Matlab: What is it?

- Computing & programming environment
 - High-level, interpreted language
 - Rapid prototyping & integrated debugging
 - Built-in graphics & visualization tools
 - Built-in toolboxes for many applications
- Compared to other common languages
 - C/C++ is a low-level, compiled language
 - Better performance
 - Harder to write, difficult data visualization, harder to employ toolboxes, longer codes, less portable
 - Python
 - Somewhere in between

Matlab: Why learn it?

- Computation and data analysis are two fundamental components of virtually any discipline.
 - Math
 - Science
 - Engineering
 - Humanities
 - Social Sciences
- A little bit of time upfront to write a (good) Matlab code can yield a long-term tool to iteratively test & debug algorithms, explore & process data, automate calculations & analysis, visualize results, etc.

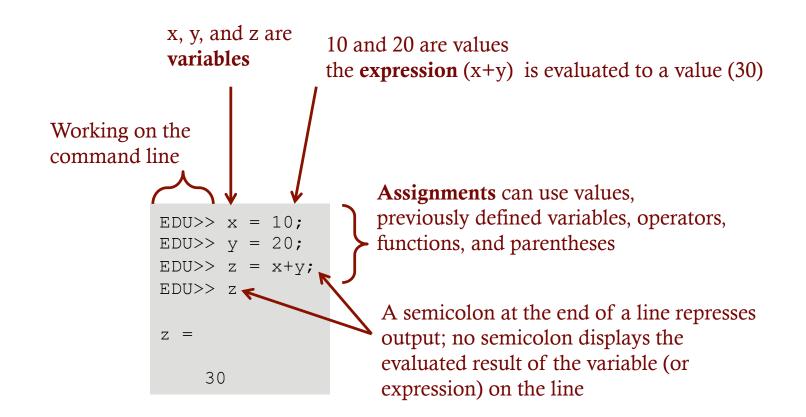
Lecture 1 Topics

- Basic concepts
 - Variables, expressions, & assignments
 - Operators
 - Built-in functions
 - Good programming practices
 - Data types
- Vectors and matrices
- Strings and cell arrays
- Getting help

Basic concepts

A variable is any symbolic representation for a "value" An assignment gives the variable its value.

An expression is a mathematical statement which evaluates to a value



Basic concepts

• Operators: Addition: + Multiplication: * Exponentiation: ^

Subtraction: – Division: /

- Order of operations (including parentheses) as usual
- Common built in functions:
 - Trig: sin(), cos(), tan(), sind(), cosd(), atan()
 - Exponentials: exp(), log()
 - Complex: abs(), conj(), imag(), real()
 - Rounding: round(), floor(), ceil(), mod()

Basic concepts

- Variable names
 - Must begin with a letter, but can contain letters, numbers, and underscores
 - Apart from the underscore, no other special characters (including spaces) are allowed.
 - Case sensitive
 - Length limited (use the function namelengthmax in the command window to determine this value)
 - Cannot/should not use reserved words (such as names of built-in functions, such as sin(), or predefined values, such as pi)

Good programming practices

Expressions & Assignments

If an assignment or expression is very long (i.e. longer than the command/editor window), it can be broken between several lines using an ellipsis (three periods in succession).

➤ Variables

Use meaningful/conventional variable names:

```
force = jedi*midichlorian
```

is going to be harder to understand when you look back on your physics homework code than

```
force = mass*accel
```

> Use relatively short names:

root_mean_squared_velocity
while descriptive, is going to be a pain to type; a simpler choice might be
vel_rms

