

CS537

Numerical Analysis

Lecture 5

System of Linear Equations

Professor Jun Zhang

Department of Computer Science
University of Kentucky
Lexington, KY 40506-0046

October 9, 2008

System of Linear Equations

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

where a_{ij} are coefficients, x_i are unknowns, and b_i are right-hand sides. Written in a compact form is

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, \dots, n$$

The system can also be written in a matrix form

$$Ax = b$$

where the matrix is

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

and $x = [x_1, x_2, \dots, x_n]^T$, $b = [b_1, b_2, \dots, b_n]^T$

Gaussian Elimination

Linear systems are solved by Gaussian elimination, which involves repeated procedure of multiplying a row by a number and adding it to another row to eliminate a certain variable

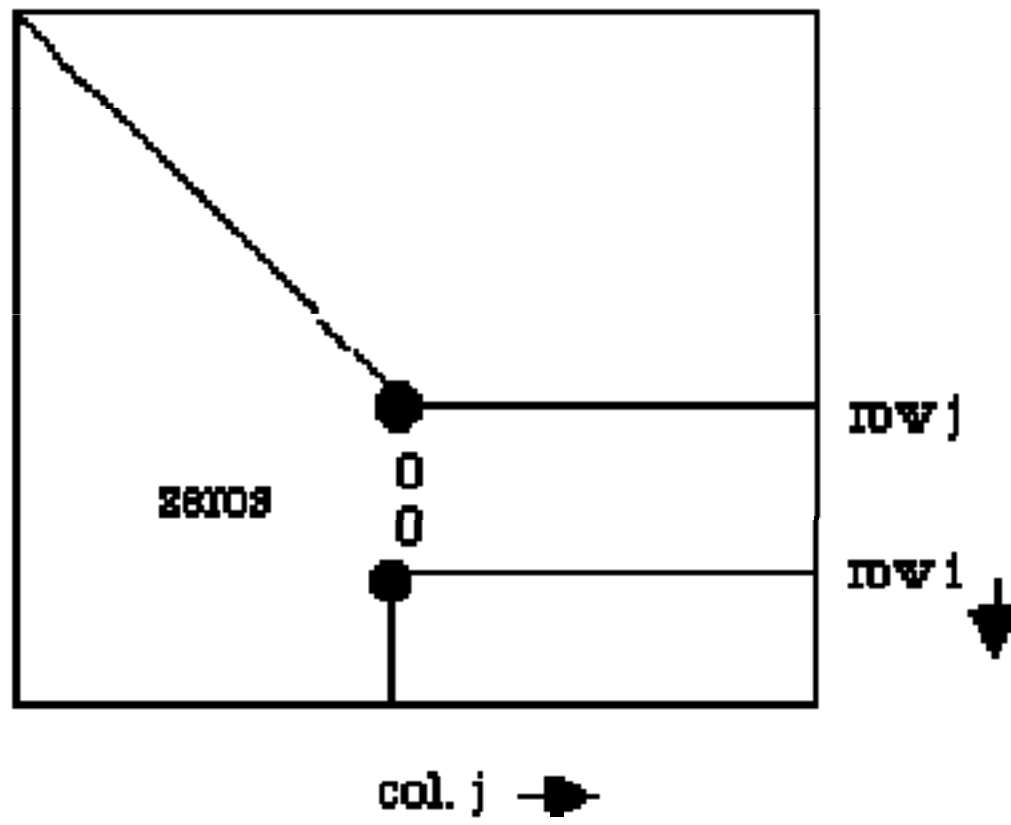
For a particular step, this amounts to

$$a_{ij} \leftarrow a_{ij} - \frac{a_{ik}}{a_{kk}} a_{kj} \quad (k \leq j \leq n)$$
$$b_i \leftarrow b_i - \frac{a_{ik}}{a_{kk}} b_k$$

After this step, the variable x_k , is eliminated in the $(k + 1)^{\text{th}}$ and in the later equations

The Gaussian elimination modifies a matrix into an upper triangular form such that $a_{ij} = 0$ for all $i > j$. The solution of an upper triangular is then easily obtained by a back substitution procedure

Illustration of Gaussian Elimination



Given a system of linear algebraic equations

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

Step 1: Each row times a_{11}/a_{k1} .
then use row one to subtract other rows.

$$\rightarrow \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & \bar{a}_{22} & \dots & \bar{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \bar{a}_{n2} & \dots & \bar{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_n \end{bmatrix}$$

Step 2: The second row and down multiply by $\bar{a}_{22}/\bar{a}_{k2}$.
then use row two to subtract every row below.

