Application of MATLAB in Solving Linear Equations

Huilin Deng1, Lifeng Li2*, Xiaohu Zhang2

1School of Science, Guizhou University of Engineering Science, Bijie, Guizhou, 551700, CHINA
2School of Architecture and Civil Engineering, Guizhou University of Engineering Science, Bijie, Guizhou, 551700, CHINA
*Corresponding author’s e-mail:lilifengbj@126.com

Abstract. In order to cultivate the interest of science and engineering students in learning the direct and iterative methods of solving linear equations, this paper takes the OBE education concept as the guiding ideology and summarizes and discusses how to solve linear equations with MATLAB relevant instructions.

1. Introduction
With the development of science and technology, the depth of the teaching concept OBE and the popularity of MATLAB software, direct method and iterative method of solving linear equations are commonly used. Wang[1] studied the iterative solution method of solving linear equations based on MATLAB software. Tong and Qin [2] classified the linear equations, discussed four frequently used solutions, and solved the linear equations by combining examples with MATLAB software. Liu [3] analysed several common solutions of linear equations by application of MATLAB software. Hao [4] explored two iterative solutions and their application in MATLAB. By practicing the concept of OBE, this paper discusses and analyses the application of MATLAB software in solving linear equations, and pays attention to the simplicity and efficiency of solving linear equations, so as to cultivate students interest in learning and avoid panic when they encounter large-scale linear equations with data.

2. Direct solution of linear equations and MATLAB related instructions
The exact solution of linear equations without iteration can be obtained through finite step arithmetic operation. The general form of the linear equations is as follows:

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

(1)

here \( X=(x_1, x_2, ..., x_n)^T \), \( b=(b_1, b_2, ..., b_n)^T \).

The matrix \( B=A,b \) of \( A \) and \( b \) is called the augmented matrix of the system of equations. If \( b=0 \), (1) is called homogeneous linear system of equations; If \( b \neq 0 \), (1) is called the inhomogeneous
linear equations. Discrimination and structure of solution of the linear equations is as follows. If the rank of the matrix $A$ satisfies $R(A) < n$, then there is a non-zero solution; If $R(A) = n$, there is only a zero solution. In addition,

1. Since $R(A) = R(B) = n$, (1) is just the definite system of equations and have a unique solution;
2. Since $R(A) = R(B) < n$, (1) is the underdetermined system of equations and have an infinite solution;
3. Since $R(A) < R(B)$, (1) is over determinate or contradictory. In general, (1) is no solution but the least squares solution can be obtained.

2.1 Cramer law

If the determinant of the coefficient matrix of the linear system of equations (1) satisfies $D = |A| \neq 0$, then (1) has a solution and the solution is unique as follows:

$$
x_1 = \frac{d_1}{D}, x_2 = \frac{d_2}{D}, \ldots, x_n = \frac{d_n}{D}
$$

where $D_j (j = 1, 2, \ldots, n)$ is the determinant of a matrix that replaces the first column of the matrix with the constant term $b$ of the system.

MATLAB instruction det(A) to find the determinant of matrix $A$, and then we get the system (1) by Cramer’s law

2.2 Direct matrix division

Just definite system of equations is a system of equations whose number of equations is equal to the number of solutions required, that is, the coefficient equation of just definite system of equations is a square matrix. In the matrix equation $Ax = B$, MATLAB provides a convenient command to divide by $"A/B", i.e. direct matrix division. When the square matrix $A$ is a non-singular matrix, the inverse matrix product method can be used, that is, inv(A)*b. When the coefficient matrix $A$ is a singular matrix or $A$ non-singular matrix, the command operation can get different results. We get the nonsingular matrix $A$ when $|A| \neq 0$; It can be known that matrix $A$ has an inverse matrix. We can use the left division $"/"$ instruction to find the solution of just definite system of equations, or inv(A)*b method to find the solution of just definite system of equations. When singular matrix $A$ is obtained, MATLAB will display warning information using the left division $"/"$ command, and NaN is given.

2.3 LU decomposition method

LU decomposition method, which is known as Gaussian elimination method, can decompose a sparse matrix into the product of lower triangular matrix $L$ and upper triangular matrix $U$ since $A=LU$. In MATLAB, LU decomposition has complete and incomplete decomposition. This section mainly analyzes complete LU decomposition. The instruction used is lu in MATLAB as follows:

- $[L, U] = lu(A)$
- $[L, U, P] = lu(A)$
- $[L, U, P, Q] = lu(A)$
- $[L, U, P, Q, R] = lu(A)$
- $[\ldots] = lu(A, 'vector')$

here (1) The input matrix $A$ needs to be a square matrix.

(2) When the output is in $[L, U]$ format, $L$ is the unit lower triangular matrix, $U$ is the upper triangular matrix, $lu(A)$ is the lu decomposition of the matrix $A$, and satisfies the condition that $A=LU$.

