

MAP 2210 – Aplicações de Álgebra Linear

1º Semestre - 2020

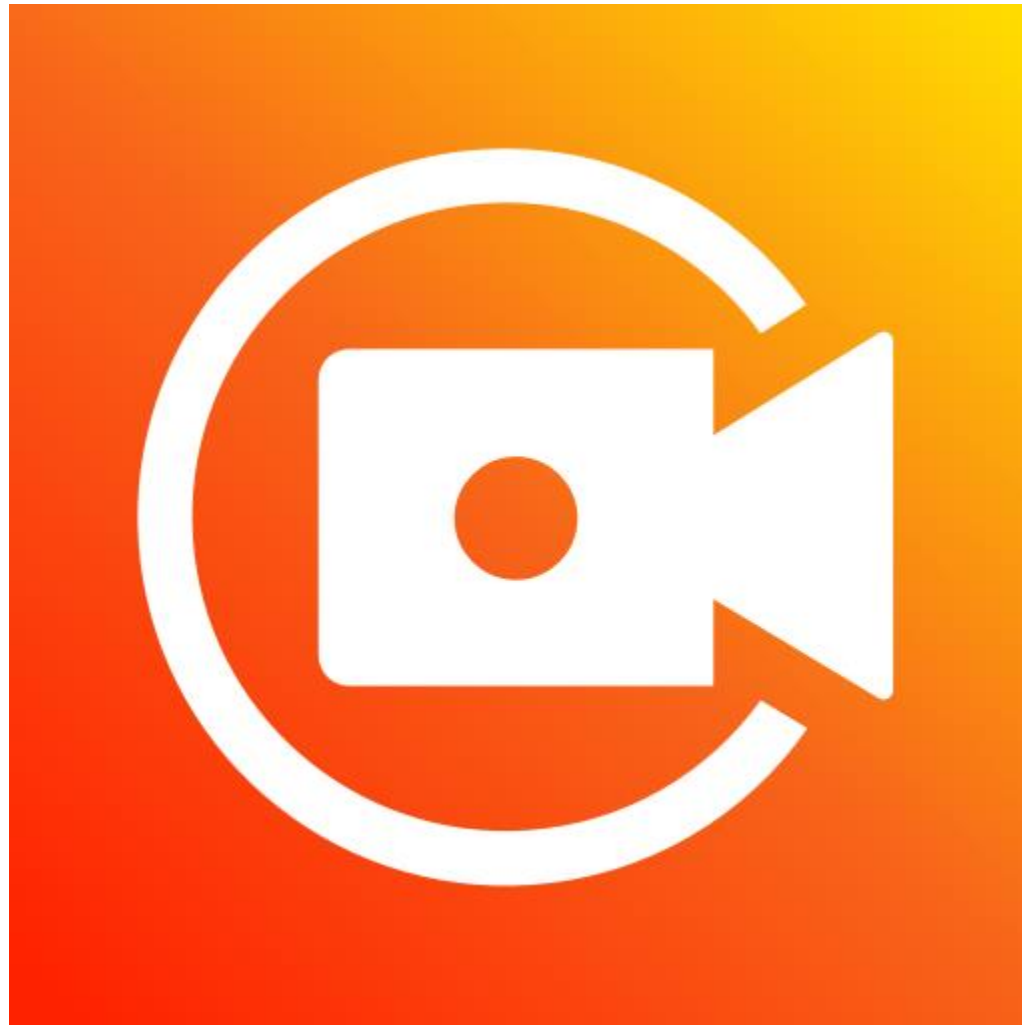
Prof. Dr. Luis Carlos de Castro Santos

lsantos@ime.usp.br

Objetivos

Formação básica de álgebra linear aplicada a problemas numéricos. Resolução de problemas em microcomputadores usando linguagens e/ou software adequados fora do horário de aula.

