CS 138 - Numerical Analysis II

Problem Set 1: Lecture Problems + Feedback

Carmelo Ellezandro Atienza
CS 138 WFW
2021-08090

Arturo Miguel Saquilayan
CS 138 WFY
2021-04603

1. Roll Call

1.1. Modeling the Problem

Let x_1, x_2, x_3 and x_4 represent a Normal Zombie, Conehead Zombie, Ducky Tube Zombie, and Conehead Ducky Tube Zombie. We form our system of linear equations as follows.

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 32 \\ x_2 + x_4 = 11 \\ x_3 + x_4 = 16 \\ x_1 - 3x_2 = 0 \end{cases}$$

We can then model $Ax = b$ as follows, $\boxed{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & -3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 32 \\ 11 \\ 16 \\ 0 \end{bmatrix}}$.

1.2. Norms

Computing the 1-norm n_1, $\boxed{n_1 = max(1+3, 1+1+3, 1+1, 1+1+1) = 5}$.

Computing the ∞-norm n_∞, $\boxed{n_\infty = max(1+1+1+1, 1+1, 1+1, 1+3) = 4}$.

1.3. Standard Gaussian Elimination

$$\left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 32 \\ 0 & 1 & 0 & 1 & 11 \\ 0 & 0 & 1 & 1 & 16 \\ 1 & -3 & 0 & 0 & 0 \end{array}\right] \overset{-R_1+R_4\to R_4}{\Longrightarrow} \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 32 \\ 0 & 1 & 0 & 1 & 11 \\ 0 & 0 & 1 & 1 & 16 \\ 0 & -4 & -1 & -1 & -32 \end{array}\right] \overset{4R_2+R_4\to R_4}{\Longrightarrow} \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 32 \\ 0 & 1 & 0 & 1 & 11 \\ 0 & 0 & 1 & 1 & 16 \\ 0 & 0 & -1 & 3 & 12 \end{array}\right] \overset{R_3+R_4\to R_4}{\Longrightarrow}$$

$$\left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 32 \\ 0 & 1 & 0 & 1 & 11 \\ 0 & 0 & 1 & 1 & 16 \\ 0 & 0 & 0 & 4 & 28 \end{array}\right] \Longrightarrow \begin{cases} x_1 + x_2 + x_3 + x_4 = 32 \\ x_2 + x_4 = 11 \\ x_3 + x_4 = 16 \\ 4x_4 = 28 \end{cases} \Longrightarrow \begin{cases} x_1 = 12 \\ x_2 = 4 \\ x_3 = 9 \\ x_4 = 7 \end{cases} \Longrightarrow$$

$$\boxed{x = \begin{bmatrix} 12 \\ 4 \\ 9 \\ 7 \end{bmatrix}}$$

1.4. LU Factorization

Based on the previous item, we give the Doolittle Decomposition of the matrix A based on the elementary row operations performed. Thus, we yield,

$$\boxed{L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & -4 & -1 & 1 \end{bmatrix}, U = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix}}$$

)

1.5. Substitution with LU

Given L and U from the previous item, we compute for the solution via Forward Substitution with $Ly = b$ and Backward Substitution with $Ux = y$.

For $Ly = b$, given $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & -4 & -1 & 1 \end{bmatrix}$, and $b = \begin{bmatrix} 27 \\ 6 \\ 15 \\ 0 \end{bmatrix}$, we obtain vector y as follows,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & -4 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 27 \\ 6 \\ 15 \\ 0 \end{bmatrix} \implies \begin{cases} y_1 = 27 \\ y_2 = 6 \\ y_3 = 15 \\ y_1 - 4y_2 - y_3 + y_4 = 0 \end{cases}$$

Solving for y_4,

$$\begin{gathered} y_1 - 4y_2 - y_3 + y_4 = 0 \\ 27 - 4(6) - 15 + y_4 = 0 \\ -12 + y_4 = 0 \\ y_4 = 12 \end{gathered}$$

Thus,

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 27 \\ 6 \\ 15 \\ 12 \end{bmatrix}$$

