Linear Systems – Iterative methods

1. Jacobi Method
2. Gauss-Siedel Method

Iterative Methods

Iterative methods can be expressed in the general form: $x^{(k)} = F(x^{(k-1)})$

where s s.t. $F(s)=s$ is called a Fixed Point

Hopefully: $x^{(k)} \rightarrow s$ (solution of my problem)

- Will it converge? How rapidly?

Iterative Methods

Stationary:

$$x^{(k+1)} = Gx^{(k)} + c$$

where G and c do not depend on iteration count (k)

Non Stationary:

$$x^{(k+1)} = x^{(k)} + a_k p^{(k)}$$

where computation involves information that change at each iteration

Iterative – Stationary Jacobi

In the *i-th* equation solve for the value of x_i while assuming the other entries of x remain fixed:

$$\sum_{j=1}^{N} m_{ij} x_j = b_i \rightarrow x_i = \frac{b_i - \sum_{j \neq i} m_{ij} x_j}{m_{ii}} \quad \Rightarrow \quad x_i^{(k)} = \frac{b_i - \sum_{j \neq i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes: $x^{(k)} = D^{-1} \; L + U \; x^{k-1} + D^{-1} b$

where D, L and U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

Iterative – Stationary Gauss-Seidel

Like Jacobi, but now assume that previously computed results are used as soon as they are available:

$$\sum_{j=1}^{N} m_{ij} x_j = b_i \rightarrow x_i = \frac{b_i - \sum_{j \neq i} m_{ij} x_j}{m_{ii}} \quad \Rightarrow \quad x_i^{(k)} = \frac{b_i - \sum_{j<i} m_{ij} x_j^{(k)} - \sum_{j>i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes: $x^{(k)} = D - L\ ^{-1}(Ux^{k-1} + b)$

where D, L and U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

Iterative – Stationary
Successive Overrelaxation (SOR)

Devised by extrapolation applied to Gauss-Seidel in the form of weighted average:

$$x_i^{(k)} = \omega \bar{x}_i^{(k)} + (1-\omega) x_i^{(k-1)} \qquad x_i^{(k)} = \frac{b_i - \sum_{j<i} m_{ij} x_j^{(k)} - \sum_{j>i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes:

$$x^{(k)} = \; D - wL \;^{-1} (wU + (1-w)D) x^{\;k-1} \; + w(D - wL)^{-1} b$$

where D, L and U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

ω is chosen to increase convergence

Jacobi iteration

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

$$
x^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_n^0 \end{bmatrix}
$$

$$
x_1^1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2^0 - \cdots - a_{1n}x_n^0)
$$

$$
x_2^1 = \frac{1}{a_{22}}(b_2 - a_{21}x_1^0 - a_{23}x_3^0 - \cdots - a_{2n}x_n^0)
$$

$$
x_n^1 = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^0 - a_{n2}x_2^0 - \cdots - a_{nn-1}x_{n-1}^0)
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^{n} a_{ij}x_j^k\right]
$$

Gauss-Seidel (GS) iteration

Use the latest update

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

$$
x^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_n^0 \end{bmatrix}
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^{k} \right]
$$

$$
x_1^1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2^0 - \cdots - a_{1n}x_n^0)
$$

$$
x_2^1 = \frac{1}{a_{22}}(b_2 - a_{21}x_1^1 - a_{23}x_3^0 - \cdots - a_{2n}x_n^0)
$$

$$
x_n^1 = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^1 - a_{n2}x_2^1 - \cdots - a_{nn-1}x_{n-1}^1)
$$

Gauss-Seidel Method

An ***iterative*** method.

Basic Procedure:

-Algebraically solve each linear equation for x_i

-Assume an initial guess solution array

-Solve for each x_i and repeat

-Use absolute relative approximate error after each iteration to check if error is within a pre-specified tolerance.

Gauss-Seidel Method

Why?

The Gauss-Seidel Method allows the user to control round-off error.

Elimination methods such as Gaussian Elimination and LU Decomposition are prone to prone to round-off error.

Also: If the physics of the problem are understood, a close initial guess can be made, decreasing the number of iterations needed.

Gauss-Seidel Method

Algorithm

A set of *n* equations and *n* unknowns:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

$$\vdots \qquad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n$$

If: the diagonal elements are non-zero

Rewrite each equation solving for the corresponding unknown

ex:

First equation, solve for x_1

Second equation, solve for x_2

Gauss-Seidel Method

Algorithm

Rewriting each equation

$$x_1 = \frac{c_1 - a_{12}x_2 - a_{13}x_3 \ldots\ldots - a_{1n}x_n}{a_{11}}$$ ← From Equation 1

$$x_2 = \frac{c_2 - a_{21}x_1 - a_{23}x_3 \ldots\ldots - a_{2n}x_n}{a_{22}}$$ ← From equation 2

$$\vdots \qquad \vdots \qquad \vdots$$

$$x_{n-1} = \frac{c_{n-1} - a_{n-1,1}x_1 - a_{n-1,2}x_2 \ldots\ldots - a_{n-1,n-2}x_{n-2} - a_{n-1,n}x_n}{a_{n-1,n-1}}$$ ← From equation n-1

$$x_n = \frac{c_n - a_{n1}x_1 - a_{n2}x_2 - \ldots\ldots - a_{n,n-1}x_{n-1}}{a_{nn}}$$ ← From equation n

