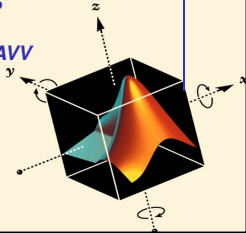


MATLAB Basic Operations

-Dr. G. L. Prajapati, IET-DAVV



Special characters

- `>>` % default command prompt
- `%` % comment - MATLAB simply ignores anything to the right of this sign (till the end of the line).
- `>> % my comment`
- `;` % semicolon at the end of the line will prevent MATLAB from echoing the information you type on the screen.

```
>> a=20
a =
    20
>> b=20;
```

Creating Variables

- Matlab as a calculator:
- `>>2+5`
- `>>7*10+8`
- `>>5^2`
- 'ans' - "answer", used in MATLAB as the default variable.

```
>> 2+5
ans =
     7
>> ans
ans =
     7
```

Defining Your Own Variables

- When Matlab comes across a new variable name - it automatically creates it.
- Begins with a LETTER, e.g., A2z.
- Can be a mix of letters, digits, and underscores (e.g., **vector_A**, but not **vector-A**)
- Not longer than 31 characters.
- No spaces
- Different mixes of capital and small letters = different variables.

For example: "A_VaRIAbLe", "a_variable", "A_VARIABLE", and "A_variabLe"

- `>> String='this is a string'`

Listing & Clearing Variables

```
>> a=10
>> b = 20
>> the_average = (a + b) / 2
>> whos
```

Name	Size	Bytes	Class
a	1x1	8	double array
b	1x1	8	double array
the_average	1x1	8	double array

Grand total is 3 elements using 24 bytes

```
>> clear, clear all %clear variables from memory
```

Creating vectors

Row separator: space/coma (,)

```
>> a=[1 2 3]
a =
     1     2     3
>> a2=[1, 2, 3]
a2 =
     1     2     3
```

Column separator: Semicolon (:)

```
>> a4=[1; 2; 3]
a4 =
     1
     2
     3
```

Creating sequences:

- From : jump: till
- `linespec(X1, X2, N)` generates N points between X1 and X2.

```
>> a3=1:1:3
a3 =
     1     2     3
```

Creating Matrices

```
>> My_matrix=[1 2; 3 4]
My_matrix =
     1     2
     3     4
```

- Matrices must be rectangular.
- Creating random matrices:


```
>> rand_mat=randi(4,2,4)
rand_mat =
     0.9501     0.6048     0.8913     0.4565
     0.2311     0.4840     0.7621     0.0185
```

 2-by-4 random matrix
(2 rows and 4 columns).

Creating Matrices

- You can combine existing vectors as matrix elements:


```
>> mat_from_vec=[a, 2*a]
mat_from_vec =
     1     2     3
     2     4     6
```
- You can combine existing matrices as matrix elements:


```
>> mat_from_mat=[mat_from_vec, 2*mat_from_vec]
mat_from_mat =
     1     2     3     2     4     6
     2     4     6     4     8    12
```

Indexing Into a Matrix

		1	2	3	4	5
A -	1	4	10	1	6	2
	2	8	1.2	9	4	25
	3	7.2	5	7	1	11
	4	0	0.5	4	5	5.6
	5	23	03	13	0	10

A(1:5,6) A(1:end,end)
 A(:,5) A(:,end)
 A(21:29) A(21:end)
 A(4:5,2:3)
 A([9 14; 10 19])

- Use () parentheses to specify index
- colon operator (:) specifies range / ALL
- [] to create matrix of index subscripts
- end specifies maximum index value

```
>> B=A(3,1);
>> A(:,end)=[1;7;3;8;4];
```

- The row number is first, followed by the column number.

Linear Algebra Operations

Matrix Operators	Array operators	Common Matrix Functions	
() parentheses		inv()	matrix inverse
' comp. transpose	.' array transpose	det()	determinant
^ power	.^ array power	rank()	matrix rank
* multiplication	.* array mult.	eig()	eigenvectors & values
/ division	./ array division	svd()	singular value dec.
\ left division		norm()	matrix / vector norm
+ addition			
- subtraction			

(In order of precedence)

```
>> help ops
>> help matfun
```

Matrix Multiplication

- Inner dimensions must be equal
- Dimension of resulting matrix = outermost dimensions of multiplied matrices
- Resulting elements = dot product of the rows of the 1st matrix with the columns of the 2nd matrix

```
>> a = [1 2 3 4; 5 6 7 8];
>> b = ones(4,3);
>> c = a*b
```

[2x4] [4x3] → [2x3]

10	10	10
26	26	26

← a(2nd row), b(3rd column)

Vectors - (dot) Multiplication

The size of A and B must be the same or either could be a scalar.

```
>> A = [2 3 4];
>> B = [1 2 3];
>> A .* B
ans =
     2     6    12
```

Note: the result is the product of the individual elements of the vectors.

Creating a Matrix: Using eye, ones and zeros



Arrays can be constructed from built-in Matlab functions
(see Prataap pp. 70-71)

```
>> A = eye(3)           (eye(n) constructs an n x n "identity" matrix)
```

```
A =  
    1    0    0  
    0    1    0  
    0    0    1
```

```
>> A = ones(3,2)       (ones(m,n) constructs an m x n array of 1's)
```

```
A =  
    1    1  
    1    1  
    1    1
```

```
>> A = zeros(2,3)      (zeros(m,n) constructs an m x n array of 0's)
```

```
A =  
    0    0    0  
    0    0    0
```