For $Ux = y$, given $U = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix}$, we obtain the solution x as follows,

$$U = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 27 \\ 6 \\ 15 \\ 12 \end{bmatrix} \implies \begin{cases} x_1 + x_2 + x_3 + x_4 = 27 \\ x_2 + x_4 = 6 \\ x_3 + x_4 = 15 \\ x_4 = 3 \end{cases}$$

Solving for x_2 and x_3,

$$\begin{gathered} x_2 + x_4 = 6 \Rightarrow x_2 + 3 = 6 \Rightarrow x_2 = 3 \\ x_3 + x_4 = 15 \Rightarrow x_3 + 3 = 15 \Rightarrow x_3 = 12 \end{gathered}$$

Solving for x_1,

$$x_1 + x_2 + x_3 + x_4 = 27$$
$$x_1 + 3 + 12 + 3 = 27$$
$$x_1 + 18 = 27$$
$$x_1 = 9$$

Thus, our solution is

$$\boxed{x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 9 \\ 3 \\ 12 \\ 3 \end{bmatrix}}$$

2. Threepeater Tally

2.1. What the β

Given the definition of a banded matrix B, where $B_{\{ij\}} = 0$ if $|i - j| > \beta$, we observe that the only nonzero elements of a banded matrix is along the main diagonal of B, as well as the elements that are above or below the main diagonal of B.

For matrix A, we claim that $\beta = 1$. Along the diagonal entries, $i = j \Rightarrow |i - j| = 0$. So, the diagonal entries are nonzero. Along the entries adjacent to the diagonal (top or bottom), $|i - j| = 1$. Hence, the adjacent entries to the diagonal are nonzero. For any other entries, where $|i - j| > 1$, the entries are zero. Therefore, $\boxed{\beta = 1}$.

2.2. Sparsity for $n = 5$

Consider the 5×5 banded matrix A,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

We can count that the nonzero entries are the diagonal entries, the entries adjacent to the top of the diagonal, and the entries adjacent to the bottom the diagonal (for brevity, we call these the top and bottom entries, respectively). Counting, we get 5 diagonal entries, 4 top entries, and 4 bottom entries. Thus, we have $5 + 4 + 4 = 13$ nonzero entries. For a 5×5 matrix, the total number of entries is 25. Computing for sparsity(A) if $n = 5$,

$$\boxed{\text{sparsity(A)} = \frac{13}{25} = 0.52}$$

2.3. Sparsity for Arbitrary n

For arbitrary n, note that the number of diagonal entries is n, while the number of top and bottom entries is $n - 1$. Hence, sparsity(A) for arbitrary n is computed as follows,

$$\text{sparsity(A)} = \frac{n + (n-1) + (n-1)}{n^2}$$
$$= \frac{3n-2}{n^2}$$

Therefore, for an arbitrary n, $\boxed{\text{sparsity(A)} = \frac{3n-2}{n^2}}$.

2.4. Sparsity in Terms of n and β

A banded matrix A of size $n \times n$ with bandwidth β has nonzero entries that are only within the band as defined by β. Meaning, the nonzero entries appear at most β rows above and β rows below a given diagonal element.

2.4.1. Counting Nonzero Elements: Boundary Case

We first consider the starting edge case. For columns $j = 1, 2, ..., \beta$ i.e. the first β columns, some entries above the diagonal will not exist. We can compute the starting edge case nonzero elements z_s as follows,

$$z_s = \sum_{i=1}^{\beta} = \beta + i$$

In fact, the same applies for the ending edge case. For columns $j = n - \beta + 1, n - \beta + 2, ..., n$, i.e. the last β columns, the ending edge case nonzero elements z_e is functionally equivalent to,

$$z_e = z_s = \sum_{i=1}^{\beta} = \beta + i$$

Thus, the number of nonzero elements for the boundary case z_b is essentially,

$$z_b = z_s + z_e = 2\left(\sum_{i=1}^{\beta} = \beta + i\right)$$

2.4.2. Counting Nonzero Elements: General Case

In general, the nonzero elements for columns $j = \beta + 1, \beta + 2, ..., n - \beta$ can be computed by,

$$z_g = \sum_{i=\beta+1}^{n-\beta} = 2\beta + 1$$

where $2\beta + 1$ bands the maximum amount of nonzero elements for a given column.