Gauss-Seidel Method

Algorithm

General Form of each equation

$$x_1 = \frac{c_1 - \sum\limits_{\substack{j=1 \\ j\neq 1}}^{n} a_{1j}x_j}{a_{11}}$$

$$x_{n-1} = \frac{c_{n-1} - \sum\limits_{\substack{j=1 \\ j\neq n-1}}^{n} a_{n-1,j}x_j}{a_{n-1,n-1}}$$

$$x_2 = \frac{c_2 - \sum\limits_{\substack{j=1 \\ j\neq 2}}^{n} a_{2j}x_j}{a_{22}}$$

$$x_n = \frac{c_n - \sum\limits_{\substack{j=1 \\ j\neq n}}^{n} a_{nj}x_j}{a_{nn}}$$

Gauss-Seidel Method

Algorithm

General Form for any row 'i'

$$x_i = \frac{c_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j}{a_{ii}}, i = 1, 2, \ldots, n.$$

How or where can this equation be used?

Gauss-Seidel Method

Solve for the unknowns

Assume an initial guess for [X]

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$$

Use rewritten equations to solve for each value of x_i.

Important: **Remember to use the most recent value of x_i**. Which means to apply values calculated to the calculations remaining in the **current** iteration.

Gauss-Seidel Method

Calculate the Absolute Relative Approximate Error

$$\left|\in_a\right|_i = \left|\frac{x_i^{new} - x_i^{old}}{x_i^{new}}\right| \times 100$$

So when has the answer been found?

The iterations are stopped when the absolute relative approximate error is less than a prespecified tolerance for all unknowns.

Suppose that for conciseness we limit ourselves to a 3X3 set of equations.

$$x_1^j = \frac{b_1 - a_{12}x_2^{j-1} - a_{13}x_3^{j-1}}{a_{11}} \qquad \text{(a)}$$

$$x_2^j = \frac{b_2 - a_{21}x_1^{j} - a_{23}x_3^{j-1}}{a_{22}} \qquad \text{(b)}$$

$$x_3^j = \frac{b_3 - a_{31}x_1^{j} - a_{32}x_2^{j}}{a_{33}} \qquad \text{(c)}$$

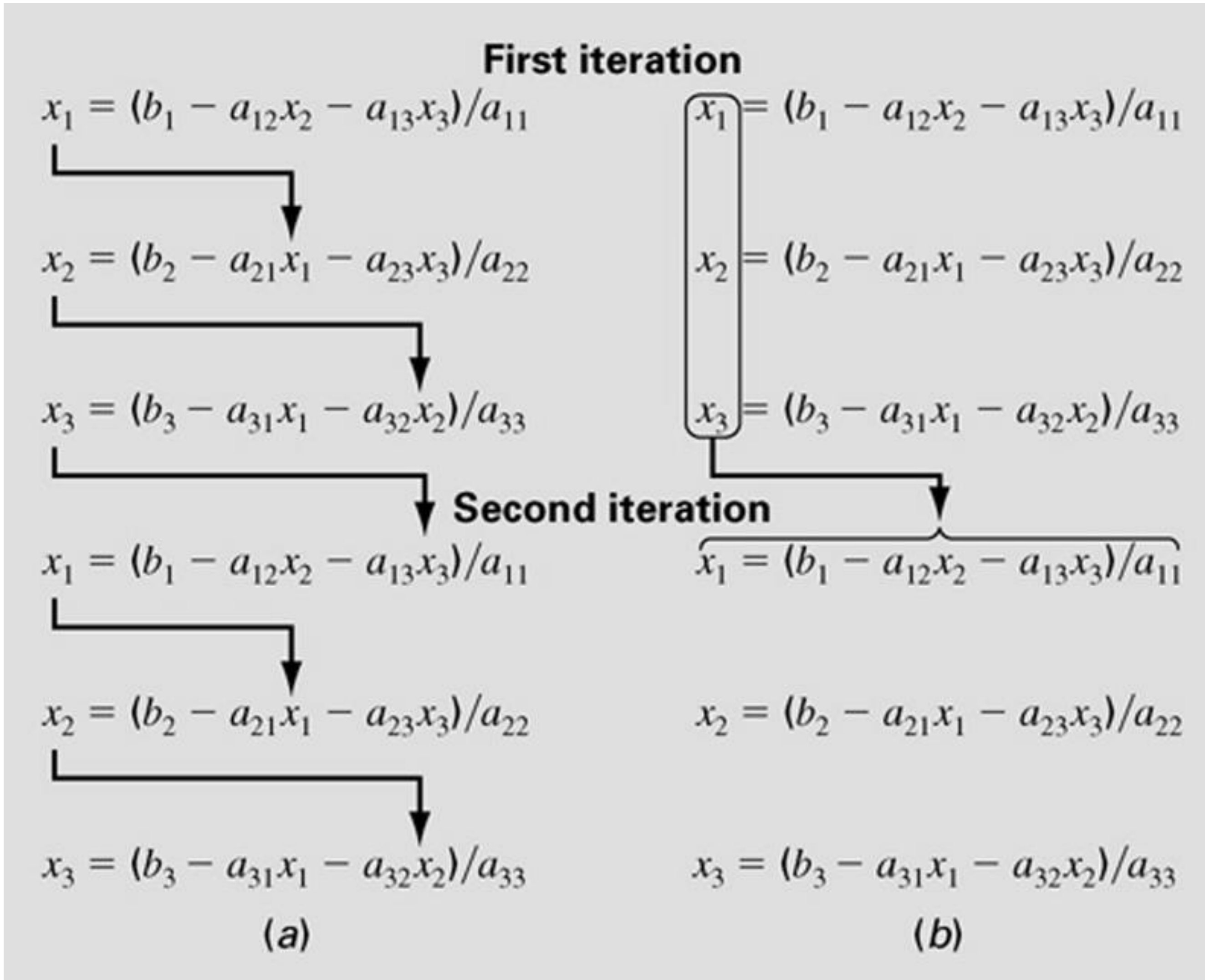
where j and j -1 are the present and previous iterations.