Don't make variable names too similar:

rcubed and rCubed is asking for a debugging nightmare

Be consistent in case & capitalization style

- > underscore_case
- ➤ camelCase

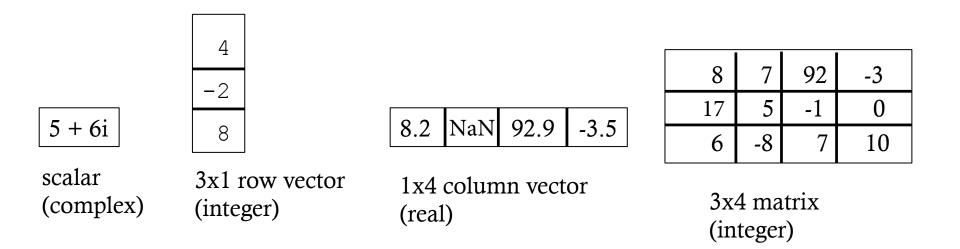
Basic concepts

- Data types (values that can be assigned to variables)
 - Numbers
 - Integers entered without a decimal point
 - Real use realmin and realmax to return the smallest and largest real numbers represented in Matlab
 - Complex represented in rectangular form; imaginary unit $\sqrt{-1}$ defined as i and j in Matlab

EDU>> i ans = 0 + 1.0000i

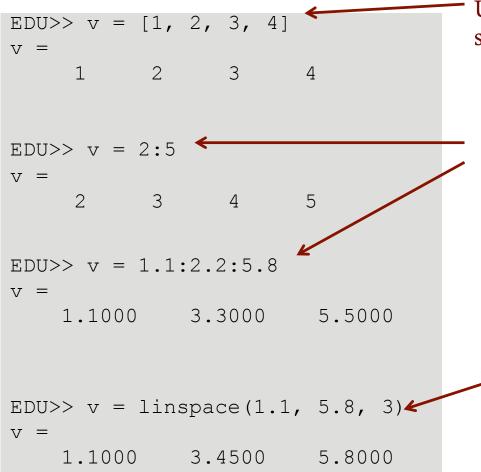
- "Non-numbers": -Inf, Inf, NaN
- Vectors/matrices
- Strings
- Cell arrays
- Structures, function handles, plot handles, ...

Vectors and Matrices



• Store values of the same type only

Creating Row Vectors

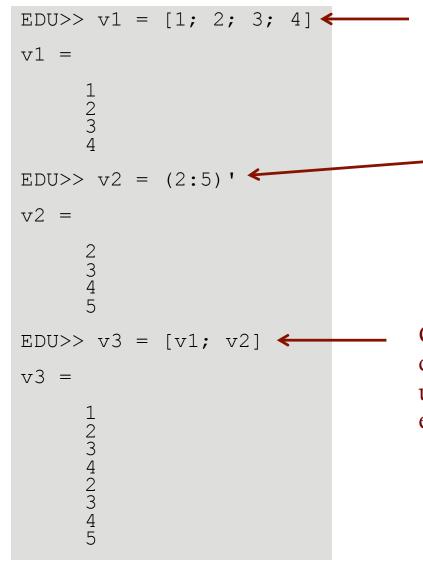


Use brackets with comma or simply space separated elements

Using the colon operator you can choose a beginning value, end value, and increment size (all of which must be integers or real numbers, not complex)

linspace(start, end, n),
where n is the number of equal
increments in which to divide the
range start to end