NÃO ESQUEÇA DE INICIAR A GRAVAÇÃO



MAP 2210 – Aplicações de Álgebra Linear

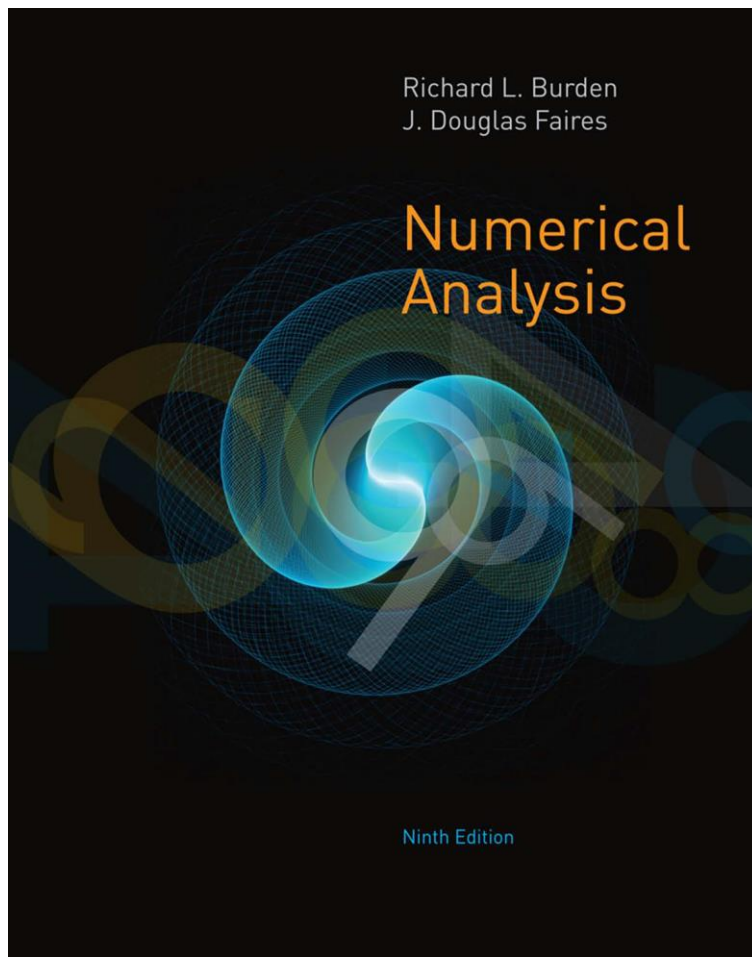
1º Semestre - 2020

Prof. Dr. Luis Carlos de Castro Santos

lsantos@ime.usp.br

Objetivos

Formação básica de álgebra linear aplicada a problemas numéricos. Resolução de problemas em microcomputadores usando linguagens e/ou software adequados fora do horário de aula.



Numerical Analysis

NINTH EDITION

Richard L. Burden

Youngstown State University

J. Douglas Faires

Youngstown State University

6 Direct Methods for Solving Linear Systems 357

- 6.1 Linear Systems of Equations 358
- 6.2 Pivoting Strategies 372
- 6.3 Linear Algebra and Matrix Inversion 381
- 6.4 The Determinant of a Matrix 396
- 6.5 Matrix Factorization 400
- 6.6 Special Types of Matrices 411
- 6.7 Survey of Methods and Software 428

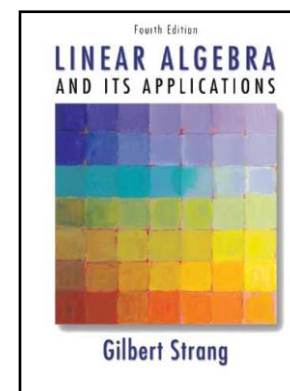
7 Iterative Techniques in Matrix Algebra 431

- 7.1 Norms of Vectors and Matrices 432
- 7.2 Eigenvalues and Eigenvectors 443
- 7.3 The Jacobi and Gauss-Siedel Iterative Techniques 450
- ➔ 7.4 Relaxation Techniques for Solving Linear Systems 462
- 7.5 Error Bounds and Iterative Refinement 469
- 7.6 The Conjugate Gradient Method 479
- 7.7 Survey of Methods and Software 495

9 Approximating Eigenvalues 561

- 9.1 Linear Algebra and Eigenvalues 562
- 9.2 Orthogonal Matrices and Similarity Transformations 570
- 9.3 The Power Method 576
- 9.4 Householder's Method 593
- 9.5 The QR Algorithm 601
- 9.6 Singular Value Decomposition 614
- 9.7 Survey of Methods and Software 626

+



+



7.4 Relaxation Techniques for Solving Linear Systems

We saw in Section 7.3 that the rate of convergence of an iterative technique depends on the spectral radius of the matrix associated with the method. One way to select a procedure to accelerate convergence is to choose a method whose associated matrix has minimal spectral radius. Before describing a procedure for selecting such a method, we need to introduce a new means of measuring the amount by which an approximation to the solution to a linear system differs from the true solution to the system. The method makes use of the vector described in the following definition.