To start the solution process, initial guesses must be made for the x's. A simple approach is to assume that they are all zero.

Convergence can be checked using the criterion that for *i*,

$$\varepsilon_{a,i} = \left| \frac{x_i^j - x_i^{j-1}}{x_i^j} \right| \times 100\% \leq \varepsilon_s$$

Graphical depiction of the difference between (*a*) the Gauss-Seidel and (*b*) the Jacobi iterative methods for solving simultaneous linear algebraic equations.



Jacobi's Method

Perhaps the simplest iterative method for solving $\boldsymbol{Ax}=\boldsymbol{b}$ is Jacobi's Method.

Note that the simplicity of this method is both good and bad:

- good, because it is relatively easy to understand and thus is a good first taste of iterative methods;
- bad, because it is not typically used in practice (although its potential usefulness has been reconsidered with the advent of parallel computing).
- Still, it is a good starting point for learning about more useful, but more complicated, iterative methods.

https://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-jacobis-method

Jacobi Iterative Technique

Consider the following set of equations.

$$
\begin{aligned}
10x_1 \quad - x_2 \quad + 2x_3 \quad \quad &= 6 \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\
2x_1 - x_2 + 10x_3 - x_4 &= -11 \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}
$$

Convert the set Ax = b in the form of x = Tx + c.

$$x_1 = \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5}$$

$$x_2 = \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}$$

$$x_3 = -\frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10}$$

$$x_4 = -\frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}$$

Start with an initial approximation of:

$$x_1^{(0)} = 0,\ x_2^{(0)} = 0,\ x_3^{(0)} = 0 \text{ and } x_4^{(0)} = 0.$$

$$x_1^{(1)} = \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} + \frac{3}{5}$$

$$x_2^{(1)} = \frac{1}{11}x_1^{(0)} + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11}$$

$$x_3^{(1)} = -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} + \frac{1}{10}x_4^{(0)} - \frac{11}{10}$$

$$x_4^{(1)} = -\frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} + \frac{15}{8}$$

$$x_1^{(1)} = \frac{1}{10}(0) - \frac{1}{5}(0) + \frac{3}{5}$$

$$x_2^{(1)} = \frac{1}{11}(0) + \frac{1}{11}(0) - \frac{3}{11}(0) + \frac{25}{11}$$

$$x_3^{(1)} = -\frac{1}{5}(0) + \frac{1}{10}(0) + \frac{1}{10}(0) - \frac{11}{10}$$

$$x_4^{(1)} = -\frac{3}{8}(0) + \frac{1}{8}(0) + \frac{15}{8}$$

$x_1^{(1)} = 0.6000$, $x_2^{(1)} = 2.2727$,

$x_3^{(1)} = -1.1000$ **and** $x_4^{(1)} = 1.8750$

$$x_1^{(2)} = \frac{1}{10}x_2^{(1)} - \frac{1}{5}x_3^{(1)} + \frac{3}{5}$$

$$x_2^{(2)} = \frac{1}{11}x_1^{(1)} + \frac{1}{11}x_3^{(1)} - \frac{3}{11}x_4^{(1)} + \frac{25}{11}$$

$$x_3^{(2)} = -\frac{1}{5}x_1^{(1)} + \frac{1}{10}x_2^{(1)} + \frac{1}{10}x_4^{(1)} - \frac{11}{10}$$

$$x_4^{(2)} = -\frac{3}{8}x_2^{(1)} + \frac{1}{8}x_3^{(1)} + \frac{15}{8}$$

$$x_1^{(k)} = \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} + \frac{3}{5}$$

$$x_2^{(k)} = \frac{1}{11}x_1^{(k-1)} + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}$$

$$x_3^{(k)} = -\frac{1}{5}x_1^{(k-1)} + \frac{1}{10}x_2^{(k-1)} + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}$$

$$x_4^{(k)} = -\frac{3}{8}x_2^{(k-1)} + \frac{1}{8}x_3^{(k-1)} + \frac{15}{8}$$

Results of Jacobi Iteration:

k	0	1	2	3
$x_1^{(k)}$	**0.0000**	**0.6000**	**1.0473**	**0.9326**
$x_2^{(k)}$	**0.0000**	**2.2727**	**1.7159**	**2.0530**
$x_3^{(k)}$	**0.0000**	**-1.1000**	**-0.8052**	**-1.0493**
$x_4^{(k)}$	**0.0000**	**1.8750**	**0.8852**	**1.1309**

Gauss-Seidel Iterative Technique

Consider the following set of equations.

$$
\begin{aligned}
10x_1 - x_2 + 2x_3 \quad &= 6 \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25 \\
2x_1 - x_2 + 10x_3 - x_4 &= -11 \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}
$$

$$x_1^{(k)} = \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} + \frac{3}{5}$$

$$x_2^{(k)} = \frac{1}{11}x_1^{(k-1)} + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}$$

$$x_3^{(k)} = -\frac{1}{5}x_1^{(k-1)} + \frac{1}{10}x_2^{(k-1)} + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}$$

$$x_4^{(k)} = -\frac{3}{8}x_2^{(k-1)} + \frac{1}{8}x_3^{(k-1)} + \frac{15}{8}$$

$$x_1^{(k)} = \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} + \frac{3}{5}$$

$$x_2^{(k)} = \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}$$

$$x_3^{(k)} = -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}$$

$$x_4^{(k)} = -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8}$$

Results of Gauss-Seidel Iteration:
(Blue numbers are for Jacobi iterations.)