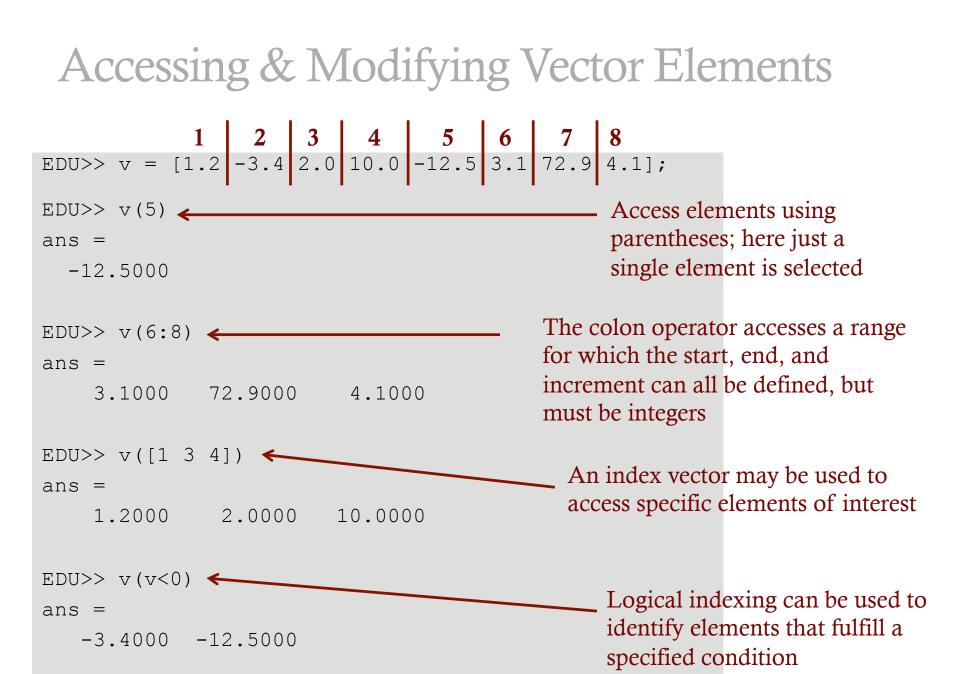
Creating Column Vectors



Use brackets with semicolon separated elements

Transpose operator acting on row vector (because operations within parentheses evaluated first, could use colon operator or linspace to form initial row vector)

Concatenating two column vectors. This can also be done with two row vectors using a comma or space separated elements within the brackets.



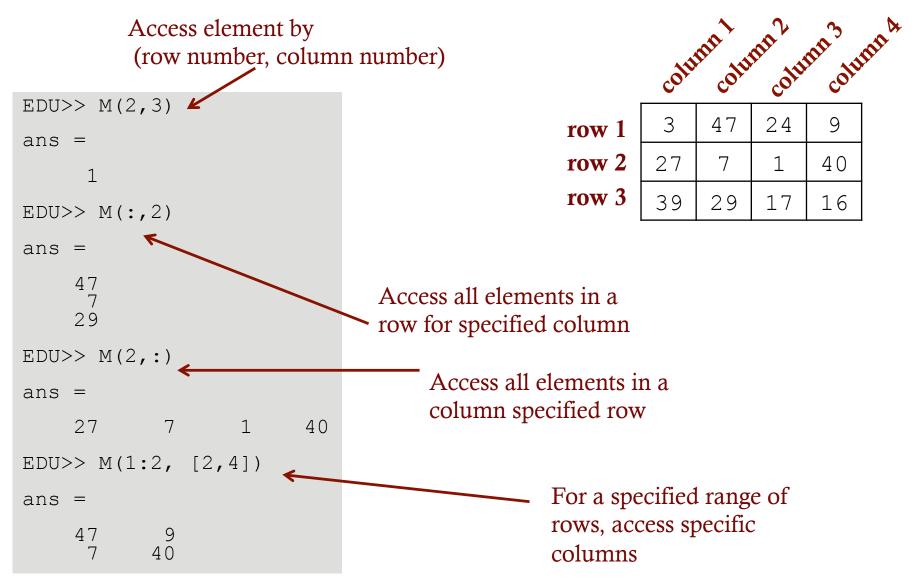
Creating	. Commas or spaces between elements in rows, semicolons			
EDU>> M = [8 7 9 NaN 5 6 -8 M =	92 -3; -1 0; 7 10]		between rows. Note the u of an ellipsis to break up assignment expression	lote the use break up the
8 7 NaN 5 6 -8			Colon operators linspace can also create the rows o	o be used to or columns.
EDU>> M = [1:4; M = 1 2 9 8	9:-1:6] 3 4 7 6		Just make sure to proper separator row vs column d	between
5.5000 3.	.0000 -1. .0000 1.	3))',(2:4)', 0000 0000 0000	(-1:2:4) /]	