(3) When the output is in $[L, U, P, Q]$ format, $U$ is the upper triangular matrix, $L$ is the unit lower triangular matrix, $P$ is the permutation matrix, and satisfies $LU=PA$. 

MATLAB instruction det(A) to find the determinant of matrix $A$, and then we get the system (1) by Cramer’s law $x_1 = \frac{d_1}{D}, x_2 = \frac{d_2}{D}, \ldots, x_n = \frac{d_n}{D}$.
(4) When output is in the format \([L,U,P,Q,R]\), \(L\) is the unit lower triangular matrix, \(U\) is the upper triangular matrix, \(P\) is the row rearrangement matrix, \(Q\) is the column rearrangement matrix, and satisfies \(P*A*Q=L*U\).

(5) When output using \([...]=lu(A, 'vector')\), specify that the returned value is in vector form.

### 2.4 Cholesky decomposition

The solution method is a decomposition method that divides the positive definite matrix into the product of upper and lower triangular matrices. In MATLAB, the decomposition method has complete and incomplete decomposition. In MATLAB, the instruction used for complete decomposition of matrix \(A\) is \(\text{chol}\). When \(\text{chol}\) is used for complete decomposition, it is best to obtain all eigenvalues of the matrix through the instruction \(\text{eig}\) first, and check whether the eigenvalues are positive or not. The instruction \(\text{chol}\) is used as follows:

\[
\begin{align*}
R &= \text{chol}(A) \\
L &= \text{chol}(A, 'lower') \\
[R, p] &= \text{chol}(A) \\
[L, p] &= \text{chol}(A, 'lower') \\
[R, p] &= \text{chol}(A, 'upper') \\
[R, p, S] &= \text{chol}(A) \\
[R, p, s] &= \text{chol}(A, 'vector')
\end{align*}
\]

Here (1) When using \(R=\text{chol}(A)\) format, matrix \(R\) is the upper triangular matrix generated by the diagonal and upper triangle of matrix \(A\), which conforms to \(R' * R = A\).

(2) When using the format \(L=\text{chol}(A, 'lower')\), matrix \(L\) is the lower triangular matrix generated by the diagonal and lower triangle of matrix \(A\), which conforms to \(L' * L = A\).

(3) When \([R, p]=\text{chol}(A)\) format is used, if matrix \(A\) is a positive definite matrix, then \(p=0\) and \(R\) is the decomposition factor; if matrix \(A\) is a nonpositive definite matrix, then \(p\) is a positive integer and \(R\) is an ordered upper triangular matrix.

(4) When using \([L, p]=\text{chol}(A, 'lower')\) format, if matrix \(A\) is a positive definite matrix, then \(p=0\) and \(L\) is the lower triangular matrix; if matrix \(A\) is a nonpositive definite matrix, then \(p\) is a positive integer and \(L\) is an ordered lower triangular matrix.

(5) When \([R, p]=\text{chol}(A, 'upper')\) format is used, if matrix \(A\) is a positive definite matrix, then \(p=0\) and \(R\) is an upper triangular matrix; if matrix \(A\) is a nonpositive definite matrix, then \(p\) is a positive integer and \(R\) is an ordered upper triangular matrix.

(6) When \([R, p, S]=\text{chol}(A)\) format is used, if matrix \(A\) is a sparse matrix, \(S\) is the transpose matrix returned; when, \(R\) is the upper triangular matrix, and \(R' * R = S' * A * S\); when, \(R\) is the upper triangular matrix of \(q * n\).

(7) When \([R, p, s]=\text{chol}(A, 'vector')\) format is used, if matrix \(A\) is a sparse matrix, vector transformation information \(s\) is returned, so that \(A(s, s)=R' * R\).

### 3. Iterative solution of linear equations and MATLAB related instructions

If the coefficient matrix is a nonsingular coefficient matrix, then the system has a unique solution. The square matrix \(A\) of this system of equations is decomposed into \(A=C-D\). If square matrix \(C\) is a nonsingular matrix, then substitute \(A\) into the equation and get

\[
(C - D)x = b
\]

\[
\Rightarrow C^{-1}(C - D)x = C^{-1}b
\]

\[
\Rightarrow x = Mx + g, M = C^{-1}D, \quad g = C^{-1}b
\]

Based on this, the iterative formula can be constructed:

\[
x_{k+1} = Mx_k + g
\]  

(3)

\(M\) is called an iterative matrix. \(M\) is a kind of approximation and the non-singular matrix, which can be called the split matrix. After the split matrix selection is not the same, we can get different iterative method. There are two common iterative methods described below.
3.1 Jacobi iterative method

If the coefficient matrix $A$ of the system (1) is not singular and the matrix $A$ can be decomposed into:

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

where $D$ is a diagonal matrix, $L$ is a strictly lower triangular matrix, and $U$ is a strictly upper triangular matrix, then diagonal matrix

$$D = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn})$$

and

$$L = \begin{bmatrix}
0 & \cdots & 0 & 0 \\
-a_{21} & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
-a_{n1} & \cdots & 0 & 0
\end{bmatrix},
U = \begin{bmatrix}
0 & \cdots & -a_{n1} & 0 \\
0 & \cdots & \ddots & \vdots \\
\vdots & \ddots & 0 & 0 \\
0 & \cdots & 0 & 0
\end{bmatrix}$$