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & \bar{a}_{22} & \bar{a}_{23} & \dots & \bar{a}_{2n} \\ 0 & 0 & \bar{a}_{33} & \dots & \bar{a}_{3n} \\ \vdots & \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \bar{a}_{n3} & \dots & \bar{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ \vdots \\ \bar{b}_n \end{bmatrix}$$

Step 3: Similar to the previous two steps, repeat until all elements in the lower triangle of the matrix A become zero.

$$\Rightarrow \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & \bar{a}_{22} & \dots & \bar{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_n \end{bmatrix}$$

Back Substitution

The obtained upper triangular system is

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

$$a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2$$

$$a_{33}x_3 + \cdots + a_{3n}x_n = b_3$$

$$\vdots \quad \vdots \quad \vdots$$

$$a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1}$$

$$a_{nn}x_n = b_n$$

We can compute

$$x_n = \frac{b_n}{a_{nn}}$$

From the last equation and substitute its value in other equations and repeat the process

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

For $i = n - 1, n - 2, \dots, 1$

Example: The system of equations
$$\begin{cases} x + y + z = 3 \\ 2x + 3y + 7z = 0 \\ x + 3y - 2z = 17 \end{cases}$$
 has augmented matrix

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 2 & 3 & 7 & 0 \\ 1 & 3 & -2 & 17 \end{array} \right].$$

Row operations can be used to express the matrix in row-echelon form.

$$\begin{aligned} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 2 & 3 & 7 & 0 \\ 1 & 3 & -2 & 17 \end{array} \right] &\rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & 1 & 5 & -6 \\ 0 & 2 & -3 & 14 \end{array} \right] \\ &\rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & 1 & 5 & -6 \\ 0 & 0 & -13 & 26 \end{array} \right] \\ &\rightarrow \left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & 1 & 5 & -6 \\ 0 & 0 & 1 & -2 \end{array} \right] \end{aligned}$$

The system has become
$$\begin{cases} x + y + z = 3 \\ y + 5z = -6 \\ z = -2 \end{cases}$$
. By back-substitution we

find that $x = 1$, $y = 4$, and $z = -2$.

Condition Number and Error

A quantity used to measure the quality of a matrix is called condition number, defined as

$$\rho(A) = \|A\| \|A^{-1}\|$$

The condition number measures the transfer of error from the matrix A to the right hand side vector b . If A has a large condition number, small error in A may yield large error in the solution $x = A^{-1}b$. Such a matrix is called ill-conditioned

The error e is defined as the difference between a computed solution and the exact solution

$$e = x - \tilde{x}$$

Since the exact solution is generally unknown, we measure the residual

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

As an indicator of the size of the error

Small Pivot

$$\varepsilon x_1 + x_2 = 1$$

$$x_1 + x_2 = 2$$

For some small ε . After the step of Gaussian elimination

$$\varepsilon x_1 + x_2 = 1$$

$$\left(1 - \frac{1}{\varepsilon}\right)x_2 = 2 - \frac{1}{\varepsilon}$$

We have

$$x_2 = \frac{2 - 1/\varepsilon}{1 - 1/\varepsilon}$$

$$x_1 = \frac{1 - x_2}{\varepsilon}$$

For very small ε , the computer result will be $x_2 = 1$ and $x_1 = 0$. The correct results are

$$x_1 = \frac{1}{1 - \varepsilon} \approx 1$$

$$x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1$$

Scaled Partial Pivoting

We need to choose an element which is large relative to other elements of the same row as the pivot