k	0	1	2	3
$x_1^{(k)}$	**0.0000**	**0.6000** **0.6000**	**1.0300** **1.0473**	**1.0065** **0.9326**
$x_2^{(k)}$	**0.0000**	**2.3272** **2.2727**	**2.0370** **1.7159**	**2.0036** **2.0530**
$x_3^{(k)}$	**0.0000**	**-0.9873** **-1.1000**	**-1.0140** **-0.8052**	**-1.0025** **-1.0493**
$x_4^{(k)}$	**0.0000**	**0.8789** **1.8750**	**0.9844** **0.8852**	**0.9983** **1.1309**

The solution is: $x_1 = 1$, $x_2 = 2$, $x_3 = -1$, $x_4 = 1$

It required 15 iterations for Jacobi method and 7 iterations for Gauss-Seidel method to arrive at the solution with a tolerance of 0.00001.

While Jacobi would usually be the slowest of the iterative methods, it is well suited to illustrate an algorithm that is well suited for parallel processing!!!

EXAMPLE Gauss-Seidel Method

Problem Statement. Use the Gauss-Seidel method to obtain the solution for

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Note that the solution is $x^T = \lfloor 3 \quad -2.5 \quad 7 \rfloor$

Solution. First, solve each of the equations for its unknown on the diagonal:

$$x_1 = \frac{7.85 + 0.1x_2 + 0.2x_3}{3} \qquad (1)$$

$$x_2 = \frac{-19.3 - 0.1x_1 + 0.3x_3}{7} \qquad (2)$$

$$x_3 = \frac{71.4 - 0.3x_1 + 0.2x_2}{10} \qquad (3)$$

By assuming that x_2 and x_3 are zero

$$x_1 = \frac{7.85 + 0.1(0) + 0.2(0)}{3} = 2.616667$$

This value, along with the assumed value of x_3 =0, can be substituted into Eq.(2) to calculate

$$x_2 = \frac{-19.3 - 0.1(2.616667) + 0.3(0)}{7} = -2.794524$$

The first iteration is completed by substituting the calculated values for x_1 and x_2 into Eq.(3) to yield

$$x_3 = \frac{71.4 - 0.3(2.616667) + 0.2(-2.794524)}{10} = 7.005610$$

For the second iteration, the same process is repeated to compute

$$x_1 = \frac{7.85 + 0.1(-2.794524) + 0.2(7.005610)}{3} = 2.990557$$

$$x_2 = \frac{-19.3 - 0.1(2.990557) + 0.3(7.005610)}{7} = -2.499625$$

$$x_3 = \frac{71.4 - 0.3(2.990557) + 0.2(-2.499625)}{10} = 7.000291$$

The method is, therefore, converging on the true solution. Additional iterations could be applied to improve the answers. Consequently, we can estimate the error. For example , for x_1

$$\varepsilon_{a,1} = \left| \frac{2.990557 - 2.616667}{2.990557} \right| \times 100\% = 12.5\%$$

For x_2 and x_3 , the error estimates are

$$\varepsilon_{a,2} = 11.8\%$$

$$\varepsilon_{a,3} = 0.076\%$$

Repeat to it again until the result is known to at least the tolerance specified by

$$\varepsilon_s$$

Example: Unbalanced three phase load

Three-phase loads are common in AC systems. When the system is balanced the analysis can be simplified to a single equivalent circuit model. However, when it is unbalanced the only practical solution involves the solution of simultaneous linear equations. In a model the following equations need to be solved.

$$\begin{bmatrix} 0.7460 & -0.4516 & 0.0100 & -0.0080 & 0.0100 & -0.0080 \\ 0.4516 & 0.7460 & 0.0080 & 0.0100 & 0.0080 & 0.0100 \\ 0.0100 & -0.0080 & 0.7787 & -0.5205 & 0.0100 & -0.0080 \\ 0.0080 & 0.0100 & 0.5205 & 0.7787 & 0.0080 & 0.0100 \\ 0.0100 & -0.0080 & 0.0100 & -0.0080 & 0.8080 & -0.6040 \\ 0.0080 & 0.0100 & 0.0080 & 0.0100 & 0.6040 & 0.8080 \end{bmatrix} \begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 120 \\ 0.000 \\ -60.00 \\ -103.9 \\ -60.00 \\ 103.9 \end{bmatrix}$$

Find the values of I_{ar}, I_{ai}, I_{br}, I_{bi}, I_{cr}, and I_{ci} using the Gauss-Seidel method.