Then push the iteration format by $Ax = b$ as follows

$$x = D^{-1}(L + U)x + b \Rightarrow x_{k+1} = D^{-1}(L + U)x_k + D^{-1}b$$

$$M = D^{-1}(L + U)$$ is called the Jacobi iteration matrix. The component form of Jacobi iteration is

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{n} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k+1)} \right), \quad i = 1, 2, \ldots, n$$

(5)

3.2 Gauss-Seidel iteration method

Gauss-Seidel iteration is similar to Jacobi iteration. The difference between them is that Jacobi iteration only uses the last iteration value for each iteration, while Gauss-Seidel iteration makes full use of the latest iteration value for each iteration.

Assume that linear equations $Ax = b$, $A = D - L - U$, then we get

$$D - L - U)x = b \Rightarrow (D - L)x = Ux + b \Rightarrow x = (D - L)^{-1}Ux + (D - L)^{-1}b$$

Based on this, the iterative formula can be constructed

$$x_i^{(k+1)} = (D - L)^{-1}Ux^{(k+1)} + (D - L)^{-1}b, \quad k = 0, 1, 2, \ldots$$

(6)

here $M = (D - L)^{-1}$ is called Gauss-Seidel iteration matrix. Its component form is:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right), \quad i = 1, 2, \ldots, n$$

(7)

3.3 QR decomposition

QR decomposition is the orthogonal decomposition of matrix. QR decomposition is the product of a matrix $A$ ($m \times n$) into an orthogonal matrix $Q$ ($m \times n$) and an upper triangular matrix $R$ ($m \times n$), $A = QR$. QR decomposition gives QR instruction in MATLAB. Instruction $qr$ is used as follows:

$$[Q, R] = qr(A)$$
$$[Q, R] = qr(A, 0)$$
$$[Q, R, E] = qr(A)$$
$$[Q, R, E] = qr(A, 0)$$
$$[Q, R, E] = qr(A, \text{vector})$$

$R = qr(A)$
$$[C, R] = qr(A, b)$$
$$R = qr(A, 0)$$
$$[C, R] = qr(A, b, 0)$$

(1) When using $[Q, R] = qr(A)$ format, an orthogonal matrix $Q$ and upper triangular matrix $R$ of the same order as matrix $A$ will be generated. $Q$ and $R$ satisfy $A = QR$.

(2) When the format $[Q, R] = qr(A, 0)$ is used, the economic orthogonal decomposition of matrix $A$ is generated. If the matrix is $m \times n$ order and $m > n$, the first $n$ columns of $Q$ can only be calculated.

(3) When $[Q, R, E] = qr(A)$ format is used, a permutation matrix $E$, orthogonal matrix $Q$ and upper triangular matrix $R$ can be generated, so that $AE = QR$ can be satisfied.

(4) When the format $[Q, R, E] = qr(A, 0)$ is used, matrix $E$ is the displacement vector, whose function is to make the diagonal elements of matrix $R$ descending, and $Q^*R = A(:, E)$.

(5) When using $[Q, R, E] = qr(A, \text{vector})$ format, it is not a matrix of replacement information but
return as the carrier.

(6) When R=qr(A) format is used, the sparse matrix A is decomposed to produce only one upper triangular matrix R, which satisfies R '*R=A' *A. The operation of A '*A by this method can reduce the loss of internal digital information.

(7) When R=qr(A,0) format is used, the economic decomposition of sparse matrix A is carried out.

(8) When the format [C, R]=qr(A,b,0) is used, the economic decomposition of the sparse least squares problem is carried out.

4. Conclusion
The solution to the linear equation summarized above is not directly output through MATLAB, but if the input process goes wrong, the solution is wrong. Compared with manual calculation, MATLAB Calculation can obtain the results effectively and accurately.

Acknowledgements
This work is supported by undergraduate teaching quality improvement project of Guizhou University of Engineering Science in 2018 (No.2018JG130 and No.2018JG119), the Science and Technology Foundation of Guizhou Province (No.LH[2014]7535), and the Guizhou University of Engineering Science Doctor's Fund (No.G2018016).

Reference
[1] Wang X. (2014) Iterative solution of linear equations based on MATLAB. Journal of Wuyi university, 33: 6-9.
[2] Tong X., Qin X. (2013) Discussion on solving linear equations and MATLAB implementation. Mathematics learning and research, 21: 86-88.
[3] Liu C. (2015) Linear equations and MATLAB implementation. Journal of natural sciences of Harbin normal university, 31:34-36.
[4] Hao Y. (2017) Jacobi iterative method and Gauss-Seidel iterative method. Journal of Shanxi Datong university, 33: 3-5.