Let $L = (l_1, l_2, \dots, l_n)$ be an index array of integers. We first compute an array of scaling factor as $S = (s_1, s_2, \dots, s_n)$ where

$$s_i = \max_{1 \leq j \leq n} |a_{ij}| \quad (1 \leq i \leq n)$$

The first row i is choosing such that the ratio $|a_{i,l_1}|/s_i$ is the greatest. Suppose this index is l_1 , then appropriate multipliers of equation l_1 are subtracted from the other equations to eliminate x_1 from the other equations

Suppose initially $L = (l_1, l_2, \dots, l_n) = (1, 2, \dots, n)$, if our first choice is l_j , we will interchange l_j and l_1 in the index set, not actually interchange the first and the l_j rows, to avoid moving data around the memory

Example

Straightforward Gaussian elimination does not work well (not robust)

$$\varepsilon x_1 + x_2 = 1$$

$$x_1 + x_2 = 2$$

The scale factor will be computed as $S = \{1, 1\}$. In the first step, the scale factor ratio array $\{\varepsilon, 1\}$. So the 2nd row is the pivoting row

After eliminating x_1 from the 1st equation, we have

$$(1 - \varepsilon)x_2 = 1 - 2\varepsilon$$

$$x_1 + x_2 = 2$$

It follows that

$$x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1$$

$$x_1 = 2 - x_2 \approx 1$$

We computed correct results by using scaled partial pivoting strategy

Gaussian Elimination with Partial Pivoting

$$\begin{array}{l}
 \left[\begin{array}{cccc} 1 & 2 & 3 & 1 \\ 2 & 3 & 5 & 0 \\ 3 & 4 & 5 & 0 \end{array} \right] \begin{array}{l} (1) \\ (2) \\ (3) \end{array} \rightarrow \left[\begin{array}{cccc} 3 & 4 & 5 & 0 \\ 2 & 3 & 5 & 0 \\ 1 & 2 & 3 & 1 \end{array} \right] \begin{array}{l} (3) \\ (2) \\ (1) \end{array} \\
 \rightarrow \left[\begin{array}{cccc} 3 & 4 & 5 & 0 \\ 0 & 1/3 & 5/3 & 0 \\ 0 & 2/3 & 4/3 & 1 \end{array} \right] \begin{array}{l} (1) \\ (2) - (2/3)(1) \\ (3) - (1/3)(1) \end{array} \\
 \rightarrow \left[\begin{array}{cccc} 3 & 4 & 5 & 0 \\ 0 & 2/3 & 4/3 & 1 \\ 0 & 1/3 & 5/3 & 0 \end{array} \right] \begin{array}{l} (1) \\ (3) \\ (2) \end{array} \\
 \rightarrow \left[\begin{array}{cccc} 3 & 4 & 5 & 0 \\ 0 & 2/3 & 4/3 & 1 \\ 0 & 0 & 1 & -\frac{1}{2} \end{array} \right] \begin{array}{l} (1) \\ (2) \\ (3) - (1/2)(2) \end{array}
 \end{array}$$

Long Operation Count

We count the number of multiplications and divisions, ignore summations and subtractions

The 1st step, finding a pivoting costs n divisions

Additional n operations are needed to multiply a factor to the pivoting row for each of the $n - 1$ eliminations. The cost is $n(n - 1)$ operations. The total cost of this step is n^2 operations

The computation is repeated on the remaining $(n - 1)$ equations. The total costs of Gaussian elimination with scaled partial pivoting is

$$n^2 + (n-1)^2 + \dots + 4^2 + 3^2 + 2^2 =$$
$$\frac{n(n+1)(2n+1)}{6} - 1 \approx \frac{n^3}{3}$$

Back substitution costs $n(n - 1)/2$ operations

Tridiagonal and Banded Systems

Banded system has a coefficient matrix such that $a_{ij} = 0$ if $|i - j| \geq w$. For a tridiagonal system, $w = 2$

$$\begin{bmatrix}
 d_1 & c_1 & & & & & \\
 a_1 & d_2 & c_2 & & & & \\
 & a_2 & d_3 & c_3 & & & \\
 & & \vdots & \vdots & \vdots & & \\
 & & & a_{n-1} & d_{n-1} & c_{n-1} & \\
 & & & & a_{n-1} & d_n & \\
 \end{bmatrix}
 \begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 \vdots \\
 x_{n-1} \\
 x_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 b_1 \\
 b_2 \\
 b_3 \\
 \vdots \\
 b_{n-1} \\
 b_n
 \end{bmatrix}$$

