Linear Systems

- Many problems are either quadratic, or approximated by a quadratic functions. To find their local min/max we need to take their gradient and set them equal to zero, leading to solving a linear system.
  - $f(x) = x^T Ax + b^T x + c$

    $$\rightarrow \nabla f(x) = 2Ax + b = 0.$$

  - $f(x) = \alpha \|x\|^2 + \|Ax - b\|^2$

    $$\nabla f(x) = 2\alpha Ix + 2A^T(Ax - b) = (2A^T A + 2\alpha I)x - 2A^T b = 0.$$

- Some problems are already formulated into a linear system of equations
  - Cost and revenue problems
  - polynomial interpolation
  - solution of ordinary differential equations
  - linear regression
  - ranking methods: PageRank

Example

Suppose that a company is planning to invest the total 1 million dolors into three projects, A,B,C. The estimated one-year returns for each project under two different scenarios, I, II, are summarized below

Project	I	II
A	1.4	0.9
B	0.8	2.0
C	1.2	1.0

The manager's goal is to distribute funds between the projects in a way that would yield the total return of \$1.2 under both scenarios.
Introduce variables: x_1, x_2, x_3 for the amount of money (in million of dolors) to be invested in project A, B, C, respectively. Then the problem can be formulated as the following system of linear equations:

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 1.4x_1 + 0.8x_2 + 1.2x_3 &= 1.2 \\ 0.9x_1 + 2.0x_2 + 1.0x_3 &= 1.2 \end{aligned}$$

General Form

An $m \times n$ systems of linear equations has the following general form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

- n unknowns variables $x_j, j = 1, \ldots, n$
- m linear equations
- $m \times n$ known coefficients $a_{ij} \in \mathbb{R}$
- m known constant terms b_i

If all $b_i = 0, i = 1, 2, \ldots, m$, then the system is called homogeneous!

Equivalent Matrix Form

$$Ax = b,$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Equivalent Matrix Form

$$Ax = b,$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Note:

- If a system $Ax = b$ has two distinct solutions, say x^* and $\hat{x}$. Then it has infinitely many solutions. Why?

Equivalent Matrix Form

$$Ax = b,$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Note:

- If a system $Ax = b$ has two distinct solutions, say x^* and $\hat{x}$. Then it has infinitely many solutions. Why? Think about $x_\alpha = \alpha x^* + (1 - \alpha)\hat{x}$ for any $\alpha \in \mathbb{R}$.

Equivalent Matrix Form

$$Ax = b,$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Note:

- If a system $Ax = b$ has two distinct solutions, say x^* and $\hat{x}$. Then it has infinitely many solutions. Why? Think about $x_\alpha = \alpha x^* + (1 - \alpha)\hat{x}$ for any $\alpha \in \mathbb{R}$.
- Thus, the system $Ax = b$ has either a unique solution or infinitely many solutions.
- $Ax = b$ has a unique solutions $\iff$ A is nonsingular $\iff$ $\det(A) \neq 0$

Outlines

Numerical Methods for solving systems of linear equations are classified into two categories:

- **Direct methods** provide the exact answer (assuming no round-off errors) in a finite number of steps. Direct methods include
  - Gauss-Elimination
  - Gauss-Jordan and Matrix Inversion
  - Triangular/LU factorization

- **Iterative methods** are not guaranteed to compute an exact solution in a finite number of steps, however, they yield high-quality approximate solutions and are easier to implement than the direct methods. Iterative methods include
  - Jacobi
  - Gauss-Seidel

Triangular systems

Example 1. Solve the system

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 1 \\ x_2 - x_3 &= 2 \\ 2x_3 &= 4 \end{aligned}$$

Apply backward substitution

$$\begin{aligned} 2x_1 &= 4 \\ x_1 - x_2 &= 2 \\ 3x_1 + 2x_2 + x_3 &= 1 \end{aligned}$$

Apply forward substitution

Gaussian Elimination

By elementary row operations, transform the system to a triangular system

$$Ax = b \quad \rightarrow \quad \bar{A}x = \bar{b} \quad (\bar{A} : \text{upper triangular matrix})$$

Elementary row operations:

- changing the order of rows;
- multiplying a row by a nonzero constant;
- replacing a row by its sum with a multiple of another row

Methodology

Set $A = [a_{ij}^{(1)}]$. Consider the system

$$M^{(1)} = [A|b] = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}^{(1)} & a_{m2}^{(1)} & \dots & a_{mn}^{(1)} & b_m^{(1)} \end{array}\right],$$

If $a_{11}^{(1)} \neq 0$, eliminate elements $a_{i1}^{(1)}, i = 2, \dots, m$ by turning them out to zero:

$$a_{i1}^{(2)} = a_{i1}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1j}^{(1)}$$

$$R_i^{(2)} = R_i^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} R_1^{(1)}, \quad i = 2, \dots, m$$

where R_i stands for row i.

$$M^{(2)} = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{m2}^{(2)} & \dots & a_{mn}^{(2)} & b_m^{(2)} \end{array}\right],$$

Assuming $a_{22}^{(2)} \neq 0$, use the second row to obtain zeros in rows $3, \dots, n$ of the second column

$$R_i^{(3)} = R_i^{(2)} - \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} R_2^{(2)}, \quad i = 3, \dots, m$$

Continue this to obtain an upper triangular system, solve it by the back substitution

$$M^{(n)} = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(2)} & b_1^{(2)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{mn}^{(n)} & b_m^{(n)} \end{array}\right] = [\bar{A}|\bar{b}].$$

Example 2.

$$\begin{aligned} x_1 + 2x_2 - 4x_3 &= -4 \\ 5x_1 + x_2 - x_3 &= -1 \\ 3x_1 - 2x_2 + 3x_3 &= 2 \end{aligned}$$

Gauss-Jordan

It's a modified Gaussian Elimination method. It uses elementary row operations in the augmented matrix in order to transform the matrix of coefficient into the identity matrix.

$$[A|b] \quad \rightarrow \quad [I|\bar{b}]$$

This approach can be used to invert a (nonsingular) matrix :

$$[A|I] \quad \rightarrow \quad [I|B] \quad (B = A^{-1})$$

Example 3. Apply Gauss-Jordan to solve the system

$$\begin{aligned} x_1 + 2x_2 - 4x_3 &= -4 \\ 5x_1 + x_2 - x_3 &= -1 \\ 3x_1 - 2x_2 + 3x_3 &= 2 \end{aligned}$$

LU factorization of a matrix A

Find a lower triangular matrix L and an upper triangular matrix U such that

$$A = LU$$

LU factorization of a matrix A

Find a lower triangular matrix L and an upper triangular matrix U such that

$$A = LU$$

The basic idea is to use left-multiplication of $A \in \mathbb{R}^{n\times n}$ by (elementary) lower triangular matrices $L_1, L_2, \cdots, L_{n-1}$ to convert A to an upper triangular matrix, i.e.,

$$\underbrace{L_{n-1}L_{n-2}\ldots L_2L_1}_{\tilde{L}}A = U$$

- The product of lower triangular matrices is a lower triangular matrix
- The inverse of a lower triangular matrix is also lower triangular
- $\tilde{L}A = U \quad \rightarrow \quad A = (\tilde{L})^{-1}U := LU$

$$\text{where} \quad (\tilde{L})^{-1} = (L_{n-1}L_{n-2}\ldots L_2L_1)^{-1} = L_1^{-1}\ldots L_{n-1}^{-1}$$

Example 4. Find the LU factorization of

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix}.$$

Answer. We choose L_1 such that left-multiplication corresponds to subtracting multiples of row 1 from the rows below such that the entries in the first column of A are zeroed out. Thus

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix}$$

Answer. We choose L_1 such that left-multiplication corresponds to subtracting multiples of row 1 from the rows below such that the entries in the first column of A are zeroed out. Thus

$$L_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix}$$

Next, we repeat this operation analogously for L_2 (in order to zero what is left in column 2 of the matrix on the right-hand side above):

$$L_2(L_1 A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix} = U$$

Now $L = (L_2 L_1)^{-1} = L_1^{-1} L_2^{-1}$ with

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \qquad L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

hence

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}$$

Double check your answer

$$U = L^{-1} A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix}$$

Why would we want to do this?

