MATLAB Marina: Linear Algebraic Equations

Student Learning Objectives

After completing this module, one should:

1. Be able to perform arithmetic operations on matrices using MATLAB including: addition, subtraction, multiplication, transpose, and inverse.
2. Be able to solve systems of linear equations using the matrix form of the equations and MATLAB.

Terms

vector, matrix, identity matrix, matrix addition, matrix subtraction, matrix multiplication, matrix transpose, matrix inverse, system of linear equations, nonsingular matrix, determinant, Gaussian elimination (back division)

MATLAB Functions, Keywords, and Operators

+, -, *, ‘, \, inv, det, cond, eye

Matrices

A matrix like an array is a multi-dimensional collection of data of the same data type. Matrices are treated the same as arrays in MATLAB. Matrices are created, accessed, and modified in the same way that arrays are. Generally, when performing matrix operations on an array, such as matrix multiplication or matrix inverse, the array will be referred to as a matrix rather than an array.

The two-dimensional matrix C below has n rows and m columns (n times m total elements). The element in row p and column q is referred to as the pq element of the matrix.

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & & c_{2m} \\ \vdots & & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{pmatrix}$$

One dimensional matrices are typically referred to as vectors. A 1 x m matrix is a row vector and an m x 1 matrix is a column vector.

$$v_{row} = \left[v_1,\ v_2,\ \cdots\ v_m\right]$$

$$v_{col} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

The Identity matrix, denoted by I_n, is a square matrix with ones along its diagonal and zeros everywhere else. The MATLAB function `eye` will create an identity matrix. The statement eye(M) creates a M by M identity matrix and eye(N,M) creates a N by M matrix. The N by M matrix is not an identity matrix as it is not square (unless N = M).

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
>> eye(2,3)
ans =
     1     0     0
     0     1     0
```

Figure 1. MATLAB eye Function

Operations on Matrices

Most built in MATLAB functions will take matrices as arguments since matrices are arrays. Generally, MATLAB functions perform their operation on each element in the argument and return a result the same size as the argument. The length and size functions work for matrices just as for arrays.

Matrix Equality

Two matrices A and B are equal if the matrices have the same dimensions and all their corresponding entries are equal. The two matrices A and B below

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

are equal if $a_{11} = b_{11}$, $a_{12} = b_{12}$, $a_{21} = b_{21}$, and $a_{22} = b_{22}$.

The MATLAB equality operator == performs element by element comparison for equality and so it is not a matrix equality operator. The element by element equality can be combined with the all function to implement a matrix equality operation. For matrices, the default for the all function is to operate along the columns but an optional argument allows one to specify to operate along all the dimensions and thus produce a scalar result.

Matrix Addition, Subtraction, and Multiplication

Matrix addition and subtraction are defined for matrices of the same dimensions. Matrix addition and subtraction are defined the same as element by element addition and subtraction. There is not a separate dot operator for the element by element versions of addition and subtraction.

Matrix addition is commutative but matrix subtraction is not. The sum (or difference) of two matrices is a matrix of the same size whose entries consist of the sum (difference) of the two corresponding entries of the two matrices. For the 2 x 2 matrices A and B below

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

The sum and difference of the matrices are

$A+B = \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{pmatrix}$ and $A-B = \begin{pmatrix} a_{11}-b_{11} & a_{12}-b_{12} \\ a_{21}-b_{21} & a_{22}-b_{22} \end{pmatrix}$

```
A = [0.0, 5.0;
     2.0, -4.0];
B = [0.0, 5.0;
     2.0, -4.0];
C = [0.0, 1.0;
     -1.0, 1.0];

AeqB_log = all(A == B,[1,2])
AeqB_log =
  logical 1

AeqC_log = all(A == C,[1,2])
AeqC_log =
  logical 0
```

Figure 2. Matrix Equality

Matrix multiplication is defined for matrices where the size of the inner dimensions match, i.e. the number of columns of first matrix equals the numbers of rows of the second matrix. An $m \times p$ matrix A can be multiplied by a $q \times n$ matrix B if $p = q$. The result of multiplying an $m \times p$ matrix A by a $p \times n$ matrix B is a $m \times n$ matrix whose ij entry is the ith row of the matrix A multiplied by the jth column of matrix B. Matrix multiplication is not commutative.

The product of a $1 \times n$ row vector and a $n \times 1$ column vector is a 1×1 matrix (a scalar)

$$\begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n = \sum_{k=1}^{n} v_k w_k$$

The product of two 2×2 matrices is a 2×2 matrix

$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$, then $A * B = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$

Matrix multiplication is not the same operation as element by element multiplication; for matrix multiplication one uses the * operator. Multiplying a matrix by an appropriately sized identity matrix does not change the matrix, i.e. the result is the original matrix.

Matrix power, ^, is defined only for square matrices and the power must be scalar value. Matrix power is not the same operation as element by element power; A^n equivalent to performing matrix multiplication of the matrix n-1 times.