General elimination procedure

$$d_i \leftarrow d_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) c_{i-1}$$

$$b_i \leftarrow b_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) b_{i-1}$$

The array c_i is not modified. No additional nonzero is created

Matrix can be stored in three vector arrays

Tridiagonal Systems

The back substitution is straightforward

$$x_n = \frac{b_n}{d_n}$$
$$x_i = \frac{b_i - c_i x_{i+1}}{d_i} \quad (i = n-1, \dots, 1)$$

No pivoting is performed, otherwise the procedure will be quite different due to the fill-in

Diagonal dominance: A matrix $A = (a_{ij})_{n \times n}$ is diagonally dominant if

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad (1 \leq i \leq n)$$

For a diagonally dominant tridiagonal system, no pivoting is needed, i.e., no division by zero

We want to show Gaussian elimination preserves diagonal dominance, i.e.,

$$|d_i| > |a_{i-1}| + |c_i|$$

Tridiagonal Systems

The new coefficient matrix has 0 elements at the a_i 's places. The new diagonal elements are determined recursively as

$$\hat{d}_1 = d_1$$
$$\hat{d}_i = d_i - c_{i-1} \left(\frac{a_{i-1}}{\hat{d}_{i-1}} \right) \quad (2 \leq i \leq n)$$

We assume that

$$|d_i| > |a_{i-1}| + |c_i|$$

We want to show that

$$|\hat{d}_i| > |c_i|$$

We use induction to prove the inequality

It is obviously true for $i=1$, as

$$\hat{d}_1 = d_1$$

Tridiagonal Systems

If we assume that

$$|\hat{d}_{i-1}| > |c_{i-1}|$$

We prove for index i , as

$$\begin{aligned} |\hat{d}_i| &= \left| d_i - c_{i-1} \frac{a_{i-1}}{\hat{d}_{i-1}} \right| \\ &\geq |d_i| - |a_{i-1}| \frac{|c_{i-1}|}{|\hat{d}_{i-1}|} \\ &> |a_{i-1}| + |c_i| - |a_{i-1}| = |c_i| \end{aligned}$$

It follows that the new diagonal entries will not be zero, the Gaussian elimination procedure can be carried out without any problem

Example of a pentadiagonal matrix. It is nearly tridiagonal. The matrix with only nonzero entries on the main diagonal, and the first two diagonals above and below it

$$\begin{pmatrix} c_1 & d_1 & e_1 & 0 & \dots & \dots & 0 \\ b_1 & c_2 & d_2 & e_2 & \ddots & & \vdots \\ a_1 & b_2 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & a_2 & \ddots & \ddots & \ddots & e_{n-3} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & d_{n-2} & e_{n-2} \\ \vdots & & \ddots & a_{n-3} & b_{n-2} & c_{n-1} & d_{n-1} \\ 0 & \dots & \dots & 0 & a_{n-2} & b_{n-1} & c_n \end{pmatrix}.$$

LU Factorization 1

As we showed before, an $n \times n$ system of linear equations can be written in a matrix form as

$$Ax = b$$

where the coefficient matrix A has the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

x is the unknown vector and b is right-hand side known vector

We also assume A is of full rank, and most entries of A are not zero

LU Factorization 2

There are two special forms of matrices. One is called *lower triangular*

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix}$$

The other is *upper triangular*

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

We want to find a pair of L and U matrices, such that

$$A = LU$$

Example

Take a system of linear equations

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 26 \\ -19 \\ -34 \end{bmatrix}$$

The Gaussian elimination process finally yields an upper triangular system

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

This could be achieved by multiplying the original system with a matrix M , such that

$$MAx = Mb$$

Example

We want the matrix M to be special, so that MA is upper triangular

$$MA = \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} = U$$

The question is: can we find such a matrix M ? Look at the first step of the Gaussian elimination

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -27 \\ -18 \end{bmatrix}$$

This step can be achieved by multiplying the original system with a lower triangular matrix

$$M_1 Ax = M_1 b$$

Example

Here the lower triangular matrix M_1 is

$$M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

This matrix is nonsingular, because it is lower triangular with a main diagonal containing all 1's.

