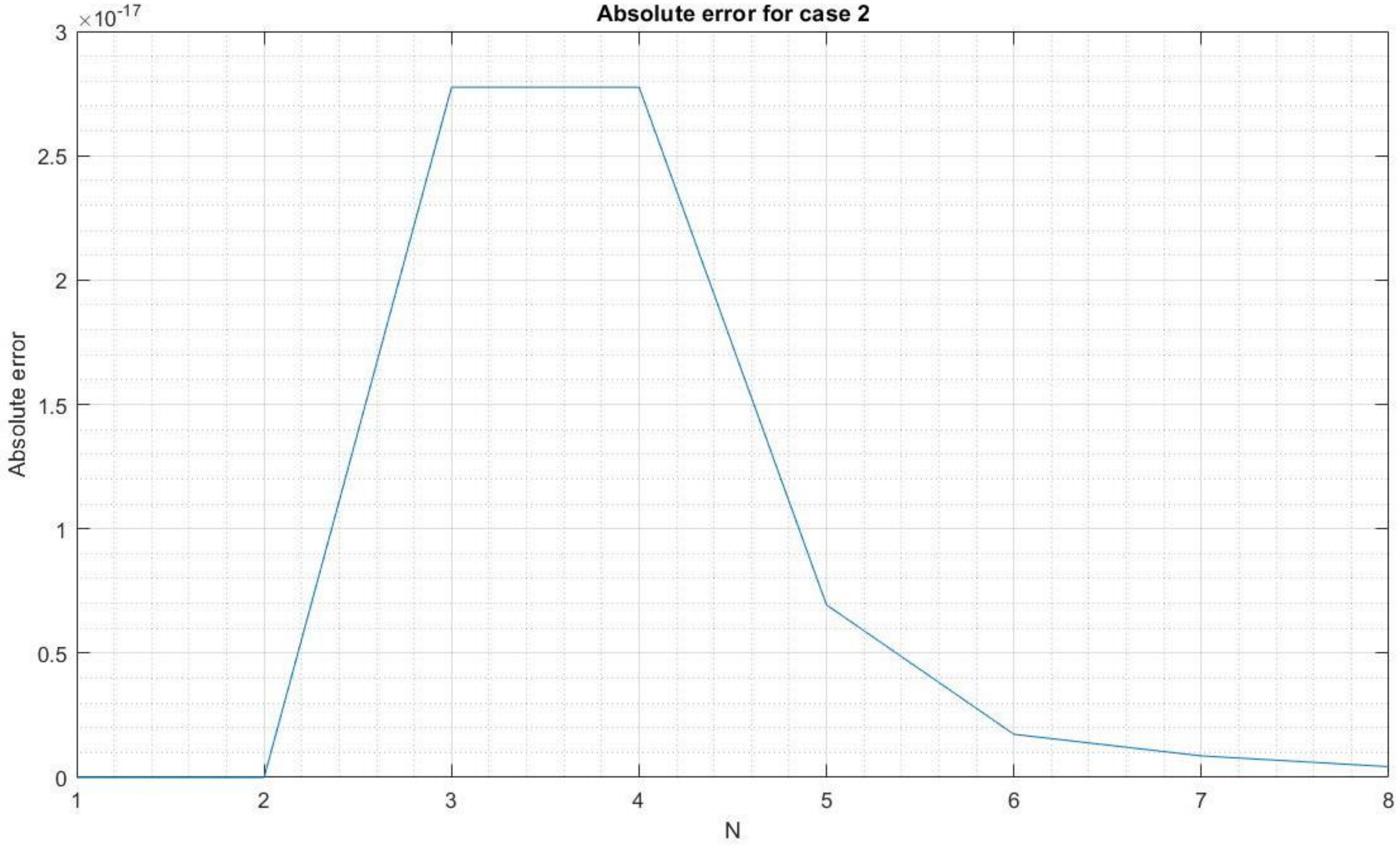
I set all resistance values to $R = 1\ \Omega$ for simplicity.

After some experimentation, I found that a value of $N = 8$ provided a close enough value to the golden ratio.

The solutions for $V_1, \ldots, V_8$ are given here.

The absolute error is plotted below.

Voltage (V)
1.6180
0.6180
0.2361
0.0902
0.0345
0.0133
0.0053
0.0027


Absolute error for case 2
×10^-17
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
N
Absolute error

CONCLUSIONS

Analysis and Conclusions

From these results, we can see that the Jacobi method provides very accurate solutions when it is set up to iterate many times.

- However, the conditions (square matrix, spectral radius, etc.) are strict. If they are not met, the results could be very inaccurate or the method could fail entirely
- Because this particular problem satisfies all conditions, no errors were met
- But there are many conditions where my code fails to solve the problem. Given a system with either:
  - *A non-square matrix*
  - *An iteration matrix with a spectral radius of greater than 1*
  - *A matrix that is not strictly diagonally dominant*

  the Jacobi method could fail to give an accurate result