Consider the system

$$Ax = b, \qquad A = LU$$

Therefore we can perform (a now familiar) 2-step solution procedure:

- Solve the lower triangular system for y by forward substitution

$$Ly = b$$

- Solve the upper triangular system for x by back substitution

$$Ux = y$$

Advantage of LU factorization over Gauss elimination

LU decomposition is a better way to implement Gauss elimination, especially for repeated solving a number of equations with the same left-hand side. That is, for solving the equation $Ax = b$ with different values of b for the same A.

Note that in Gauss elimination the left-hand side (A) and the right-hand side (b) are modified within the same loop and there is no way to save the steps taken during the elimination process. If the equation has to be solved for different values of b, the elimination step has do to done all over again.

General Iterative Methods for Solving $Ax = b$

Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$

- Start with an initial guess $x^{(0)} \in \mathbb{R}^n$
- Choose the matrix C such that

$$\|I_n - CA\| < 1$$

- Produce a sequence of vectors $x^{(1)}, x^{(2)}, \ldots$ such that

$$x^{(k)} = x^{k-1} + C(b - Ax^{(k-1)}), \qquad k = 1, 2, \ldots$$

Note that

$$\begin{aligned} x^{(1)} &= x^{(0)} + C(b - Ax^{(0)}) \\ x^{(2)} &= x^{(1)} + C(b - Ax^{(1)}) \\ x^{(3)} &= x^{(2)} + C(b - Ax^{(2)}) \\ \vdots &= \vdots \end{aligned}$$

The sequence $x^{(k)}$ converges to the (approximate) solution.

Let x^* be the solution of $Ax = b$, thus $Ax^* = b$. Then

$$\begin{aligned} x^{(k)} - x^* &= x^{k-1} + C(b - Ax^{(k-1)}) - x^* \\ &= x^{k-1} + Cb - CAx^{(k-1)} - x^* \\ &= x^{k-1} + CAx^* - CAx^{(k-1)} - x^* \\ &= (x^{(k-1)} - x^*) - CA(x^{(k-1)} - x^*) \\ &= (I_n - CA)(x^{(k-1)} - x^*) \end{aligned}$$

Thus taking norms from the both sides

$$\|x^{(k)} - x^*\| \leq \|I_n - CA\| \|x^{(k-1)} - x^*\| \leq \|I_n - CA\|^k \|x^{(0)} - x^*\|$$

Since $\|I_n - CA\| < 1$ then

$$\|I_n - CA\|^k \to 0, \quad k \to \infty$$

Thus

$$x^{(k)} \to x^*, \quad k \to \infty$$

Jacobi Method to solve $Ax = b$

$$A = L + U + D$$

where

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1,n-1} & a_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} a_{11} & 0 & \dots & 0 & 0 \\ 0 & a_{22} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{nn} \end{bmatrix}$$

$$C = D^{-1} = \begin{bmatrix} 1/a_{11} & 0 & \dots & 0 & 0 \\ 0 & 1/a_{22} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1/a_{nn} \end{bmatrix}$$

Jacobi iterates:

$$x^{(k)} = x^{k-1} + D^{-1}(b - Ax^{(k-1)}), \qquad k = 1, 2, \dots$$

If A is **strictly dominant by rows or columns** then Jacobi method converges.

Definition

A matrix $A \in \mathbb{R}^{n\times n}$ is called strictly dominant by rows if

$$|a_{ii}| > \sum_{j=1, i\neq j}^{n} |a_{ij}|, \qquad i = 1, 2, \ldots, n$$

Then

$$\|I_n - D^{-1}A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1, i\neq j}^{n} \frac{|a_{ij}|}{|a_{ii}|} < 1$$

where

$$I_n - D^{-1}A = \begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \cdots & \frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix}.$$