The nonzero elements in the first column are the negatives of the multipliers in the positions where 0's were created in the original matrix

$$M_1Ax = M_1b$$

Example

If we continue to the second step of the Gaussian elimination, we have

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -21 \end{bmatrix}$$

This can be achieved with the multiplication of another lower triangular matrix

$$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix}$$

Thus, we have

$$M_2 M_1 A x = M_2 M_1 b$$

Example

The last step is

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

with another lower triangular matrix

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

And, we have

$$M_3 M_2 M_1 A x = M_3 M_2 M_1 b$$

Example

If we define

$$M = M_3 M_2 M_1$$

Then M is lower triangular and MA is upper triangular

Since

$$M_3 M_2 M_1 A = U$$

the inverse of a lower triangular is again a lower triangular matrix. The product of two lower triangular matrices is a lower triangular matrix

$$A = M^{-1}U = M_1^{-1}M_2^{-1}M_3^{-1}U = LU$$

This shows that we have transformed a matrix A into the product of a lower triangular matrix and an upper triangular matrix

This process is called LU factorization or LU decomposition

Example

The lower triangular matrix is

$$L = M_1^{-1} M_2^{-1} M_3^{-1}$$

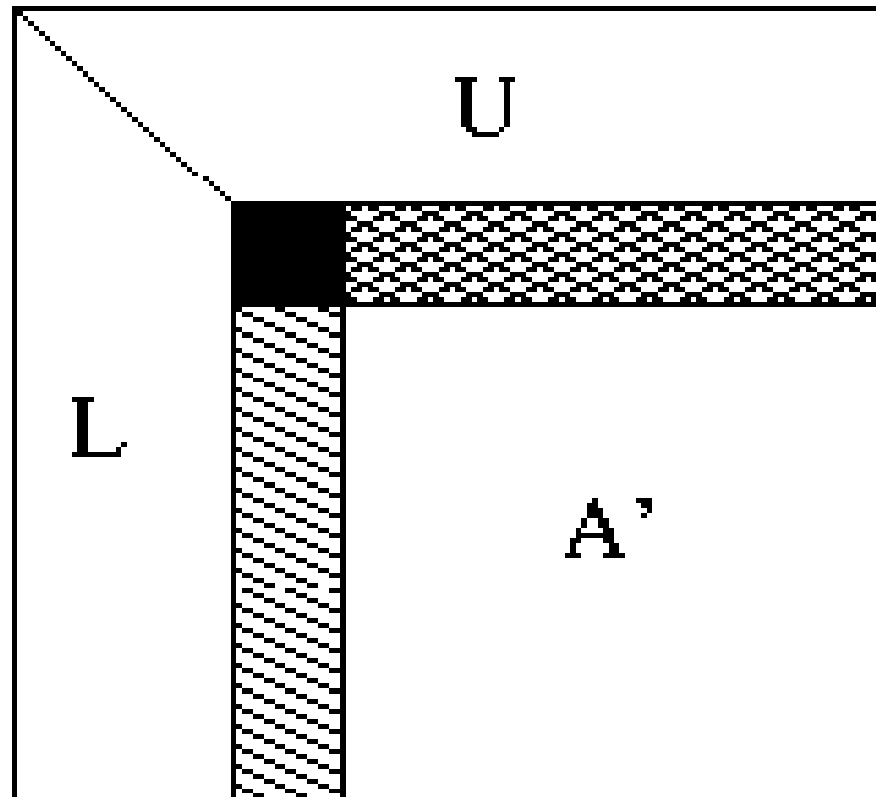
Which in the current example is

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix}$$

And, we have

$$LU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} = A$$

Illustration of LU Factorization



Example

The original system

$$Ax = b$$

can be factored into

$$LUx = b$$

The solution process can be separated into two phases. First, forward elimination with an intermediate variable z , as

$$Lz = b$$

Second, back substitution step

$$Ux = z$$

The advantage of LU factorization or LU decomposition is that several linear systems with the same coefficient matrix but different right hand side vector can be solved more efficiently. We only need do the factorization once. The forward elimination and back substitution steps are $O(n^2)$ operations

Compute the Inverse

The system of linear equations

$$Ax = b$$

can be symbolically solved as

$$x = A^{-1}b$$

But the inverse of A is seldom computed explicitly. This is because performing Gaussian elimination is less expensive.