Definition 7.23

Suppose $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is an approximation to the solution of the linear system defined by $A\mathbf{x} = \mathbf{b}$. The **residual vector** for $\tilde{\mathbf{x}}$ with respect to this system is $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$. ■

Resíduo

$$\mathbf{r} = \mathbf{b} - A \mathbf{x}$$

Gauss-Seidel

$$\mathbf{x}_i^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \dots \\ x_{i-1}^{(k)} \\ x_i^{(k-1)} \\ \dots \\ x_n^{(k-1)} \end{bmatrix}$$

$$\mathbf{r}_i^{(k)} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_{i-1} \\ b_i \\ \dots \\ b_n \end{bmatrix} - A \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \dots \\ x_{i-1}^{(k)} \\ x_i^{(k-1)} \\ \dots \\ x_n^{(k-1)} \end{bmatrix} = \begin{bmatrix} r_{1i}^{(k)} \\ r_{2i}^{(k)} \\ \dots \\ r_{(i-1)i}^{(k)} \\ r_{ii}^{(k)} \\ \dots \\ r_{ni}^{(k)} \end{bmatrix}$$

In procedures such as the Jacobi or Gauss-Seidel methods, a residual vector is associated with each calculation of an approximate component to the solution vector. The true objective is to generate a sequence of approximations that will cause the residual vectors to converge rapidly to zero. Suppose we let

$$\mathbf{r}_i^{(k)} = (r_{1i}^{(k)}, r_{2i}^{(k)}, \dots, r_{ni}^{(k)})^t$$

denote the residual vector for the Gauss-Seidel method corresponding to the approximate solution vector $\mathbf{x}_i^{(k)}$ defined by

$$\mathbf{x}_i^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k-1)}, \dots, x_n^{(k-1)})^t.$$

The m th component of $\mathbf{r}_i^{(k)}$ is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^n a_{mj} x_j^{(k-1)}, \quad (7.13)$$

or, equivalently,

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^n a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

for each $m = 1, 2, \dots, n$.

In particular, the i th component of $\mathbf{r}_i^{(k)}$ is

$$r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} - a_{ii}x_i^{(k-1)},$$

so

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)}. \quad (7.14)$$

Recall, however, that in the Gauss-Seidel method, $x_i^{(k)}$ is chosen to be

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \right], \quad (7.15)$$

so Eq. (7.14) can be rewritten as

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = a_{ii}x_i^{(k)}.$$

Consequently, the Gauss-Seidel method can be characterized as choosing $x_i^{(k)}$ to satisfy

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}. \quad (7.16)$$

We can derive another connection between the residual vectors and the Gauss-Seidel technique. Consider the residual vector $\mathbf{r}_{i+1}^{(k)}$, associated with the vector $\mathbf{x}_{i+1}^{(k)} = (x_1^{(k)}, \dots, x_i^{(k)}, x_{i+1}^{(k-1)}, \dots, x_n^{(k-1)})^t$. By Eq. (7.13) the i th component of $\mathbf{r}_{i+1}^{(k)}$ is

$$\begin{aligned} r_{i,i+1}^{(k)} &= b_i - \sum_{j=1}^i a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} \\ &= b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} - a_{ii}x_i^{(k)}. \end{aligned}$$

By the manner in which $x_i^{(k)}$ is defined in Eq. (7.15) we see that $r_{i,i+1}^{(k)} = 0$. In a sense, then, the Gauss-Seidel technique is characterized by choosing each $x_{i+1}^{(k)}$ in such a way that the i th component of $\mathbf{r}_{i+1}^{(k)}$ is zero.

Choosing $x_{i+1}^{(k)}$ so that one coordinate of the residual vector is zero, however, is not necessarily the most efficient way to reduce the norm of the vector $\mathbf{r}_{i+1}^{(k)}$. If we modify the Gauss-Seidel procedure, as given by Eq. (7.16), to

$$x_i^{(k)} = x_i^{(k-1)} + \omega \frac{r_{ii}^{(k)}}{a_{ii}}, \quad (7.17)$$

then for certain choices of positive ω we can reduce the norm of the residual vector and obtain significantly faster convergence.

Methods involving Eq. (7.17) are called **relaxation methods**. For choices of ω with $0 < \omega < 1$, the procedures are called **under-relaxation methods**. We will be interested in choices of ω with $1 < \omega$, and these are called **over-relaxation methods**. They are used to accelerate the convergence for systems that are convergent by the Gauss-Seidel technique. The methods are abbreviated **SOR**, for **Successive Over-Relaxation**, and are particularly useful for solving the linear systems that occur in the numerical solution of certain partial-differential equations.