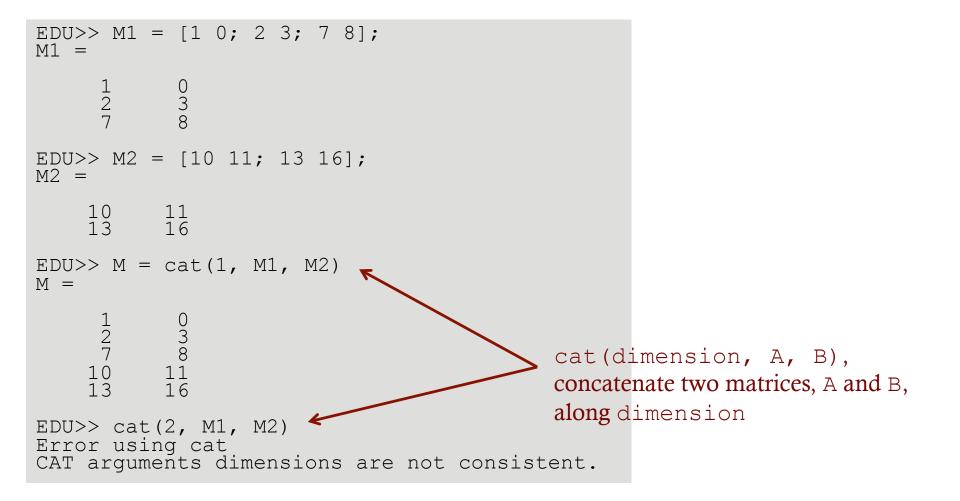
Creating Matrices

EDU>> M = [1 2 3; 4 5]; Error using vertcat CAT arguments dimensions are not consistent. Must have same number of elements in each row and same number of elements in each column

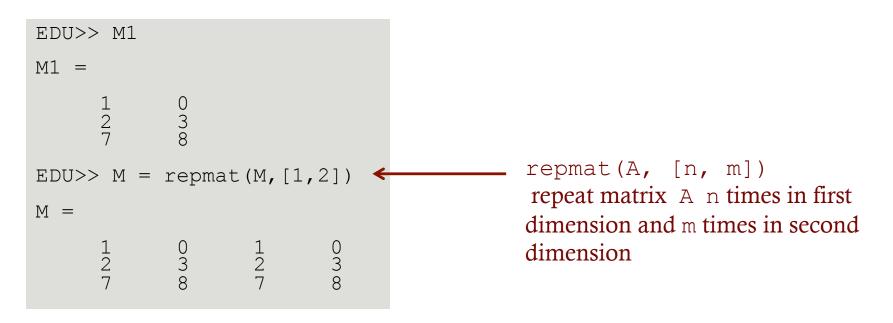
• Matlab also supplies a number of special matrix definitions, such as zeros(), ones(), rand(), randi(), diag(), eye(), ...

Accessing & Modifying Matrix Elements





Accessing & Modifying Matrix Elements



22

Vector and Matrix Operations

- Matrix vs element-wise operations:
 - Matrix multiplication:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} X \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

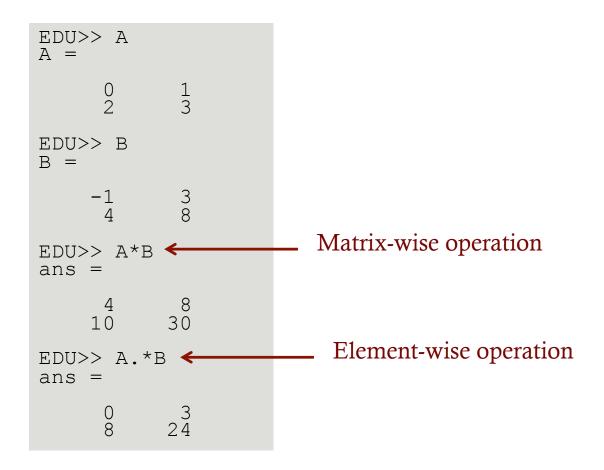
$$A \qquad B \qquad C$$

• Element-wise multiplication:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} X \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae & bf \\ cg & dh \end{bmatrix}$$