If the explicit inverse of a matrix A is needed, it can be obtained by using LU factorization of A , note that

$$AX = I$$

We can compute a series of linear systems as

$$LU[x^{(1)}, x^{(2)}, \dots, x^{(n)}] = [I^{(1)}, I^{(2)}, \dots, I^{(n)}]$$

And

$$A^{-1} = [x^{(1)}, x^{(2)}, \dots, x^{(n)}]$$

Singular Value Decomposition (SVD)

The eigenvalues and eigenvectors of a matrix A , are

$$Ax = \lambda x$$

I.e., the application of A on the vector of x is equivalent to a scaling. Here λ is an eigenvalue and x is the associated eigenvector

The singular values of a matrix A are the nonnegative square roots of the eigenvalues of $A^T A$

By the Spectral Theorem for Matrices, $A^T A$ can be diagonalized by an orthogonal matrix, Q , as

$$A^T A = Q \Sigma Q^{-1}$$

where

$$Q^T Q = Q^T Q = I, \quad \text{i.e.,} \quad Q^T = Q^{-1}$$

The diagonal matrix Σ contains the eigenvalues of $A^T A$ on its diagonal

Singular Value Decomposition II

We can see that

$$A^T A Q = Q \Sigma$$

So the columns of Q are eigenvectors of $A^T A$

If λ is an eigenvalue of $A^T A$ and x is a corresponding eigenvector, then

$$A^T A x = \lambda x$$

And

$$\|Ax\|^2 = (Ax)^T (Ax) = x^T A^T A x = x^T \lambda x = \lambda \|x\|^2$$

It follows that the eigenvalue λ is real and nonnegative.

Since Q is an orthogonal matrix, its columns form an orthonormal base.

They are unit eigenvectors of $A^T A$

If v_j is the j th column of Q, we have

$$A^T A v_j = \lambda_j v_j$$

Iterative Solution Methods

Obtain an approximate solution to a linear system iteratively

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ &\dots \quad \dots \quad \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_{nn} &= b_n \end{aligned}$$

Create a single equation for each variable

$$\begin{aligned} x_1 &= \frac{b_1 - (a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n)}{a_{11}} \\ x_2 &= \frac{b_2 - (a_{21}x_1 + a_{23}x_3 + \cdots + a_{2n}x_n)}{a_{22}} \\ &\dots \quad \dots \\ x_{nn} &= \frac{b_n - (a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots)}{a_{nn}} \end{aligned}$$

We assign an arbitrary value to each of the variables in the right hand side to start iteration

Jacobi Iteration

We repeatedly use each set of compound values as the next iterates, and hope the process will converge to the true solution. (May stop early if necessary.) This iteration procedure is called Jacobi iteration method. Convergence is not guaranteed for general matrices

$$\begin{aligned}x_1^{(k)} &= \frac{b_1 - (a_{12}x_2^{(k-1)} + a_{13}x_3^{(k-1)} + \dots + a_{1n}x_n^{(k-1)})}{a_{11}} \\x_2^{(k)} &= \frac{b_2 - (a_{21}x_1^{(k-1)} + a_{23}x_3^{(k-1)} + \dots + a_{2n}x_n^{(k-1)})}{a_{22}} \\&\dots \quad \dots \\x_{nn}^{(k)} &= \frac{b_n - (a_{n1}x_1^{(k-1)} + a_{n2}x_2^{(k-1)} + a_{n3}x_3^{(k-1)} + \dots)}{a_{nn}}\end{aligned}$$