2.4.3. Final Answer

Piecing it together, the maximum sparsity of a banded $n \times n$ matrix S with bandwidth β is,

$$\boxed{\text{sparsity(S)} = \frac{z_b + z_g}{n^2} = \frac{2\left(\sum_{i=1}^{\beta} = \beta + i\right) + \sum_{i=\beta+1}^{n-\beta} = 2\beta + 1}{n^2}}$$

2.5. Gaussian Elimination for $n = 6$

$$\left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 1&1&1&0&0&0&10\\ 0&1&1&1&0&0&10\\ 0&0&1&1&1&0&10\\ 0&0&0&1&1&1&12\\ 0&0&0&0&1&1&11 \end{array}\right] \overset{-R_1+R_2\to R_2}{\Longrightarrow} \left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 0&0&1&0&0&0&6\\ 0&1&1&1&0&0&10\\ 0&0&1&1&1&0&10\\ 0&0&0&1&1&1&12\\ 0&0&0&0&1&1&11 \end{array}\right] \overset{R_2\leftrightarrow R_3}{\Longrightarrow} \left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 0&1&1&1&0&0&10\\ 0&0&1&0&0&0&6\\ 0&0&1&1&1&0&10\\ 0&0&0&1&1&1&12\\ 0&0&0&0&1&1&11 \end{array}\right] \overset{-R_3+R_4\to R_4}{\Longrightarrow}$$

$$\left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 0&1&1&1&0&0&10\\ 0&0&1&0&0&0&6\\ 0&0&0&1&1&0&4\\ 0&0&0&1&1&1&12\\ 0&0&0&0&1&1&11 \end{array}\right] \overset{-R_4+R_5\to R_5}{\Longrightarrow} \left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 0&1&1&1&0&0&10\\ 0&0&1&0&0&0&6\\ 0&0&0&1&1&0&4\\ 0&0&0&0&0&1&8\\ 0&0&0&0&1&1&11 \end{array}\right] \overset{R_5\leftrightarrow R_6}{\Longrightarrow} \left[\begin{array}{cccccc|c} 1&1&0&0&0&0&4\\ 0&1&1&1&0&0&10\\ 0&0&1&0&0&0&6\\ 0&0&0&1&1&0&4\\ 0&0&0&0&1&1&11\\ 0&0&0&0&0&1&8 \end{array}\right] \Longrightarrow$$

$$\begin{cases} x_1 + x_2 = 4 \\ x_2 + x_3 + x_4 = 10 \\ x_3 = 6 \\ x_4 + x_5 = 4 \\ x_5 + x_6 = 11 \\ x_6 = 8 \end{cases} \Longrightarrow \begin{cases} x_1 = 1 \\ x_2 = 3 \\ x_3 = 6 \\ x_4 = 1 \\ x_5 = 3 \\ x_6 = 8 \end{cases} \Longrightarrow \boxed{x = \begin{bmatrix} 1\\3\\6\\1\\3\\8 \end{bmatrix}}$$

2.6. LU Factorization for $n = 6$

From the previous item, we have,

$$P = \begin{bmatrix} 1&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0 \end{bmatrix}, A = \begin{bmatrix} 1&1&0&0&0&0\\ 1&1&1&0&0&0\\ 0&1&1&1&0&0\\ 0&0&1&1&1&0\\ 0&0&0&1&1&1\\ 0&0&0&0&1&1 \end{bmatrix} \Longrightarrow PA = \begin{bmatrix} 1&1&0&0&0&0\\ 0&1&1&1&0&0\\ 1&1&1&0&0&0\\ 0&0&1&1&1&0\\ 0&0&0&0&1&1\\ 0&0&0&1&1&1 \end{bmatrix}$$

For PA, we have the LU Decomposition,

$$\boxed{L = \begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 1&0&1&0&0&0\\ 0&0&1&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&1&0&1 \end{bmatrix}, U = \begin{bmatrix} 1&1&0&0&0&0\\ 0&1&1&1&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&1&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{bmatrix}}$$

2.7. LU Factorization for arbitrary n

We want to find the decomposition of PA for an arbitrary n. We can derive some patterns from the previous items and generalize it. Unless otherwise stated, assume that we initialize an $n \times n$ matrix filled with $0'$s for both L and U.