Example: Unbalanced three phase load

Rewrite each equation to solve for each of the unknowns

$$I_{ar} = \frac{120.00 - (-0.4516)I_{ai} - 0.0100I_{br} - (-0.0080)I_{bi} - 0.0100I_{cr} - (-0.0080)I_{ci}}{0.7460}$$

$$I_{ai} = \frac{0.000 - 0.4516I_{ar} - 0.0080I_{br} - 0.0100I_{bi} - 0.0080I_{cr} - 0.0100I_{ci}}{0.7460}$$

$$I_{br} = \frac{-60.00 - 0.0100I_{ar} - (-0.0080)I_{ai} - (-0.5205)I_{bi} - 0.0100I_{cr} - (-0.0080)I_{ci}}{0.7787}$$

$$I_{bi} = \frac{-103.9 - 0.0080I_{ar} - 0.0100I_{ai} - 0.5205I_{br} - 0.0080I_{cr} - 0.0100I_{ci}}{0.7787}$$

$$I_{cr} = \frac{-60.00 - 0.0100I_{ar} - (-0.0080)I_{ai} - 0.0100I_{br} - (-0.0080)I_{bi} - (-0.6040)I_{ci}}{0.8080}$$

$$I_{ci} = \frac{103.9 - 0.0080I_{ar} - 0.0100I_{ai} - 0.0080I_{br} - 0.0100I_{bi} - 0.6040I_{cr}}{0.8080}$$

Example: Unbalanced three phase load

For iteration 1, start with an initial guess value

Initial Guess: $$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 20 \\ 20 \\ 20 \\ 20 \\ 20 \\ 20 \end{bmatrix}$$

Example: Unbalanced three phase load

Substituting the guess values into the first equation

$$I_{ar} = \frac{120 - (-0.4516)I_{ai} - 0.0100I_{br} - (-0.0080)I_{bi} - 0.0100I_{cr} - (-0.0080)I_{ci}}{0.7460}$$

$$= 172.86$$

Substituting the new value of I_{ar} and the remaining guess values into the second equation

$$I_{ai} = \frac{0.00 - 0.4516I_{ar} - 0.0080I_{br} - 0.0100I_{bi} - 0.0080I_{cr} - 0.0100I_{ci}}{0.7460}$$

$$= -105.61$$

Example: Unbalanced three phase load

Substituting the new values I_{ar}, I_{ai}, and the remaining guess values into the third equation

$$I_{br} = \frac{-60.00 - 0.0100 I_{ar} - (-0.0080) I_{ai} - (-0.5205) I_{bi} - 0.0100 I_{cr} - (-0.0080) I_{ci}}{0.7787}$$

$$= -67.039$$

Substituting the new values I_{ar}, I_{ai}, I_{br}, and the remaining guess values into the fourth equation

$$I_{bi} = \frac{-103.9 - 0.0080 I_{ar} - 0.0100 I_{ai} - 0.5205 I_{br} - 0.0080 I_{cr} - 0.0100 I_{ci}}{0.7787}$$

$$= -89.499$$

Example: Unbalanced three phase load

Substituting the new values I_{ar}, I_{ai}, I_{br}, I_{bi}, and the remaining guess values into the fifth equation

$$I_{cr} = \frac{-60.00 - 0.0100 I_{ar} - (-0.0080) I_{ai} - 0.0100 I_{br} - (-0.0080) I_{bi} - (-0.6040) I_{ci}}{0.8080}$$

$$= -62.548$$

Substituting the new values I_{ar}, I_{ai}, I_{br}, I_{bi}, I_{cr}, and the remaining guess value into the sixth equation

$$I_{ci} = \frac{103.9 - 0.0080 I_{ar} - 0.0100 I_{ai} - 0.0080 I_{br} - 0.0100 I_{bi} - 0.6040 I_{cr}}{0.8080}$$

$$= 176.71$$

Example: Unbalanced three phase load

At the end of the first iteration, the solution matrix is:

$$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 172.86 \\ -105.61 \\ -67.039 \\ -89.499 \\ -62.548 \\ 176.71 \end{bmatrix}$$

How accurate is the solution? Find the absolute relative approximate error using:

$$\left|\in_a\right|_i = \left|\frac{x_i^{new} - x_i^{old}}{x_i^{new}}\right| \times 100$$

Example: Unbalanced three phase load

Calculating the absolute relative approximate errors

$$\left|\in_a\right|_1 = \left|\frac{172.86-20}{172.86}\right|\times 100 = 88.430\%$$

$$\left|\in_a\right|_2 = \left|\frac{-105.61-20}{-105.61}\right|\times 100 = 118.94\%$$

$$\left|\in_a\right|_3 = \left|\frac{-67.039-20}{-67.039}\right|\times 100 = 129.83\%$$

$$\left|\in_a\right|_4 = \left|\frac{-89.499-20}{-89.499}\right|\times 100 = 122.35\%$$

$$\left|\in_a\right|_5 = \left|\frac{-62.548-20}{-62.548}\right|\times 100 = 131.98\%$$

$$\left|\in_a\right|_6 = \left|\frac{176.71-20}{176.71}\right|\times 100 = 88.682\%$$

The maximum error after the first iteration is:

131.98%

Another iteration is needed!

Example: Unbalanced three phase load

Starting with the values obtained in iteration #1

$$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 172.86 \\ -105.61 \\ -67.039 \\ -89.499 \\ -62.548 \\ 176.71 \end{bmatrix}$$

