Department of Electrical Engineering
University of Arkansas

A Tutorial on Matlab

Ch. 2 Matrix Operations

Dr. Jingxian Wu

wuj@uark.edu

OUTLINE

- Vectors
- Matrices
- Matrix operations
- Exercises

VECTORS AND MATRICES

- **Vectors and matrices are fundamental to Matlab**
  - Matlab is designed specifically for vector and matrix operations.
  - Direct vector and matrix operations are usually much faster compared to process scalars with loops
  - Try to use vector and matrix as often as possible
    - minimize the use of loops (loops are slow in Matlab)

VECTORS

- **Vector**
  - Define a row vector (use , as the separator)

```
>> a = [3, 1, 2, 8]          % [ ] indicates a vector
a =
    3    1    2    8
```

  - Define a column vector (use ; as the separator)

```
>> b = [3;1;2;8]
b =
    3
    1
    2
    8
```

  - Transpose operator .’ (a dot followed by an apostrophe)

```
>> b.'
ans =
    3    1    2    8
```

VECTORS

- **More ways of definition vectors**
  - Starting from 1, ends at 10, with a step size 2

```
>> c = [1:2:10]        % the bracket is optional
c =
    1     3     5     7     9
```

  - Starting from 5, ends at 8, with a step size 1

```
>> d = 5:8
d =
    5     6     7     8
```

VECTORS

- **More ways of defining vectors (Cont'd)**
  - Starting from 6, ends at 0, with a step size -2

```
>> e = [6:-2:0]
    e =
        6     4     2     0
```

  - Generate 5 equally spaced points between 2 and 20

```
>> linspace(2, 20, 5)
    ans =
                2.0000    6.5000   11.0000   15.5000   20.0000
```

VECTORS

- **Vector indexing**

  Assume a = [3 1 2 8]

  – The 3rd element of a vector

  ```
  >> a(3)
  ans =
      2
  ```

  – The 2, 3, 4th elements (vectors can be used as indices)

  ```
  >> a(2:4)    % similarly a(2:end) since a only has 4 elements
  ans =
      1    2    8
  ```

VECTORS

- **Vector indexing**

  Assume a = [3 1 2 8]

  – The 2nd and 4th elements

  ```
  >> a(2:2:4)
  ans =
      1  8
  ```

  – The 1st and 4th elements (be careful about the bracket, which is used to indicate a vector)

  ```
  >> a([1,4])
  ans =
      3  8
  ```

OUTLINE

- Vectors
- **Matrices**
- Matrix operations
- Exercises

MATRICES

- **Define a matrix**
  - Define a 2 x 3 matrix

```
>> a = [2, 5, 7; 1, 4, 9]
a =
    2    5    7
    1    4    9
```

  - Matrix transpose .’

```
>> a.'
ans =
    2    1
    5    4
    7    9
```

MATRICES

- **Matrix indexing**

$$a = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}$$

  - access the (1, 3)-th element (1st row, 3rd column)

```
>> a(1, 3)
ans =
     7
```

  - access the first row

```
>> a(1, :)
ans =
   2     5     7
```

  - access the 2nd column

```
>> a(:, 2)
ans =
     5
     4
```

MATRICES

- **Matrix indexing**

$$a = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}$$

  - Access the 2nd and 3rd element on the 2nd row

```
>> a(2, [2,3])
ans =
    4    9
```

  - Access the 1st and 3rd columns

```
>> a(:, [1, 3])
ans =
    2    7
    1    9
```

  - In general

    a(row_index, column_index)

MATRICES

- **Matrix indexing**

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

  - Create a submatrix that contains the 2nd and 3rd rows, and 1st and 3rd columns

```
>> b([2, 3], [1, 3])
ans =
    4    6
    7    9
```

  - Exchange the 1st and 3rd column

```
>> b(:, [3, 2, 1])
ans =
    3    2    1
    6    5    4
    9    8    7
```

MATRIX

- **Build a matrix with vectors**
  - Build a matrix with row vectors

```
>> v1 = [1 3 5];
>> v2 = [2 4 6];
>> m = [v1; v2] % semicolon separates rows
m =
    1    3    5
    2    4    6
```

  - Build a matrix with column vectors

```
>> c1 = [1; 2];  % semicolon separates rows (a column vector with 2 rows)
>> c2 = [3; 4];
>> c3 = [5; 6];
>> m2 = [c1, c2, c3]   % comma separates columns ( a matrix with 3 columns)
m2 =
    1    3    5
    2    4    6
```

OUTLINE

- Vectors
- Matrices
- **Matrix operations**
- Exercises

MATRIX OPERATIONS

- **Arithmetic**

$$\mathrm{m1} = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}, \mathrm{m2} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, v1 = [7 \quad 8 \quad 9], v2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix},$$

  - Summation and subtraction (matrix of vector size must match)

```
>> m1-m2
ans =
     1     3     4
    -3    -1     3

>> v1+v2
Error using  +
Matrix dimensions must agree.

>> v1+v2.'
ans =
     8    10    12
```

MATRIX OPERATIONS

- **Arithmetic**

$$\mathrm{m1} = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}, \mathrm{m2} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, v1 = [7 \quad 8 \quad 9], v2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix},$$

– Element wise multiplication: .* (dot followed by star)

```
>> m1.*m2
ans =
    2   10   21
    4   20   54
```