For U, we first fill the main diagonal with 1. That is, for $i = 1, 2, ..., n$, $U_{ii} = 1$. Then, we copy the entries of PA that are **above the diagonal** and place it in the same position in U. That is, for $i = 1, 2, ..., n$ and $j = i + 1, i + 2, ..., n$, $U_{ij} = [PA]_{ij}$.

For L, we also fill the diagonals with 1. That is, for $i = 1, 2, ..., n$, $L_{ii} = 1$. Then, we iterate through columns $j = 1, 2, ..., n$. We define some cases when filling up our lower diagonal elements,

$$\begin{cases} L_{j+2,j} = 1 \text{ if } j \bmod 3 = 1 \\ L_{j+1,j} = 1 \text{ if } j \bmod 3 = 0 \end{cases}$$

Basically, we count in batches of 3. For columns $j \bmod 3 = 1$, i.e. every 1st column, we place a 1 that is one row away from L_{jj}. For columns $j \bmod 3 = 0$, i.e. every 3rd column, we place a 1 that is directly beneath L_{jj}. It must be noted that for columns $j \bmod 2 = 2$, i.e. every 2nd column, we do not modify any lower diagonal entries in the jth column. This algorithm is repeated until we reach the nth column.

2.8. Modified Gaussian For Banded Matrices with $\beta = 1$

We modify the elimination step to only operate on the non-zero elements of A which are within the bandwidth $\beta = 1$.

Our goal is to reduce the banded matrix A to an upper triangular matrix which will allow for efficient back substitution. For each $i = 1, 2, ..., n - 1$, we eliminate the entry below the main diagonal by the following steps,

$$m_i = \frac{A_{i+1,i}}{A_{i,i}}$$

$$A_{i+1,i+1} = A_{i+1,i+1} - m_i i \cdot A_{i,i+1}$$

$$b_{i+1} = b_{i+1} - m_i \times b_i$$

This step only modifies two entries (the one on the diagonal and one off-diagonal) per row, so the number of operations is linear with respect to n.

Then we perform the back substitution. At this point, A should be an upper triangular matrix. Starting from the bottom, for each $j = n, n - 1, ..., 1$, we perform the substitution step,

$$x_j = \begin{cases} \frac{b_n}{A_{j,j}} \text{ if } j = n \\ \frac{b_j - A_{i,i+1} \cdot x_{i+1}}{A_{j,j}} \text{ if } j \neq n \end{cases}$$

This process is also in linear time since we are still doing simple arithmetic operations when computing for each element of the solution vector x.

Thus, we can see the overall time complexity is $O(n)$ by leveraging the unique properties of the banded matrix which allows it to be converted to an upper triangular matrix in linear time. Similarly, the computation of the solution vector is done in linear time due to the properties of the upper triangular matrix which allows for efficient back substitution.

As for the space complexity, since we are modifying the elements of A in-place during the transformation of A to an upper triangular matrix, we did not need to store additional elements apart from the computation of m_i, which should still scale linearly with the the amount of rows n. Thus, this algorithm has a space complexity of $O(n)$ as well.

3. Tall-Nut Bowling

3.1. Solving with Jacobi

We are asked to solve the system of linear equations $M_W x = b$ using the Jacobi Method, where

$$M_W = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 9 \\ 9 \\ 9 \end{bmatrix}, x^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

To start, let us break the system into three equations. Using the matrix M_E, we get:

$$2x_1 + 0x_2 + 1x_3 = 9$$

$$1x_1 + 1x_2 + 1x_3 = 9$$

$$1x_1 + 0x_2 + 2x_3 = 9$$

We can then isolate each x_i:

$$x_1 = \frac{1}{2}(9 - x_3)$$

$$x_2 = 9 - x_1 - x_3$$

$$x_3 = \frac{1}{2}(9 - x_1)$$

Using the initial guess $x^{(0)} = [1 \;\; 1 \;\; 1]^T$:

$$x_1^{(1)} = \frac{1}{2}(9 - 1) = \frac{1}{2} \times 8 = 4$$

$$x_2^{(1)} = 9 - 1 - 1 = 7$$

$$x_3^{(1)} = \frac{1}{2}(9 - 1) = \frac{1}{2} \times 8 = 4$$

Thus, $x^{(1)} = [4 \;\; 7 \;\; 4]^T$.