Substituting the values from Iteration 1 into the first equation

$$I_{ar} = \frac{120.00 - (-0.4516)I_{ai} - 0.0100 I_{br} - (-0.0080) I_{bi} - 0.0100 I_{cr} - (-0.0080) I_{ci}}{0.7460}$$

$$= 99.600$$

Example: Unbalanced three phase load

Substituting the new value of I_{ar} and the remaining values from Iteration 1 into the second equation

$$I_{ai} = \frac{0.00 - 0.4516 I_{ar} - 0.0080 I_{br} - 0.0100 I_{bi} - 0.0080 I_{cr} - 0.0100 I_{ci}}{0.7460}$$

$$= -60.073$$

Substituting the new values I_{ar}, I_{ai}, and the remaining values from Iteration 1 into the third equation

$$I_{br} = \frac{-60.00 - 0.0100 I_{ar} - (-0.0080) I_{ai} - (-0.5205) I_{bi} - 0.0100 I_{cr} - (-0.0080) I_{ci}}{0.7787}$$

$$= -136.15$$

Example: Unbalanced three phase load

Substituting the new values I_{ar}, I_{ai}, I_{br}, and the remaining values from Iteration 1 into the fourth equation

$$I_{bi} = \frac{-103.9 - 0.0080 I_{ar} - 0.0100 I_{ai} - 0.5205 I_{br} - 0.0080 I_{cr} - 0.0100 I_{ci}}{0.7787}$$

$$= -44.299$$

Substituting the new values I_{ar}, I_{ai}, I_{br}, I_{bi}, and the remaining values From Iteration 1 into the fifth equation

$$I_{cr} = \frac{-60.00 - 0.0100 I_{ar} - (-0.0080) I_{ai} - 0.0100 I_{br} - (-0.0080) I_{bi} - (-0.6040) I_{ci}}{0.8080}$$

$$= 57.259$$

Example: Unbalanced three phase load

Substituting the new values I_{ar}, I_{ai}, I_{br}, I_{bi}, I_{cr}, and the remaining value from Iteration 1 into the sixth equation

$$I_{ci} = \frac{103.9 - 0.0080 I_{ar} - 0.0100 I_{ai} - 0.0080 I_{br} - 0.0100 I_{bi} - 0.6040 I_{cr}}{0.8080}$$

$$= 87.441$$

The solution matrix at the end of the second iteration is:

$$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 99.600 \\ -60.073 \\ -136.15 \\ -44.299 \\ 57.259 \\ 87.441 \end{bmatrix}$$

Example: Unbalanced three phase load

Calculating the absolute relative approximate errors for the second iteration

$$\left|\in_a\right|_1 = \left|\frac{99.600 - 172.86}{99.600}\right| \times 100 = 73.552\%$$

$$\left|\in_a\right|_2 = \left|\frac{-60.073 - (-105.61)}{-60.073}\right| \times 100 = 75.796\%$$

$$\left|\in_a\right|_3 = \left|\frac{-136.35 - (-67.039)}{-136.35}\right| \times 100 = 50.762\%$$

$$\left|\in_a\right|_4 = \left|\frac{-44.299 - (-89.499)}{-44.299}\right| \times 100 = 102.03\%$$

$$\left|\in_a\right|_5 = \left|\frac{57.259 - (-62.548)}{57.259}\right| \times 100 = 209.24\%$$

$$\left|\in_a\right|_6 = \left|\frac{87.441 - 176.71}{87.441}\right| \times 100 = 102.09\%$$

The maximum error after the second iteration is:

209.24%

More iterations are needed!

Example: Unbalanced three phase load

Repeating more iterations, the following values are obtained

Iteration	I_{ar}	I_{ai}	I_{br}	I_{bi}	I_{cr}	I_{ci}
1	172.86	−105.61	−67.039	−89.499	−62.548	176.71
2	99.600	−60.073	−136.15	−44.299	57.259	87.441
3	126.01	−76.015	−108.90	−62.667	−10.478	137.97
4	117.25	−70.707	−119.62	−55.432	27.658	109.45
5	119.87	−72.301	−115.62	−58.141	6.2513	125.49
6	119.28	−71.936	−116.98	−57.216	18.241	116.53

Iteration	$\left\|\in_a\right\|_1 \%$	$\left\|\in_a\right\|_2 \%$	$\left\|\in_a\right\|_3 \%$	$\left\|\in_a\right\|_4 \%$	$\left\|\in_a\right\|_5 \%$	$\left\|\in_a\right\|_6 \%$
1	88.430	118.94	129.83	122.35	131.98	88.682
2	73.552	75.796	50.762	102.03	209.24	102.09
3	20.960	20.972	25.027	29.311	646.45	36.623
4	7.4738	7.5067	8.9631	13.053	137.89	26.001
5	2.1840	2.2048	3.4633	4.6595	342.43	12.742
6	0.49408	0.50789	1.1629	1.6170	65.729	7.6884

Example: Unbalanced three phase load

After six iterations, the solution matrix is

$$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 119.28 \\ -71.936 \\ -116.98 \\ 57.216 \\ 18.241 \\ 116.53 \end{bmatrix}$$

The maximum error after the sixth iteration is:

65.729%

The absolute relative approximate error is still high, but allowing for more iterations, the error quickly begins to converge to zero.

What could have been done differently to allow for a faster convergence?

Example: Unbalanced three phase load

Repeating more iterations, the following values are obtained

Iteration	I_{ar}	I_{ai}	I_{br}	I_{bi}	I_{cr}	I_{ci}
32	119.33	−71.973	−116.66	−57.432	13.940	119.74
33	119.33	−71.973	−116.66	−57.432	13.940	119.74

Iteration	$\|\in_a\|_1$ %	$\|\in_a\|_2$ %	$\|\in_a\|_3$ %	$\|\in_a\|_4$ %	$\|\in_a\|_5$ %	$\|\in_a\|_6$ %
32	3.0666×10^{-7}	3.0047×10^{-7}	4.2389×10^{-7}	5.7116×10^{-7}	2.0941×10^{-5}	1.8238×10^{-6}
33	1.7062×10^{-7}	1.6718×10^{-7}	2.3601×10^{-7}	3.1801×10^{-7}	1.1647×10^{-5}	1.0144×10^{-6}

Example: Unbalanced three phase load

After 33 iterations, the solution matrix is

$$\begin{bmatrix} I_{ar} \\ I_{ai} \\ I_{br} \\ I_{bi} \\ I_{cr} \\ I_{ci} \end{bmatrix} = \begin{bmatrix} 119.33 \\ -71.973 \\ -116.66 \\ -57.432 \\ 13.940 \\ 119.74 \end{bmatrix}$$

The maximum absolute relative approximate error is $1.1647 \times 10^{-5}\%$.