Definition

A matrix $A \in \mathbb{R}^{n\times n}$ is called strictly dominant by columns if

$$|a_{jj}| > \sum_{i=1, i\neq j}^{n} |a_{ij}|, \qquad j = 1, 2, \ldots, n$$

Then

$$\|I_n - D^{-1}A\|_1 = \max_{1\leq j\leq n} \sum_{i=1, i\neq j}^{n} \frac{|a_{ij}|}{|a_{jj}|} < 1$$

where

$$I_n - D^{-1}A = \begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \cdots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \cdots & \frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix}.$$

Example 2. Consider a system $Ax = b$, where

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 8 \end{bmatrix}.$$

a) Is A diagonally dominant, by rows or columns?

b) Obtain the first three iterate obtained by Jacobi method with $x^{(0)} = [0, 0]^T$.

c) Evaluate the absolute error $\|x^{(k)} - x^*\|_\infty$ for these three iterates. What do you observe?

Jacobi.m

```
1  function [xJacobi, error] = Jacobi(A,b, x0, maxiter)
2  v=diag(A);
3  D = diag(v);
4  xsol= inv(A)*b;
5  x = x0;
6  error = zeros(maxiter+1,1);
7  error(1)= norm(x0-xsol,inf);
8  Iter = zeros (maxiter+1,1);
9  Iter(1) = 1;
10 for i = 1: maxiter
11     x = x + inv(D)*(b-A*x);
12     Error = norm(x-xsol,inf);
13     error(i+1) = Error;
14     Iter(i+1) = i+1;
15 end
16 xJacobi=x;
17 plot (error, Iter, '*')
18 end
19
```

Name	Value
A	[2,1;3,5]
b	[3;8]
error	21x1 doul
maxiter	20
x0	[0;0]
xJacobi	[1.0000;1.

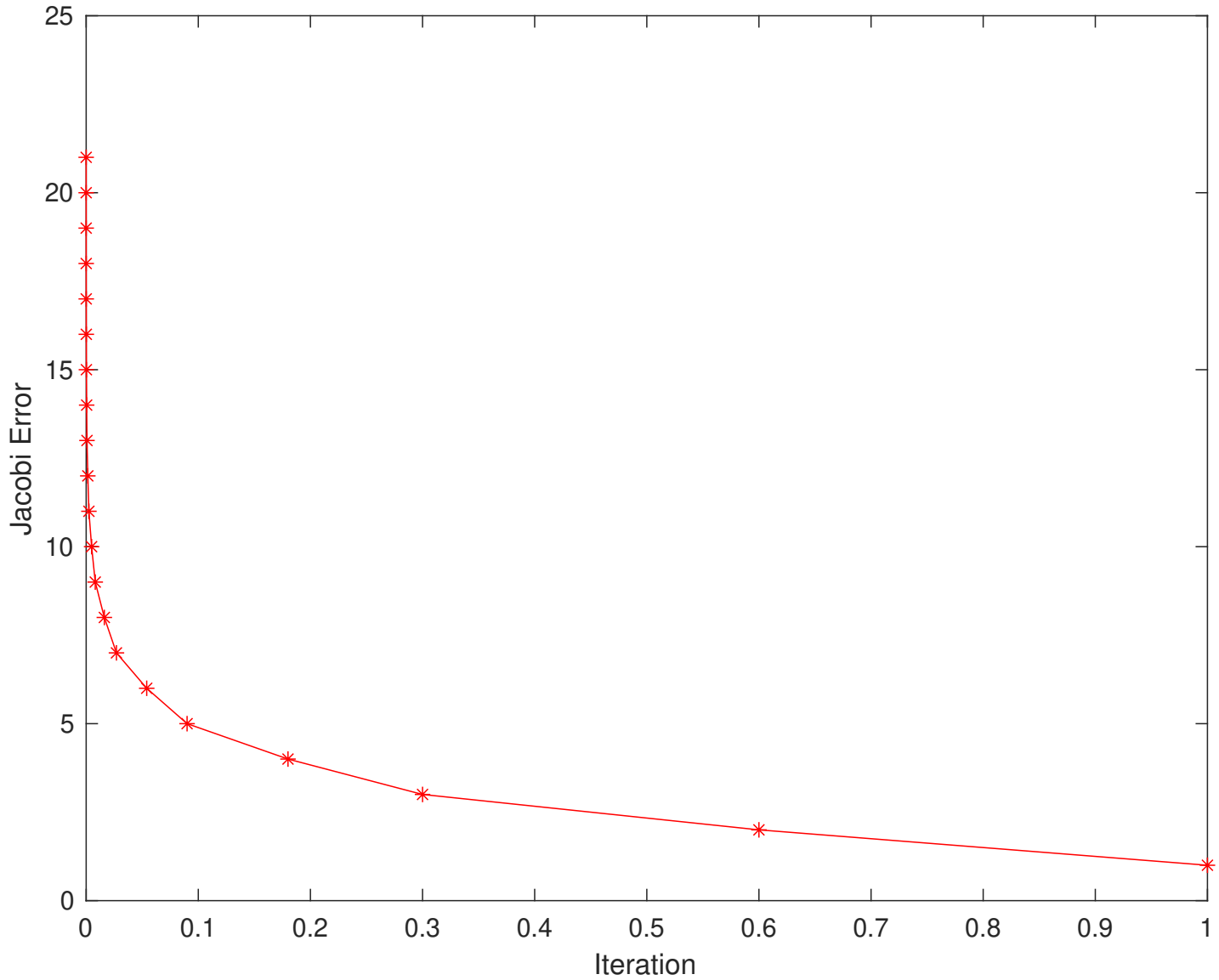
Command Window

```
>> A = [2 1;3 5]; b = [3;8]; x0 = [0;0]; maxiter = 20;
>> [xJacobi, error] = Jacobi(A,b, x0, maxiter)
```