Now using $x^{(1)} = [4 \;\; 7 \;\; 4]^T$:

$$x_1^{(2)} = \frac{1}{2}(9 - 4) = \frac{1}{2} \times 5 = 2.5$$

$$x_2^{(2)} = 9 - 4 - 4 = 1$$

$$x_3^{(2)} = \frac{1}{2}(9 - 4) = \frac{1}{2} \times 5 = 2.5$$

Thus, $x^{(1)} = [2.5 \;\; 1 \;\; 2.5]^T$.

Now using $x^{(2)} = [2.5 \;\; 1 \;\; 2.5]^T$:

$$x_1^{(3)} = \frac{1}{2}(9 - 2.5) = \frac{1}{2} \times 6.5 = 3.25$$

$$x_2^{(3)} = 9 - 2.5 - 2.5 = 4$$

$$x_3^{(3)} = \frac{1}{2}(9 - 2.5) = \frac{1}{2} \times 6.5 = 3.25$$

Thus, $x^{(3)} = [3.25 \;\; 4 \;\; 3.25]^T$. We can repeat this process, iterating until x converges on $[3 \;\; 3 \;\; 3]^T$.

3.2. Solving with Gauss-Seidel

We are now asked to solve the system of linear equations $M_E x = b$ using the Gauss-Seidel Method, where

$$M_E = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}, b = \begin{bmatrix} 7 \\ 7 \\ 7 \end{bmatrix}, x^{(0)} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

To start, let us break the system into three equations. Using the matrix M_E, we get:

$$3x_1 + 1x_2 + 0x_3 = 7$$

$$1x_1 + 3x_2 + 1x_3 = 7$$

$$0x_1 + 1x_2 + 3x_3 = 7$$

We can then isolate each x_i:

$$x_1 = \frac{1}{3}(7 - x_2)$$

$$x_2 = \frac{1}{3}(7 - x_1 - x_3)$$

$$x_3 = \frac{1}{3}(7 - x_2)$$

Using the initial guess $x^{(0)} = [2 \;\; 2 \;\; 2]^T$:

$$x_1^{(1)} = \frac{1}{3}(7 - 2) = \frac{1}{3} \times 5 \approx 1.6667$$

$$x_2^{(1)} = \frac{1}{3}(7 - 1.6667 - 2) = \frac{1}{3} \times 3.3333 \approx 1.1111$$

$$x_3^{(1)} = \frac{1}{3}(7 - 1.1111) = \frac{1}{3} \times 5.8889 \approx 1.963$$

Thus, $x^{(1)} \approx [1.6667 \;\; 1.1111 \;\; 1.963]^T$.

Now using $x^{(1)} = [1.6667 \;\; 1.1111 \;\; 1.963]^T$:

$$x_1^{(2)} = \frac{1}{3}(7 - 1.6667) = \frac{1}{3} \times 8.3333 \approx 1.963$$

$$x_2^{(2)} = \frac{1}{3}(7 - 1.963 - 1.963) = \frac{1}{3} \times 3.074 \approx 1.0247$$

$$x_3^{(2)} = \frac{1}{3}(7 - 1.0247) = \frac{1}{3} \times 5.9753 \approx 1.9918$$

Thus, $x^{(2)} \approx \begin{bmatrix} 1.963 & 1.0247 & 1.9918 \end{bmatrix}^T$.

Now using $x^{(2)} = \begin{bmatrix} 1.963 & 1.0247 & 1.9918 \end{bmatrix}^T$:

$$x_1^{(3)} = \frac{1}{3}(7 - 1.0247) = \frac{1}{3} \times 5.9753 \approx 1.9918$$

$$x_2^{(3)} = \frac{1}{3}(7 - 1.9918 - 1.9918) = \frac{1}{3} \times 3.0164 \approx 1.0055$$

$$x_3^{(3)} = \frac{1}{3}(7 - 1.0055) = \frac{1}{3} \times 5.9945 \approx 1.9982$$

Thus, $x^{(3)} \approx \begin{bmatrix} 1.9918 & 1.0055 & 1.9982 \end{bmatrix}^T$. We can repeat this process, iterating until x converges on $\begin{bmatrix} 2 & 1 & 2 \end{bmatrix}^T$.

