

Topics

• Mathematical operations
• Manipulation
• Sorting and searching

4

# Arithmetic operations

- 2 different types of arithmetic operations:
  - array operations
  - matrix operations.
- Arithmetic operations
  - adding two numbers,
  - · raising the elements of an array to a given power,
  - multiplying two matrices
  - Etc.
- · Matrix operations follow the rules of linear algebra.
- Array operations execute element by element operations and support multidimensional arrays.
  - The period character (.) distinguishes the array operations from the matrix operations.

KU LEUVEN

operator	http://www.mathworks.nl/help/techdoc/ ref/arithmeticoperators.html#f75-87292		function	
	Binary addition	A+B	plus(A,B)	
	Unary plus	+A	uplus(A)	
	Binary subtraction	A-B	minus(A,B)	1
	Unary minus	-A	uminus(A)	
	Matrix multiplication	A*B	mtimes(A,B)	
	Arraywise multiplication	A.*B	times(A,B)	
	Matrix right division Divide by	A/B	mrdivide(A,B)	
	Arraywise right division	A./B	rdivide(A,B)	
	Matrix left division Divide into	A\B	mldivide(A,B)	
	Arraywise left division	A.\B	Idivide(A,B)	
	Matrix power	A^B	mpower(A,B)	
	Arraywise power	A.^B	power(A,B)	
	Complex transpose	A'	ctranspose(A)	]
	Matrix transpose	A.'	transpose(A)	

### Array operations

- Array operations
  - execute element by element operations on corresponding elements of vectors, matrices, and multidimensional arrays.
  - If the operands have the same size, then each element in the first operand gets matched up with the element in the same location in the second operand.
  - If the operands have compatible sizes, then each input is implicitly expanded as needed to match the size of the other.

KU LEUVEN

6 7

#### 

### Scalar-Array arithmetic

- Addition, subtraction, multiplication, and division of an array by a scalar applies the operation to all elements of the array.
- Implies scalar expansion for addition and subtraction to have the mathematics correct

		4; 5 6	,		
A =					
1	2	3	4		
5	6	7	8		
>> A-2					
ans =					
-1	0	1	2		
3	4	5	6		
>> 2*A-1					
ans =					
1	3	5	7		
9	11	13	15		
>> 3*A/5	+4				
ans =					
4.60	00	5.2000	5.8000	6.4000	
7.00	00	7.6000	8.2000	8.8000	

### Element-by-element operations: dot (.) operator

- Arithmetic operations on arrays are just like the same operations for scalars but they are carried out on an element-by-element basis.
- The dot(.) before the operator indicates an array operator; it is needed only if the meaning cannot be automatically inferred.
- applies to vectors, matrices, multidimensional arrays

9

Operation	Meaning
C = a./A	$C_{ij} = a/A_{ij}$
C = A.\a	$C_{ij} = a/A_{ij}$
C = A.^a	$C_{ij} = A^a_{ij}$
C = a.^A	$C_{ij} = a^{A_{ij}}$
C = A.*B	$C_{ij} = A_{ij}B_{ij}$
C = A./B	$C_{ij} = A_{ij}/B_{ij}$
C = A.\B	$C_{ij} = B_{ij}/A_{ij}$
C = A.^B	$C_{ij} = A_{ij}^{B_{ij}}$

KU LEUVEN

#### dot (.) operator

- The dot operator, used with multiplication, division, and exponentiation, creates element-wise operations.
- The one exception to that is the use of the dot operator in creating matrix transposes. The 'regular' matrix transpose (') creates the complex-conjugate transpose of a complex vector or matrix. Using the (.') creates the transpose without doing the complex-conjugate operation.

KU LEUVEN

10 11

# Implicit expansion

- MATLAB R2016b, contains a feature called implicit expansion, which is an
  extension of the scalar expansion.
- MATLAB now treats "matrix plus vector" as a legal operation. This is a controversial change, as it means that MATLAB now allows computations that are undefined in linear alpebra

```
are undefined in linear algebra.

A = ones (2) , B = A + (1 5)

A = 2 * 2

1 1 1

B = 2 * 2

2 6

A = ones (2) + [1 5] *

A = 2 * 2

6 6
```

# More on Array Operations

 Most MATLAB functions will work equally well on both scalars and arrays (of any dimension)

```
>> A=[1 2 3 4 5];

>> sin(A)

ans =

    0.8415    0.9093

    0.1411    -0.7568

    -0.9589

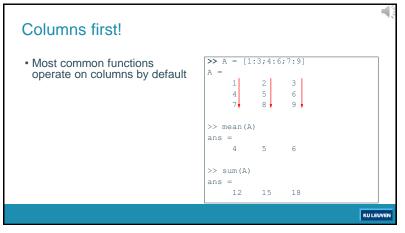
>> sqrt(A)

ans =

    1.0000    1.4142    1.7321

    2.0000    2.2361
```

KU LEUVEN



Built-in functions

help elmat: Matrix manipulation.

• fliplr: Flip matrix in left/right direction.

• flipud: Flip matrix in up/down direction.

• rot90: Rotate matrix 90 degrees.

rot90(a,n): Rotate n-times

• circshift (A, shiftsize) circularly shifts the values in the array, A, by shiftsize elements.

• shiftsize is a vector of integer scalars where the n-th element specifies the shift amount for the n-th dimension of array A.

• positive shiftsize: shift down (or to the right).

• negative shiftsize: shift up (or to the left).

15 16