Gauss-Seidel Method: Pitfall

Even though done correctly, the answer may not converge to the correct answer.

This is a pitfall of the Gauss-Seidel method: not all systems of equations will converge.

Is there a fix?

One class of system of equations always converges: One with a diagonally dominant coefficient matrix.

Diagonally dominant: [A] in [A] [X] = [C] is diagonally dominant if:

$$\left|a_{ii}\right| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} \left|a_{ij}\right|$$ for all 'i'

and

$$\left|a_{ii}\right| > \sum_{\substack{j=1 \\ j \neq i}}^{n} \left|a_{ij}\right|$$ for at least one 'i'

Gauss-Seidel Method: Pitfall

Diagonally dominant: The coefficient on the diagonal must be at least equal to the sum of the other coefficients in that row and at least one row with a diagonal coefficient greater than the sum of the other coefficients in that row.

Which coefficient matrix is diagonally dominant?

$$A = \begin{bmatrix} 2 & 5.81 & 34 \\ 45 & 43 & 1 \\ 123 & 16 & 1 \end{bmatrix} \qquad [B] = \begin{bmatrix} 124 & 34 & 56 \\ 23 & 53 & 5 \\ 96 & 34 & 129 \end{bmatrix}$$

Most physical systems do result in simultaneous linear equations that have diagonally dominant coefficient matrices.

Gauss-Seidel Method: Example 2

Given the system of equations

$$12x_1 + 3x_2 - 5x_3 = 1$$

$$x_1 + 5x_2 + 3x_3 = 28$$

$$3x_1 + 7x_2 + 13x_3 = 76$$

The coefficient matrix is:

$$A = \begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

With an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Will the solution converge using the Gauss-Seidel method?

Gauss-Seidel Method: Example 2

Checking if the coefficient matrix is diagonally dominant

$$A = \begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

$$|a_{11}| = |12| = 12 \geq |a_{12}| + |a_{13}| = |3| + |-5| = 8$$

$$|a_{22}| = |5| = 5 \geq |a_{21}| + |a_{23}| = |1| + |3| = 4$$

$$|a_{33}| = |13| = 13 \geq |a_{31}| + |a_{32}| = |3| + |7| = 10$$

The inequalities are all true and at least one row is *strictly* greater than:

Therefore: The solution should converge using the Gauss-Siedel Method

Gauss-Seidel Method: Example 2

Rewriting each equation

$$\begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 28 \\ 76 \end{bmatrix}$$

$$x_1 = \frac{1 - 3x_2 + 5x_3}{12}$$

$$x_2 = \frac{28 - x_1 - 3x_3}{5}$$

$$x_3 = \frac{76 - 3x_1 - 7x_2}{13}$$

With an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$x_1 = \frac{1 - 3(0) + 5(1)}{12} = 0.50000$$

$$x_2 = \frac{28 - (0.5) - 3(1)}{5} = 4.9000$$

$$x_3 = \frac{76 - 3(0.50000) - 7(4.9000)}{13} = 3.0923$$

Gauss-Seidel Method: Example 2

The absolute relative approximate error

$$\left|\in_a\right|_1 = \left|\frac{0.50000 - 1.0000}{0.50000}\right| \times 100 = 100.00\%$$

$$\left|\in_a\right|_2 = \left|\frac{4.9000 - 0}{4.9000}\right| \times 100 = 100.00\%$$

$$\left|\in_a\right|_3 = \left|\frac{3.0923 - 1.0000}{3.0923}\right| \times 100 = 67.662\%$$

The maximum absolute relative error after the first iteration is 100%

Gauss-Seidel Method: Example 2

After Iteration #1

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.5000 \\ 4.9000 \\ 3.0923 \end{bmatrix}$$

Substituting the x values into the equations

$$x_1 = \frac{1 - 3(4.9000) + 5(3.0923)}{12} = 0.14679$$

$$x_2 = \frac{28 - (0.14679) - 3(3.0923)}{5} = 3.7153$$

$$x_3 = \frac{76 - 3(0.14679) - 7(4.900)}{13} = 3.8118$$

After Iteration #2

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.14679 \\ 3.7153 \\ 3.8118 \end{bmatrix}$$

Gauss-Seidel Method: Example 2

Iteration #2 absolute relative approximate error

$$\left|\in_a\right|_1 = \left|\frac{0.14679 - 0.50000}{0.14679}\right| \times 100 = 240.61\%$$

$$\left|\in_a\right|_2 = \left|\frac{3.7153 - 4.9000}{3.7153}\right| \times 100 = 31.889\%$$

$$\left|\in_a\right|_3 = \left|\frac{3.8118 - 3.0923}{3.8118}\right| \times 100 = 18.874\%$$

The maximum absolute relative error after the first iteration is 240.61%

This is much larger than the maximum absolute relative error obtained in iteration #1. Is this a problem?