3.3. Solving using Successive Over-Relaxation

We are now asked to solve the system of linear equations $M_T x = b$ using Successive Over-Relaxation, where

$$M_T = \begin{bmatrix} 2 & 2 & 0 \\ 1 & 3 & 1 \\ 0 & 2 & 2 \end{bmatrix}, b = \begin{bmatrix} 10 \\ 15 \\ 4 \end{bmatrix}, x^{(0)} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}, \omega = 2$$

To start, let us break the system into three equations. Using the matrix M_E, we get:

$$2x_1 + 2x_2 + 0x_3 = 10$$

$$1x_1 + 3x_2 + 1x_3 = 15$$

$$0x_1 + 2x_2 + 2x_3 = 14$$

We can then isolate each x_i:

$$x_1 = \frac{1}{2}(10 - 2x_2)$$

$$x_2 = \frac{1}{2}(15 - x_1 - x_3)$$

$$x_3 = \frac{1}{2}(14 - 2x_2)$$

Using the initial guess $x^{(0)} = \begin{bmatrix} 3 & 3 & 3 \end{bmatrix}^T$:

$$x_1^{(1)} = (1 - 2) \times 3 + \frac{2}{2}(10 - 2 \times 3) = -3 + (1) \times (10 - 6) = -3 + 4 = 1$$

$$x_2^{(1)} = (1 - 2) \times 3 + \frac{2}{3}(15 - 1 - 3) = -3 + \frac{2}{3} \times 11 = -3 + 7.33 \approx 4.33$$

$$x_3^{(1)} = (1 - 2) \times 3 + \frac{2}{2}(14 - 2 \times 4.33) = -3 + (1) \times (14 - 8.66) = -3 + 5.34 = 2.34$$

Thus, $x^{(1)} \approx \begin{bmatrix} 1 & 4.33 & 2.34 \end{bmatrix}^T$.

Now using $x^{(1)} = \begin{bmatrix} 1 & 4.33 & 2.34 \end{bmatrix}^T$:

$$x_1^{(2)} = (1-2) \times 1 + \frac{2}{2}(10 - 2 \times 4.33) = -1 + (1) \times (10 - 8.66) = -1 + 1.34 = 0.34$$

$$x_2^{(2)} = (1-2) \times 4.33 + \frac{2}{3}(15 - 0.34 - 2.34) = -4.33 + \frac{2}{3} \times 12.32 = -4.33 + 8.21 = 3.88$$

$$x_3^{(2)} = (1-2) \times 2.34 + \frac{2}{2}(14 - 2 \times 3.88) = -2.34 + (1) \times (14 - 7.76) = -2.34 + 6.24 = 3.90$$

Thus, $x^{(2)} \approx \begin{bmatrix} 0.34 & 3.88 & 3.90 \end{bmatrix}^T$.

Now using $x^{(2)} = \begin{bmatrix} 0.34 & 3.88 & 3.90 \end{bmatrix}^T$:

$$x_1^{(3)} = (1-2) \times 0.34 + \frac{2}{2}(10 - 2 \times 3.88) = -0.34 + (1) \times (10 - 7.76) = -0.34 + 2.24 = 1.90$$

$$x_2^{(3)} = (1-2) \times 3.88 + \frac{2}{3}(15 - 1.90 - 3.90) = -3.88 + \frac{2}{3} \times 9.20 = -3.88 + 6.13 = 2.25$$

$$x_3^{(3)} = (1-2) \times 3.90 + \frac{2}{2}(14 - 2 \times 2.25) = -3.90 + (1) \times (14 - 4.50) = -3.90 + 9.50 = 5.60$$

Thus, $x^{(3)} \approx \begin{bmatrix} 1.90 & 2.25 & 5.60 \end{bmatrix}^T$. We can repeat this process, iterating until x converges on $\begin{bmatrix} 2 & 3 & 4 \end{bmatrix}^T$.

