

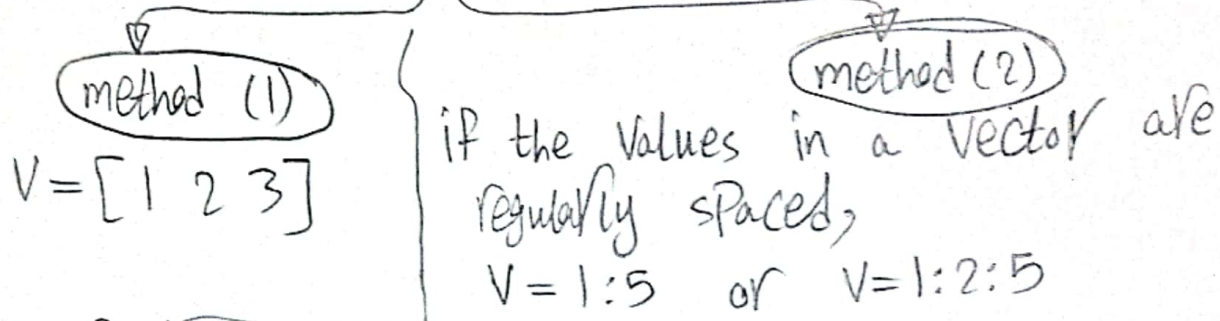
Outline

- Double arrays:
 - Creating it.
 - getting its metadata.
- Logical arrays:
 - Creating it
 - getting its metadata.
- Types of array indexing.
- Operations on the arrays.

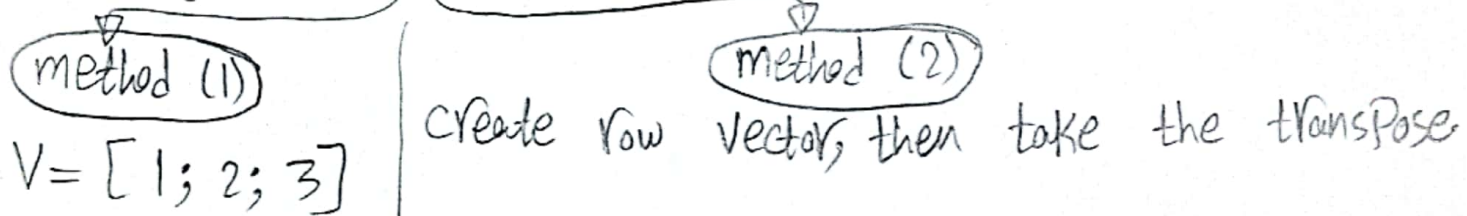
Double arrays

- In matlab, array means vector (row vector or column vector) or matrix.

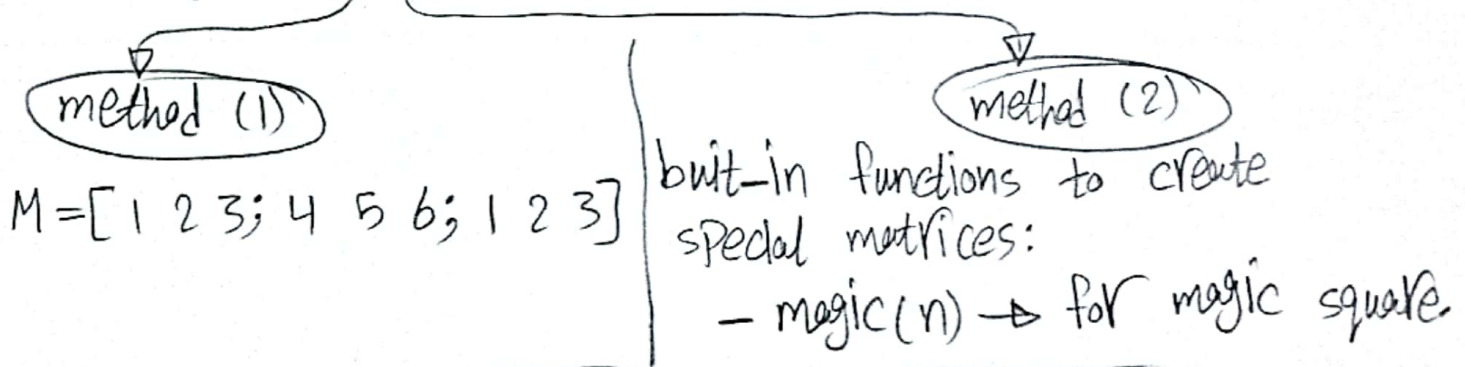
• Creating row vectors:



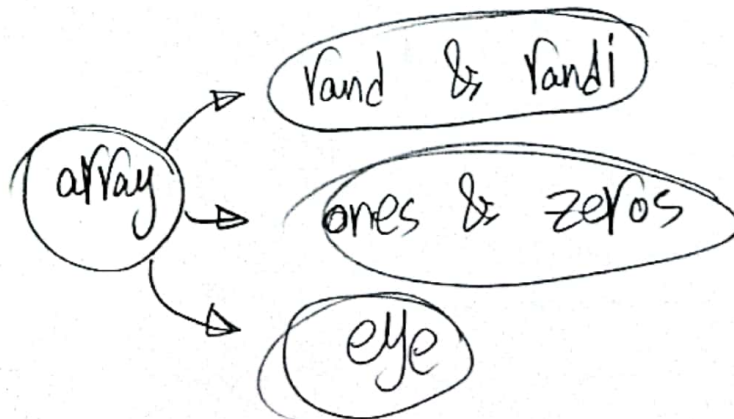
• Creating column vectors:



• Creating matrix:



• built-in functions to create special arrays:



- getting double arrays metadata:

The Function

size, length

min, max
sum, prod

⇒ operates along the first array dimension whose size does not equal one.

sign

— for vectors, use:

length
min, max
sum, prod
sign

— for matrices, use:

size
min, max
sum, prod
sign

Notes:

- end refers to → the last element in a vector.
↳ the last row or column in a matrix.
- (:) is shorthand notation for 1:end

Logical arrays

• Creating logical arrays:

method (1)

using Relational ($>, >=, <, <=, ==, \neq$)

or Logical ($\&, \mid, \sim$) operators

method (2)

built-in functions to create special arrays:

— true & false

— logical ($[1\ 0\ 1\ 1\ 1]$)

double array of zeros & ones

• getting logical arrays metadata:

The functions

any, all \Rightarrow operate along the first array dimension whose size does not equal one.

find \Rightarrow uses linear indexing

+ size, length, min, max, sum, prod, sign

- any(logical array): returns true if any element is true.
- all(logical array): returns true if all element is true.
- find(logical array): returns the indices of the true elements.

Important note:

- numerical operations can be done on logical vectors (+, -, *, /, ... so on)

Types of array indexing

indexing with element positions

indexing with a single index

indexing with logical values

- For vector (row or column) indexing, the form is

$V(\square)$

- So indexing with element positions and indexing with a single index are the same.

Ex. $V(3)$ or $V([1 \ 5 \ 6])$ or $V(1:5)$

- Indexing with logical values

Ex. $V(\text{logical}([1 \ 0 \ 1 \ 0 \ 1 \ 1]))$ ↳ logical vector

- For matrix indexing, the form is $M(\square, \square)$

- Indexing with element positions

Ex. $M(2,3)$ or $M([1 \ 5], [1 \ 2])$ or $M(1:3, 2:\text{end})$

- Indexing with a single index

Ex. $M(3)$ or $M([4 \ 6])$

- Indexing with logical values

- Special Case:

using $A(:)$ will put all the elements of the array in a column vector.

notes:

- in matlab, the indexing starts from 1.
- In a single indexing, the matrix is indexed as the following:

1	4	7
2	5	8
3	6	9

In words, column-wise indexing.

- the expression $A(A > 5)$ is equivalent to $A(\text{find}(A > 5))$

Accessing elements of a matrix

we use $()$ for indexing while we use $[]$ for the array itself.

the general form

$\text{mymat}(\square, \square)$

this could be a scalar or matrix

so that each value in the first square \square matches all the values in the second square \square .

example

$\text{mymat}([2, 4, 6], [1, 2, 3])$

$= \text{mymat}(\quad)$

those indices

	1	2	3
2	(2,1)	(2,2)	(2,3)
4	(4,1)	(4,2)	(4,3)
6	(6,1)	(6,2)	(6,3)

not only (2,1), (4,2), (6,3) as I was expecting before.

Important note (again):

— numerical operations can be done on Logical Vectors (+, -, *, /, --- so on).

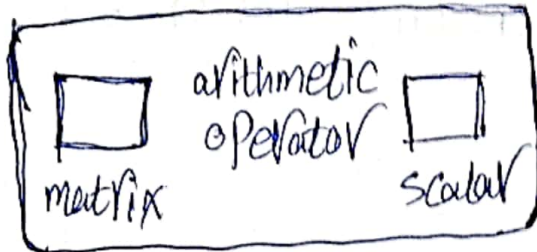
• operations on the arrays (double or Logical arrays):

— Removing elements: $A(\text{index}) = []$

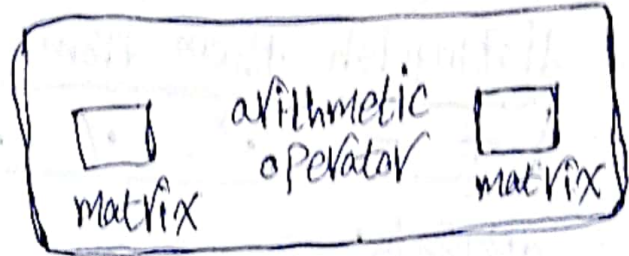
— transpose: use A' not $\text{transpose}(A)$

Operations on Vectors & matrices

Working with matrices makes you encounter two forms for arithmetic operations on matrices & vectors:



or



Types of arithmetic operations in matlab:

in order

1) Matrix operations

operations that follow the rules of linear algebra.

special case:

in the case of , the scalar will be

treated as a matrix of the same size as the other one with all elements equal the scalar.

Arithmetic operators:

+ - * / ^

operands:

the operand could be a matrix or scalar as the two forms show above.

2) array operations

operations that execute element by element.

Arithmetic operators:

We add (\cdot) before the array operators to distinguish them from the matrix operators.

$+$	$-$	$\cdot *$	$\cdot /$	$\cdot ^$
-----	-----	-----------	-----------	-----------

operands:

the operand could be a matrix or scalar as the two forms show above.

$\begin{bmatrix} \square \\ \square \end{bmatrix} \pm \begin{bmatrix} \square \\ \square \end{bmatrix}$

$\begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix} \pm \begin{matrix} \square & \square & \square \\ \square & \square & \square \end{matrix}$