Gauss-Seidel Method: Example 2

Repeating more iterations, the following values are obtained

Iteration	a_1	$\left\|\in_a\right\|_1 \%$	a_2	$\left\|\in_a\right\|_2 \%$	a_3	$\left\|\in_a\right\|_3 \%$
1	0.50000	100.00	4.9000	100.00	3.0923	67.662
2	0.14679	240.61	3.7153	31.889	3.8118	18.876
3	0.74275	80.236	3.1644	17.408	3.9708	4.0042
4	0.94675	21.546	3.0281	4.4996	3.9971	0.65772
5	0.99177	4.5391	3.0034	0.82499	4.0001	0.074383
6	0.99919	0.74307	3.0001	0.10856	4.0001	0.00101

The solution obtained $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.99919 \\ 3.0001 \\ 4.0001 \end{bmatrix}$ is close to the exact solution of $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$

Gauss-Seidel Method: Example 3

Given the system of equations

$$3x_1 + 7x_2 + 13x_3 = 76$$

$$x_1 + 5x_2 + 3x_3 = 28$$

$$12x_1 + 3x_2 - 5x_3 = 1$$

With an initial guess of

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Rewriting the equations

$$x_1 = \frac{76 - 7x_2 - 13x_3}{3}$$

$$x_2 = \frac{28 - x_1 - 3x_3}{5}$$

$$x_3 = \frac{1 - 12x_1 - 3x_2}{-5}$$

Gauss-Seidel Method: Example 3

Conducting six iterations, the following values are obtained

Iteration	a_1	$\left\|\in_a\right\|_1 \%$	A_2	$\left\|\in_a\right\|_2 \%$	a_3	$\left\|\in_a\right\|_3 \%$
1	21.000	95.238	0.80000	100.00	50.680	98.027
2	−196.15	110.71	14.421	94.453	−462.30	110.96
3	−1995.0	109.83	−116.02	112.43	4718.1	109.80
4	−20149	109.90	1204.6	109.63	−47636	109.90
5	2.0364×10^5	109.89	−12140	109.92	4.8144×10^5	109.89
6	-2.0579×10^5	109.89	1.2272×10^5	109.89	-4.8653×10^6	109.89

The values are not converging.

Does this mean that the Gauss-Seidel method cannot be used?

Gauss-Seidel Method

The Gauss-Seidel Method can still be used

The coefficient matrix is not diagonally dominant

$$A = \begin{bmatrix} 3 & 7 & 13 \\ 1 & 5 & 3 \\ 12 & 3 & -5 \end{bmatrix}$$

But this is the same set of equations used in example #2, which did converge.

$$A = \begin{bmatrix} 12 & 3 & -5 \\ 1 & 5 & 3 \\ 3 & 7 & 13 \end{bmatrix}$$

If a system of linear equations is not diagonally dominant, check to see if rearranging the equations can form a diagonally dominant matrix.

Gauss-Seidel Method

Not every system of equations can be rearranged to have a diagonally dominant coefficient matrix.

Observe the set of equations

$$x_1 + x_2 + x_3 = 3$$

$$2x_1 + 3x_2 + 4x_3 = 9$$

$$x_1 + 7x_2 + x_3 = 9$$

Which equation(s) prevents this set of equation from having a diagonally dominant coefficient matrix?

Jacobi Algorithm - pseudocode

Numerical Algorithm of Jacobi Method

Input: $A = [a_{ij}]$, $\boldsymbol{b}, \boldsymbol{XO} = \boldsymbol{x}^{(0)}$, tolerance TOL, maximum number of iterations N.

Step 1 Set $k = 1$

Step 2 while $(k \le N)$ do Steps 3-6

Step 3 For for $i = 1,2,, \ldots n$

$$x_i = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1, \\ j\neq i}}^{n}(-a_{ij}\boldsymbol{XO}_j) + b_i\right],$$

Step 4 If $\left\|\boldsymbol{x} - \boldsymbol{XO}\right\| < TOL$, then OUTPUT $(x_1, x_2, x_3, \ldots x_n)$;

STOP.

Step 5 Set $k = k + 1$.

Step 6 For for $i = 1,2,, \ldots n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 OUTPUT $(x_1, x_2, x_3, \ldots x_n)$;

STOP.

Another stopping criterion in Step 4: $\frac{||\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}||}{||\boldsymbol{x}^{(k)}||}$

Gauss-Seidel algorithm - pseudocode

Numerical Algorithm of Gauss-Seidel Method

Input: $A = [a_{ij}]$, $\boldsymbol{b}, \boldsymbol{XO} = x^{(0)}$, tolerance TOL, maximum number of iterations N.

Step 1 Set $k = 1$

Step 2 while $(k \leq N)$ do Steps 3-6

Step 3 For for $i = 1,2,, \dots n$

$$x_i = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j) - \sum_{j=i+1}^{n}(a_{ij}\boldsymbol{XO}_j) + b_i\right],$$

Step 4 If $||x - \boldsymbol{XO}|| < TOL$, then OUTPUT $(x_1 , x_2 , x_3 , \dots x_n)$;

STOP.

Step 5 Set $k = k + 1$.

Step 6 For for $i = 1,2,, \dots n$

Set $\boldsymbol{XO}_i = x_i$.

Step 7 OUTPUT $(x_1 , x_2 , x_3 , \dots x_n)$;

STOP.

Links

https://www.geogebra.org/m/XBKbmXY7

https://www.geogebra.org/m/ekjpgF3z