Before illustrating the advantages of the SOR method, we note that by using Eq. (7.14), we can reformulate Eq. (7.17) for calculation purposes as

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

To determine the matrix form of the SOR method, we rewrite this as

$$a_{ii}x_i^{(k)} + \omega \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = (1 - \omega)a_{ii}x_i^{(k-1)} - \omega \sum_{j=i+1}^n a_{ij}x_j^{(k-1)} + \omega b_i,$$

so that in vector form, we have

$$(D - \omega L)\mathbf{x}^{(k)} = [(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega \mathbf{b}.$$

That is,

$$\mathbf{x}^{(k)} = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D - \omega L)^{-1}\mathbf{b}. \quad (7.18)$$

Letting $T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]$ and $\mathbf{c}_\omega = \omega(D - \omega L)^{-1}\mathbf{b}$, gives the SOR technique the form

$$\mathbf{x}^{(k)} = T_\omega \mathbf{x}^{(k-1)} + \mathbf{c}_\omega. \quad (7.19)$$

Example 1

The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned}4x_1 + 3x_2 &= 24, \\3x_1 + 4x_2 - x_3 &= 30, \\-x_2 + 4x_3 &= -24,\end{aligned}$$

has the solution $(3, 4, -5)^t$. Compare the iterations from the Gauss-Seidel method and the SOR method with $\omega = 1.25$ using $\mathbf{x}^{(0)} = (1, 1, 1)^t$ for both methods.

Gauss-Seidel, $w = 1$

Gauss-Seidel			
w	1	(1-w)	0
SOR			
it	x1	x2	x3
0	1,00	1,00	1,00
1	5,25	3,81	-5,05
2	3,14	3,88	-5,03
3	3,09	3,93	-5,02
4	3,05	3,95	-5,01
5	3,03	3,97	-5,01
6	3,02	3,98	-5,00
7	3,01	3,99	-5,00
8	3,01	3,99	-5,00
9	3,01	4,00	-5,00
10	3,00	4,00	-5,00

SOR, $w = 1.25$

SOR			
w	1,25	(1-w)	-0,25
SOR			
it	x1	x2	x3
0	1,00	1,00	1,00
1	6,31	3,52	-6,65
2	2,62	3,96	-4,60
3	3,13	4,01	-5,10
4	2,96	4,01	-4,97
5	3,00	4,00	-5,01
6	3,00	4,00	-5,00
7	3,00	4,00	-5,00
8	3,00	4,00	-5,00
9	3,00	4,00	-5,00
10	3,00	4,00	-5,00



convergência em
duas casas
decimais

$$T_g = \text{inv}(D - L) * U$$

$$T_w = \text{inv}(D - wL) * [(1-w)D + wU]$$

Input:

$$\begin{pmatrix} 4 & 0 & 0 \\ 3 & 4 & 0 \\ 0 & -1 & 4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 & -3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Input:

$$\begin{pmatrix} 4 & 0 & 0 \\ 3 \times 1.25 & 4 & 0 \\ 0 & -1.25 & 4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} -0.25 \times 4 & -3 \times 1.25 & 0 \\ 0 & -0.25 \times 4 & 1 \times 1.25 \\ 0 & 0 & -0.25 \times 4 \end{pmatrix}$$

eigenvalues

$$\begin{pmatrix} 0 & \frac{3}{4} & 0 \\ 0 & \frac{9}{16} & \frac{1}{4} \\ 0 & \frac{9}{64} & \frac{1}{16} \end{pmatrix}$$

eigenvalues

$$\begin{pmatrix} -0.25 & -0.9375 & 0 \\ 0.234375 & 0.628906 & 0.3125 \\ 0.0732422 & 0.196533 & -0.152344 \end{pmatrix}$$

Results:

$$\lambda_1 = \frac{5}{8} \approx 0.625$$

$$\lambda_2 = 0$$

Results:

$$\lambda_1 \approx -0.25$$

$$\lambda_2 \approx 0.238281 + 0.0756454 i$$

$$\lambda_3 \approx 0.238281 - 0.0756454 i$$

Solution For each $k = 1, 2, \dots$, the equations for the Gauss-Seidel method are

$$x_1^{(k)} = -0.75x_2^{(k-1)} + 6,$$

$$x_2^{(k)} = -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5,$$

$$x_3^{(k)} = 0.25x_2^{(k)} - 6,$$

and the equations for the SOR method with $\omega = 1.25$ are

$$x_1^{(k)} = -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5,$$

$$x_2^{(k)} = -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375,$$

$$x_3^{(k)} = 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5.$$

The first seven iterates for each method are listed in Tables 7.3 and 7.4. For the iterates to be accurate to seven decimal places, the Gauss-Seidel method requires 34 iterations, as opposed to 14 iterations for the SOR method with $\omega = 1.25$. ■

Gauss-Seidel

Table 7.3

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_3^{(k)}$	1	-5.046875	-5.0292969	-5.0183105	-5.0114441	-5.0071526	-5.0044703	-5.0027940

SOR method with $\omega = 1.25$.