3.4. Iteration Matrix under Jacobi

We are now being asked to find the iteration matrix P_W of M_W under Jacobi, and to compute its spectral radius $\rho(P_W)$ using either analytical or numerical methods. Given these, we are also asked to make a conclusion about the convergence of M_W under Jacobi.

First, we decompose M_W as:

- D: the diagonal matrix of M_W,
- L: the strictly lower triangular part of M_W,
- U: the strictly upper triangular part of M_W.

Therefore, for M_W, we have:

$$D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The iteration matrix for the Jacobi method is given by:

$$P_W = -D^{-1}(L + U)$$

We get the inverse of D D^{-1} by getting the reciprocals of its non-zero elements:

$$D^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

We can get $L + U$ by combining the elements of L and U. Hence:

$$L + U = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

We can then compute the iteration matrix $P_W = -D^{-1}(L+U)$

$$P_W = -\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Performing the matrix multiplication:

$$P_W = -\begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 1 & 0 & 1 \\ \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Therefore, the iteration matrix under Jacobi is:

$$P_W = \begin{bmatrix} 0 & 0 & -\frac{1}{2} \\ -1 & 0 & -1 \\ -\frac{1}{2} & 0 & 0 \end{bmatrix}$$

Now, we compute for the spectral radius $\rho(P_W)$ from the largest absolute value of the eigenvalues of P_W. We can either compute the eigenvalues analytically by solving the characteristic equation or compute them numerically. For simplicity, let us compute them numerically.

Using numerical methods (for this instance, power and inverse iteration was used), we find that the largest and smallest eigenvalues of P_W are approximately:

$$\lambda = \pm\frac{1}{2}$$

Thus, we conclude that the spectral radius is:

$$\rho(P_W) = \max(|\lambda_1|, |\lambda_2|) \approx \frac{1}{2}$$

As the spectral radius $\rho(P_W)$ under Jacobi is $\frac{1}{2}$, which is less than 1, this implies that the Jacobi method **will converge** when solving the system $M_W x = b$.

3.5. Iteration Matrix under Gauss-Seidel

For the last part of this problem, we are now asked to find the iteration matrix P_E of M_E under Gauss-Seidel, and to compute its spectral radius $\rho(P_E)$ using either analytical or numerical methods. Given these, we are also asked to make a conclusion about the convergence of M_E under Gauss-Seidel.

First, we decompose M_E as:
- D: the diagonal matrix of M_E,
- L: the strictly lower triangular part of M_E,
- U: the strictly upper triangular part of M_E.

Therefore, for M_W, we have:

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}, L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The iteration matrix for the Jacobi method is given by:

$$P_E = -(D+L)^{-1}U$$

We get $D + L$ by combining the elements of D and L. Hence:

$$D + L = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 1 & 3 \end{bmatrix}$$

We get $(D+L)^{-1}$ by performing matrix inversion:

$$(D+L)^{-1} = \begin{bmatrix} \frac{1}{3} & 0 & 1 \\ -\frac{1}{9} & \frac{1}{3} & 1 \\ \frac{1}{27} & -\frac{1}{9} & \frac{1}{3} \end{bmatrix}$$

We can then compute the iteration matrix $P_E = -(D+L)^{-1}U$ by performing the matrix multiplication:

$$P_E = \begin{bmatrix} \frac{1}{3} & 0 & 1 \\ -\frac{1}{9} & \frac{1}{3} & 1 \\ \frac{1}{27} & -\frac{1}{9} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore, the iteration matrix under Gauss-Seidel is:

$$P_E = \begin{bmatrix} 0 & \frac{1}{3} & 0 \\ 0 & -\frac{1}{9} & \frac{1}{3} \\ 0 & \frac{1}{27} & -\frac{1}{9} \end{bmatrix}$$

Now, we compute for the spectral radius $\rho(P_E)$ from the largest absolute value of the eigenvalues of P_E. We can either compute the eigenvalues analytically by solving the characteristic equation or compute them numerically. For simplicity, let us also compute them numerically.