A B C

Vector and Matrix Operations



Operation	Matrix-wise	Element-wise	
Addition	+	+	_
Subtraction	-	-	
Multiplication	*	.*	
Left division	Λ	.\	Left division: $a \setminus b \to \frac{b}{a}$
Right division	/	./ J _R	ight division: $a/b \to \frac{a}{b}$
Exponentiation	^	.^	0

Vector and Matrix Operations

• Most functions which act on a scalar can be given a vector/ matrix argument and will act element-wise:

EDU> EDU>	·> ·>	A = sqrt	[4 (A)	16;	0	2];	
ans	=						
	2.	0000			000 414		

• Functions of a matrix in the linear algebra sense are signified by names ending in m: expm, funm, logm, sqrtm

EDU>> B = sqrtm(A) B = 2.0000 4.6863 0 1.4142 EDU>> B*B ans = 4.0000 16.0000 0 2.0000

Vector and Matrix Operations

• Vector dot product and cross product

• Matrix transpose

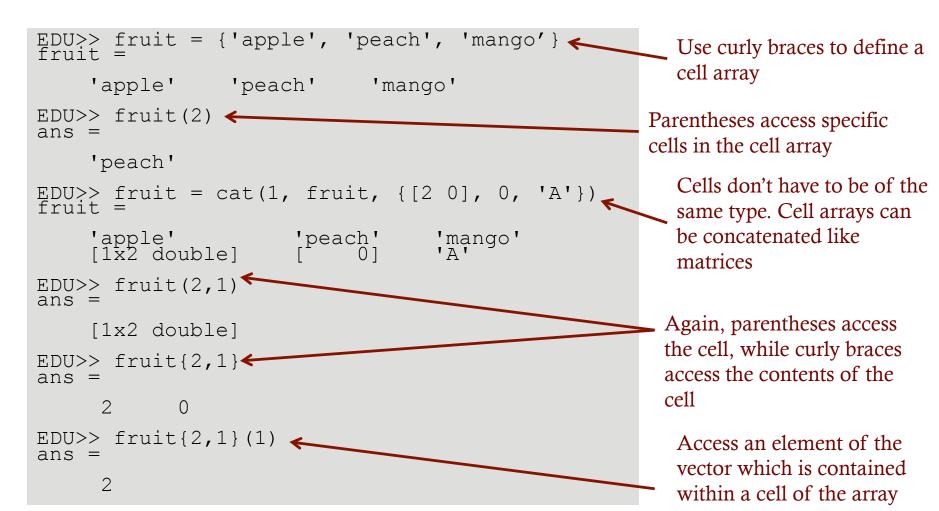
```
EDU >> A = [1 \ 2 \ 3; \ 4 \ 5 \ 6];
EDU>> A
A =
        2 3
5 6
      1
      4
EDU>> A'
ans =
      123
              4
5
EDU>> transpose(A)
ans =
              456
      1
2
3
```

• Strings are vectors of characters.

```
EDU>> fruit = 'apple' <
fruit =
                                               Single quotation marks
apple
EDU>> fruit(1)
ans =
а
EDU>> fruit = ['apple'; 'mango'; 'peach'];
EDU>> fruit
fruit =
apple
mango
peach
EDU>> fruit(2,1)
ans =
m
EDU>> fruit = ['apple'; 'mango'; 'kiwi'];
Error using vertcat
CAT arguments dimensions are not consistent.
```

Cell arrays

• Cell arrays generalize matrices to allow for arbitrary entries. Each element is called a "cell" and access of the cells is by numerical indexing



Getting help

- Information about Matlab can be found in Matlab's help facilities. Command line arguments include:
 - helpbrowser: opens a new window which contains different ways to obtain the correct information, like lists, a global index and a search function.
 - help <function name> gives a short description of the function, the syntax, and other closely related help functions. If more extensive results are needed, try the command doc <function name> which also opens the help browser window.
 - lookfor 'topic' gives the list of all possible function names which contain the specific search word.
- Online: <u>http://www.mathworks.com/help/matlab/</u> http://www.mathworks.com/matlabcentral/?s_tid=gn_mlc