Table 7.4

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	-6.6501465	-4.6004238	-5.0966863	-4.9734897	-5.0057135	-4.9982822	-5.0003486

An obvious question to ask is how the appropriate value of ω is chosen when the SOR method is used. Although no complete answer to this question is known for the general $n \times n$ linear system, the following results can be used in certain important situations.

Theorem 7.24

(Kahan)

If $a_{ii} \neq 0$, for each $i = 1, 2, \dots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$. ■

Theorem 7.25

(Ostrowski-Reich)

If A is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$. ■

Theorem 7.26

If A is positive definite and tridiagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.$$

With this choice of ω , we have $\rho(T_\omega) = \omega - 1$. ■

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}.$$

Solution This matrix is clearly tridiagonal, so we can apply the result in Theorem 7.26 if we can also show that it is positive definite. Because the matrix is symmetric, Theorem 6.24 on page 416 states that it is positive definite if and only if all its leading principle submatrices has a positive determinant. This is easily seen to be the case because

$$\det(A) = 24, \quad \det\left(\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}\right) = 7, \quad \text{and} \quad \det([4]) = 4.$$

Because

$$T_j = D^{-1}(L + U) = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix},$$

we have

$$T_j - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix},$$

so

$$\det(T_j - \lambda I) = -\lambda(\lambda^2 - 0.625).$$

Thus

$$\rho(T_j) = \sqrt{0.625}$$

and

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.$$

This explains the rapid convergence obtained in Example 1 when using $\omega = 1.25$. ■

SOR

To solve $A\mathbf{x} = \mathbf{b}$ given the parameter ω and an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n ; the entries a_{ij} , $1 \leq i, j \leq n$, of the matrix A ; the entries b_i , $1 \leq i \leq n$, of \mathbf{b} ; the entries XO_i , $1 \leq i \leq n$, of $\mathbf{XO} = \mathbf{x}^{(0)}$; the parameter ω ; tolerance TOL ; maximum number of iterations N .

OUTPUT the approximate solution x_1, \dots, x_n or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \leq N)$ do Steps 3–6.

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

$$\text{set } x_i = (1 - \omega)XO_i + \frac{1}{a_{ii}} \left[\omega \left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}XO_j + b_i \right) \right].$$

Step 4 If $\|\mathbf{x} - \mathbf{XO}\| < TOL$ then OUTPUT (x_1, \dots, x_n) ;
(The procedure was successful.)
 STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \dots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(The procedure was successful.)
 STOP.



EXERCISE SET 7.4

1. Find the first two iterations of the SOR method with $\omega = 1.1$ for the following linear systems, using $\mathbf{x}^{(0)} = \mathbf{0}$:
 - a.
$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1, \\ 3x_1 + 6x_2 + 2x_3 &= 0, \\ 3x_1 + 3x_2 + 7x_3 &= 4. \end{aligned}$$
 - b.
$$\begin{aligned} 10x_1 - x_2 &= 9, \\ -x_1 + 10x_2 - 2x_3 &= 7, \\ -2x_2 + 10x_3 &= 6. \end{aligned}$$
 - c.
$$\begin{aligned} 10x_1 + 5x_2 &= 6, \\ 5x_1 + 10x_2 - 4x_3 &= 25, \\ -4x_2 + 8x_3 - x_4 &= -11, \\ -x_3 + 5x_4 &= -11. \end{aligned}$$
 - d.
$$\begin{aligned} 4x_1 + x_2 + x_3 + x_5 &= 6, \\ -x_1 - 3x_2 + x_3 + x_4 &= 6, \\ 2x_1 + x_2 + 5x_3 - x_4 - x_5 &= 6, \\ -x_1 - x_2 - x_3 + 4x_4 &= 6, \\ 2x_2 - x_3 + x_4 + 4x_5 &= 6. \end{aligned}$$
3. Repeat Exercise 1 using $\omega = 1.3$.

Modifique sua implementação em Python de Gauss-Seidel para SOR

Fim...

ALLA 11