Gauss-Seidel Iteration

If the iteration process converges, it may be faster if we use the new values as soon as they are available. This gives the Gauss-Seidel iteration method

$$\begin{aligned}x_1^{(k)} &= \frac{b_1 - (a_{12}x_2^{(k-1)} + a_{13}x_3^{(k-1)} + \dots + a_{1n}x_n^{(k-1)})}{a_{11}} \\x_2^{(k)} &= \frac{b_2 - (a_{21}x_1^{(k)} + a_{23}x_3^{(k-1)} + \dots + a_{2n}x_n^{(k-1)})}{a_{22}} \\&\dots \quad \dots \\x_{nn}^{(k)} &= \frac{b_n - (a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + a_{n3}x_3^{(k)} + \dots)}{a_{nn}}\end{aligned}$$

In general, Gauss-Seidel is faster than Jacobi, but the later is more attractive to run on parallel computers. No convergence is guaranteed

SOR Method

Around 1950, David Young at Harvard University found that iteration can be accelerated by using a weighted average of the successive approximate solutions. Given some parameter $0 < \omega < 2$, we perform

$$x_i^{(k)} = \omega \tilde{x}_i^{(k)} + (1 - \omega)x_i^{(k-1)} \quad i = 1, 2, \dots, n$$

This step symbolizes the beginning of modern iterative methods

With an arbitrary initial guess, convergence of these basic iterative methods is guaranteed if

- 1.) The matrix is strictly diagonally dominant;
- 2.) The matrix is symmetric positive definite, i.e., $A = A^T$ and $\mathbf{x}^T A \mathbf{x} > 0$

Splitting Coefficient Matrix

The system of linear equations is

$$Ax = b$$

We can split the coefficient matrix A as

$$A = M - N$$

Such that M is nonsingular and M^{-1} is easy to compute. So we have

$$(M - N)x = b$$

and

$$Mx = Nx + b$$

So

$$x = M^{-1}(Nx + b) = M^{-1}Nx + M^{-1}b$$

We can use this relationship for an iteration scheme

$$x^{(k)} = M^{-1}Nx^{(k-1)} + M^{-1}b$$

Iteration Matrix

The matrix $M^{-1}N$ is called the iteration matrix

For the iteration process to be inexpensive to carry out, M must be easy to invert

$$x^{(k)} = M^{-1}Nx^{(k-1)} + M^{-1}b$$

Since

$$N = M - A$$

We have

$$x^{(k)} = M^{-1}(M - A)x^{(k-1)} + M^{-1}b = (I - M^{-1}A)x^{(k-1)} + M^{-1}b$$

If $x^{(k)}$ converges to the solution x , then

$$x = (I - M^{-1}A)x + M^{-1}b$$

It follows that

$$Ax = b$$

Convergence

We define the error vector at each iteration step as

$$e^{(k)} = x^{(k)} - x$$

It follows that

$$\begin{aligned} e^{(k)} = x^{(k)} - x &= (I - M^{-1}A)x^{(k-1)} - x + M^{-1}b \\ &= (I - M^{-1}A)x^{(k-1)} - (I - M^{-1}A)x \\ &= (I - M^{-1}A)(x^{(k-1)} - x) \\ &= (I - M^{-1}A)e^{(k-1)} \end{aligned}$$

If we want the error to become smaller and smaller, we want the iteration matrix

$$I - M^{-1}A$$

to be small, in some sense

Iteration Matrix

So we want the matrix

$$M^{-1}A$$

to be close to the identity matrix. It is to say that we want M is close to A

If fact, if all eigenvelues of the iteration matrix

$$I - M^{-1}A$$

is smaller than one, then the iteration

$$e^{(k)} = (I - M^{-1}A)e^{(k-1)}$$

converges to zero and the original iteration converges. This requires

$$\rho(I - M^{-1}A) < 1$$

I.e., the spectral radius of the iteration is smaller than one guarantees the convergence of the induced iteration method

Stationary Iterative Methods

For the Jacobi iteration

$$A = D - (L + U)$$

For the Gauss-Seidel Method

$$A = (L + D) - U$$

For the SOR method, the iteration is

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

These iterative methods are called stationary methods, or classic iterative methods. They converge slowly

Modern iterative methods, such as multigrid method and Krylov subspace methods converge much more rapidly. Those methods will be studied in CS623