Using numerical methods (for this instance, power and inverse iteration was used), we find that the largest and smallest eigenvalues of P_E are approximately:

$$\lambda_1 = 0, \lambda_2 = -\frac{2}{9}$$

Thus, we can now conclude that the spectral radius is:

$$\rho(P_E) = \max(|\lambda_1|, |\lambda_2|) \approx \frac{2}{9}$$

As the spectral radius $\rho(P_E)$ under Gauss-Seidel is $\frac{2}{9}$, which is less than 1, this implies that the Gauss-Seidel method **will converge** when solving the system $M_E x = b$.

4. Garden Goodbyes

4.1. What did you think of PS 0 and PS 1?

Melo: PS 0 was inspiring and it laid out some good expectations for the course. PS 1 was a pretty heavy requirement that really lived up to the infamous nature of CS 138. However, I thought that there were lots of interesting ideas that I picked up in this Problem Set 1. The banded matrices were pretty interesting to tackle, and solving **Threepeater Tally** made me do a lot of researching for the properties of these matrices. I also saw how the aforementioned directly helps out with the **Last Stand**, as it also dealt with banded matrices (to the ungodly extremes). Also, I like the PvZ theme.

Art: Well, I do agree with Melo that PS 0 laid out the course expectations pretty well. PS 1 was very sobering given the nature of our previous problem sets in CS 136. I think that all the questions in this problem set were fairly challenging, and I expect that all the future assessments will be just as, if not more difficult as this one. The PvZ theme was interesting and it took some of the mental load off thinking of the numbers as data within a game.

4.2. What did you think about the difficulty of the problems?

Melo: The straightforward matrix solving problems were generally okay. The more theoretical and computational items such as **Threepeater Tally** and **Last Stand** were very difficult. **Threepeater Tally** required some proving and intuition that is beyond to what was taught in the classroom. The **Last Stand** required a supercomputer if you did not optimize your solution, i.e. just calling `np.linalg(A,b)`. I would give PS 1 an 8/10 difficulty.

Art: I feel like this problem set is sort of a mixed bag between easy, but tedious to solve problems, and problems that are just straight up hard and tedious. If we were talking on a 10-point scale, it would probably also be an 8/10.

4.3. Which problems were your favorites? Which problems were your least favorites? Why?

Melo: My favorite problems were **Roll Call** because it was relatively easy. My least favorites were **Threepeater Tally** and **Last Stand** because of the aformentioned.

Art: I think I liked **Flower Arrangement** since it was more focused on letting you explore the ideas on how to complete the specifications and less about getting the answers themselves. I guess my least favorite would be **Last Stand** as it was a bit too difficult for my capabilities right now.

4.4. Do you think the problems will help you with the upcoming exam?

Melo: I think so. There's a reputation that the Numerical Analysis series includes some items that challenge your general understanding of the subject instead of just doing an algorithmic solution. This definitely challenged my understanding of linear algebra.

Art: For sure. It would be kind of mean if you weren't able to apply anything you learned from the problem set in the exam. Like Melo said, there will probably be a lot of questions that are beyond just computing for the answer given the formulas.

4.5. Is there anything you would suggest changing or improving on regarding any aspect of the course so far?

Melo: Perhaps announce the deadline for the Problem Set as soon as it gets released.

Art: Maybe it would be easier to break up the Problem Set into smaller deadlines instead of one long deadline so that students would be incentivized to read up on specific lessons along the lectures to solve the problems.

4.6. If you were under Ma'am Nestine in CS 136 2023.2, do you think your feedback from Feedback Fishing has been adequately addressed (if you provided any points of action to begin with)?

Melo: I was under Sir Z for my CS 136. lol

Art: Sorry, I didn't take CS 136 2023.2 under Ma'am Nestine.

4.7. Anything else you want to say?

Melo: I really do appreciate the difficult problems even if they were incredibly challenging. It gave me some awesome Eureka Moments. That said, please don't make them any harder (make them easier jk).

Art: I appreciate the effort in creating problems around a theme and trying to make them fit the capabilities of the students. Although, we might be overestimating what students can solve reliably on their own by a bit.