Command Window

```
>> A = [2 1;3 5]; b = [3;8]; x0 = [0;0]; maxiter = 20;
>> [xJacobi, error] = Jacobi(A,b, x0, maxiter)

xJacobi =

   0.999994095100000
   0.999994095100000


error =

   1.000000000000000
   0.600000000000000
   0.300000000000000
   0.180000000000000
   0.090000000000000
   0.054000000000000
   0.027000000000000
   0.016200000000000
   0.008100000000000
   0.004860000000000
   0.002430000000000
   0.001458000000000
   0.000729000000000
   0.000437400000000
   0.000218700000000
   0.000131220000000
   0.000065610000000
   0.000039366000000
   0.000019683000000
   0.000011809800000
   0.000005904900000
```

Name	Value
A	[2,1;3,5]
b	[3;8]
error	*21x1 double*
maxiter	20
x0	[0;0]
xJacobi	[1.0000;1.0000


25
20
15
10
5
0
Jacobi Error
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Iteration

Gauss-Seidel Method to solve $Ax = b$

$$A = L + U + D$$

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

Gauss-Seidel iterates:

$$x^{(k+1)} = x^{(k)} + (L + D)^{-1}(b - Ax^k), \qquad k = 0, 1, 2, \ldots$$

Multiply both sides by $(L + D)$ to get

$$(L + D)x^{(k+1)} = (L + D - A)x^{(k)} + b, \qquad k = 0, 1, 2, \ldots$$

Then

$$Dx^{(k+1)} = -Lx^{(k+1)} - Ux^{(k)} + b, \qquad k = 0, 1, 2, \ldots$$

Gauss-Seidel, for $n = 3$

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} x^{(k+1)} = - \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{bmatrix} x^{(k+1)} - \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} x^{(k)} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

The above is equivalent to

$$\begin{aligned} a_{11} x_1^{(k+1)} &= -a_{12} x_2^{(k)} - a_{13} x_3^{(k)} + b_1 \\ a_{22} x_2^{(k+1)} &= -a_{21} x_1^{(k+1)} - a_{23} x_3^{(k)} + b_2 \\ a_{33} x_3^{(k+1)} &= -a_{31} x_1^{(k+1)} - a_{32} x_2^{(k+1)} + b_3 \end{aligned}$$

Hence computing $x_i^{(k+1)}$, $i \geq 2$ requires the values of

$$x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_{i-1}^{(k+1)}$$

Since the most latest values of the unknowns are used at the next stage of the iteration, Gauss-Seidel slightly performs faster the Jacobi method.

Again, consider a system $Ax = b$, with $A = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix}$, $b = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$.