– Element wise division: ./ (dot followed by slash)

```
>> v1./v2
Error using  ./
Matrix dimensions must agree.
>> v1./v2.'
ans =
    7   4   3
```

MATRIX OPERATION

- **Arithmetic**

$$m1 = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}, m2 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, m3 = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix}$$

  - Matrix multiplication (the inner dimension of the matrices must match)

```
>> m1*m2  % m1 has 3 columns, m2 has 2 rows, inner dimension mismatch
Error using  *
Inner matrix dimensions must agree.
>> m1*m3 % m1 has 3 columns, m3 has 3 rows
ans =
    31    30    23
    31    33    20
```

  - Inverse of a square matrix

```
>> inv(m3)
ans =
   -0.2778    0.3889    0.0556
    0.0556   -0.2778    0.3889
    0.3889    0.0556   -0.2778
```

MATRIX OPERATIONS

- **Arithmetic**

$$\mathrm{m1} = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}, \mathrm{m2} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, v1 = [7 \quad 8 \quad 9],$$

  - Add v2 to the 2nd row of m1, and assign the result to a new matrix mat1

```
>> mat1 = m1;
>> mat1(2, :) = mat1(2, :) + v1
mat1 =
    2    5    7
    8   12   18
```

  - Replace the 3rd column of m2 with the 1st column of m1, and assign the result to a new matrix mat2

```
>> mat2 = m2;
>> mat2(:, 3) = m1(:, 1)
mat2 =
    1    2    2
    4    5    1
```

MATRIX OPERATION

- **Operation between a matrix and a scalar**

$$m1 = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}$$

  - Elementwise operations with the scalar

```
>> m1+2
ans =
    4    7    9
    3    6   11

>> m1*2
ans =
    4   10   14
    2    8   18

>> m1/2
ans =
    1.0000    2.5000    3.5000
    0.5000    2.0000    4.5000
```

MATRIX OPERATION

- **Transpose (.') v.s. complex transpose (')**

$$\text{m4} = \begin{bmatrix} 2+i & 5+2i & 7+3i \\ 1 & 4 & 9 \end{bmatrix}$$

– Transpose operator: .'

```
>> m4.'
ans =
   2.0000 + 1.0000i   1.0000
   5.0000 + 2.0000i   4.0000
   7.0000 + 3.0000i   9.0000
```

– Complex transpose operator: ' (transpose and complex conjugate)

```
>> m4'
ans =
   2.0000 - 1.0000i   1.0000
   5.0000 - 2.0000i   4.0000
   7.0000 - 3.0000i   9.0000
```

MATRIX OPERATIONS

- **Logic operators**

$$m1 = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}, m5 = \begin{bmatrix} 2 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix}$$

  - Example: extract all elements in m1 that is > 4

```
>> logic_mat = (m1 > 4)
logic_mat =
    0    1    1
    0    0    1
>> m1(logic_mat)
ans =
    5
    7
    9
% or equivalently
>> m1(m1>4)
ans =
    5
    7
    9
```

MATRIX OPERATIONS

- **Dimension of a matrix**

$$m1 = \begin{bmatrix} 2 & 5 & 7 \\ 1 & 4 & 9 \end{bmatrix}$$

  - size( ): returns the dimension of a matrix

```
>> size(m1)
ans =
    2     3

>> [m,n] = size(m1)
m =
    2
n =
    3
```

MATRIX OPERATIONS

- **Diagonal of a matrix**

$$\text{m3} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix}, v1 = [7 \quad 8 \quad 9],$$

  - diag(a): returns the main diagonal of the matrix a

>> diag(m3)

ans =

1

1

1

  - diag(v1): creates a diagonal matrix with vector v1 on its diagonal
  - >> diag(v1)

ans =

7	0	0
0	8	0
0	0	9

MATRIX OPERATIONS

- **Generate special matrices**
  - All-one matrix

```
>> ones(2, 3)
ans =
     1     1     1
     1     1     1
```

  - All-zero matrix

```
>> zeros(1, 2)
ans =
     0     0
```

  - Identity matrix

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

OUTLINE

- Vectors
- Matrices
- Matrix operations
- **Exercises**

EXERCISES

- 1. Create a row vector that starts with -10, ends at 18, with a step size of 4
- 2. Create a column vector that starts with 6, ends at -9, with a step size -3
- 3. Create a row vector that starts with 1, ends at 90, and has exactly 20 elements
- 4. Create the following three matrices

$$A = \begin{bmatrix} 2 & 3 & 7 \\ 1 & 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, C = \begin{bmatrix} 1+i & 2 \\ 3 & 4+i \\ 5 & 6 \end{bmatrix},$$

- 5. Calculate A + B, B + 2, A - transpose(C), element-wise A.*B and A./B
- 6. Is A*B valid? Is A*transpose(B) valid?

EXERCISES

- 7. Use matrix indexing access the following elements:
  - The element on the 2nd row and 3rd column of A
  - The 2nd row of B
  - The 3rd column of A
  - The elements on the 2nd row and 2nd and 3rd columns of A
  - A new 2x2 matrix D, the 1st column of D is the 2nd column of A, and the 2nd column of D is the 3rd column of B

- 8. Use a logic matrix to indicate the locations where the elements of A and B are the same.

- 9. Find the elements of B that is less than or equal to 3