```
A = [0.0, 5.0;
    2.0, -4.0];
B = [1.0, 5.0;
    -1.0, 0.0];
C = [0.0, 5.0, 1.0;
    2.0, -4.0, 0.0];

AplusB = A + B
AplusB =
     1    10
     1    -4

AminusB = A - B
AminusB =
    -1     0
     3    -4

AtimesC = A * C
AtimesC =
    10   -20     0
    -8    26     2

Apow4 = A^4
Apow4 =
   260  -720
  -288   836
```

Figure 3. Matrix Arithmetic Operations

Matrix division is not defined although MATLAB will allow one to divide matrices since MATLAB makes no distinction between a matrix and an array for arithmetic operations. Matrix division using forward, /, and back division, \, should generally not be used as they do not correspond to legal linear algebraic matrix operations. One exception to this is solving the matrix form Ax = b of a system of linear equations using Gaussian elimination.

Matrix Transpose

The transpose of matrix changes the rows of a matrix to columns and the columns of a matrix to rows. The conjugate transpose operator is a single quote and is a postfix operator and takes the conjugate of all complex valued elements as well as performing the transpose operation. The non-conjugate transpose operator is a period followed by a single quote and only performs the transpose operation. When the matrix consists of only real valued numbers it does not matter which transpose operator you use but it is good style to use the conjugate transpose only when the conjugate of values is intended.

The transpose of a row vector is a column vector and the transpose of a column vector is a row vector. The transpose of a 2×3 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

is a 3×2 matrix $A' = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{pmatrix}$

```
A = [0.0, 5.0, 1.0;
    2.0, -4.0, 0.0];
Atranspose = A'
Atranspose =
     0     2
     5    -4
     1     0
```

Figure 4. Matrix Transpose

Matrix Inverse

The inverse of a matrix A written A^{-1} is a matrix such that $AA^{-1} = A^{-1}A = I$, where I is the identity matrix. The inverse of a matrix is only defined for square matrices that have full rank (determinant is non-zero). MATLAB has built in functions `det` and `inv` for computing the determinant and inverse of a matrix.

```
A = [0.0, 5.0;
    2.0, -4.0];
Adet = det(A)
Adet =
   -10
Ainverse = inv(A)
Ainverse =
    0.4000    0.5000
    0.2000         0
A*inv(A)
ans =
     1     0
     0     1
```

Figure 5. Matrix Inverse

Solving Systems of Linear Equations

A system of n linear equations in n unknowns can be written in the matrix form Ax = b, where A is a n by n square matrix of coefficients, x is an n by 1 column vector of unknowns, and b is a n by 1 column vector of constants. The coefficient matrix A of the matrix form of the system of linear equations is square when the number of equations and unknowns are equal. If the coefficient matrix A is square and has full rank (determinant is nonzero) then the equations are independent and the system of linear equations has a unique solution.

If the number of independent equations is greater than the number of unknowns the system is said to be over constrained and there may not be a solution. If the number of independent equations is less than the number of unknowns the system is said to be under constrained and there are an infinite number of solutions. The coefficient matrix A of the matrix form of the system of linear equations is non-square in these cases and the coefficient matrix is thus not invertible.

The system of linear equations (two equations in two unknowns)

$3v - w = 4$

$v + w = 0$

Has the matrix form

$$\begin{pmatrix} 3 & -1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

and has the unique solution, v = 1, w = -1.

To solve the system of equations numerically, the equations should be arranged so any constants are on the right-hand side of the equation and the unknowns appear in the same order in each equation. The column vector of unknowns x should be ordered the same as the unknowns in each equation. Each row of the A matrix and b vector will consist of the coefficients for the unknowns in one of the equations and the equation constant; the order the equations are used does not matter.

Systems of linear equations in matrix form can be solved using MATLAB in two ways: multiplying on the left by the matrix inverse $x = A^{-1}b$ or using Gaussian elimination $x = A \backslash b$.

The system of three equations in three unknowns

$2u - 3v + 4w = 9$

$u + 2v + 3w = 1.5$

$-7u - v + 2w = -12$

has the matrix form

$$\begin{pmatrix} 2 & -3 & 4 \\ 1 & 2 & 3 \\ -7 & -1 & 2 \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 9 \\ 1.5 \\ -12 \end{pmatrix}$$

The solution to this system of linear equations is: u = 2.0, v = -1.0, and w = 0.5 and was determined using the MATLAB code of Figure 6. The solution was determined using both the matrix inverse and Gaussian elimination although typically one or the other would be used.

```
A = [2, -3, 4;
    1, 2, 3;
    -7, -1, 2];
b = [9; 1.5; -12];
x = inv(A)*b
x =
    2.0000
   -1.0000
    0.5000
xG = A\b
xG =
    2.0000
   -1.0000
    0.5000
```

Figure 6. Solving System of Linear Equations

When solving systems of linear equations numerically, one must be careful that the problem is well posed. The condition number of a matrix is a measure of the sensitivity of the solution of a system of linear to errors in the equation coefficients. The condition number provides an indication of whether the inverse of the A matrix and the solution of the system of linear equations will be accurate. Condition numbers near one indicate a well condition matrix (the condition number of the identity matrix is one) and large condition numbers indicate that the matrix is near to being singular (and the numerical matrix inverse may not be accurate). See MATLAB's help on the `cond` function for more information.

For systems of equations where the A matrix is singular (condition number is very large), MATLAB will return either an array of `NaN` values or an array of `Inf` and `-Inf` values if a solution is attempted using Gaussian elimination.

Last modified Friday, September 25, 2020