Signals & Systems Lab

EECS 360

Fall 2014

Syllabus

What is MATLAB?

MATrix LABoratory

MathWorks, 1984

GETTING STARTED:

1. Log into computer
2. Under Start Menu find MATLAB R2013b
3. Change directory to student drive

Most Important thing you need to know about MATLAB is:

help

If you know this you can do basically anything!

Example

Type the following in the command window

```
>> help elfun
```

Note elfun stands for elementary functions. Click on one of the results and see the wealth of information available without resorting to the internet.

Simple Mathematical Operations

+	addition
-	negation, subtraction
*	multiplication
/	divided by division (e.g. 10/5 is 2)
\	divided into division (e.g. 5\10 is 2)
^	exponentiation (e.g. 5^2 is 25)

Integers!?

```
>> a = 1000

a =

        1000

>> a = a * a

a =

     1000000

>> a = a * a

a =

   1.0000e+12
```

Your turn! Try these:

```
>> 75/5+3
>> 5\25
>> 2.71828182846^2
>>5--3
>>(4^2)-1
>>4^(2-1)
>>exp(2)
```

Order of operation precedence

()	parentheses
^	exponentiation
-	negation
*,/,\	all multiplication and division
+,-	addition and subtraction

Other helpful operators and constants

=	Assignment operator
;	Suppression operator
pi	3.14159…
i	$\sqrt{-1}$
j	$\sqrt{-1}$
inf	∞
NaN	not a number such as 0/0

Space Saving Hint

IN COMMAND WINDOW TYPE:

format compact

This will prevent the extra whitespace between lines in the command window.

Other things that can be done with format

Lets use our friend help for this one

>>help format

All about integers

Here are the eight integer classes, the range of values you can store with each type, and the MATLAB conversion function required to create that type:

Class	Range of Values	Conversion Function
Signed 8-bit integer	-2^7 to 2^7-1	`int8`
Signed 16-bit integer	-2^{15} to $2^{15}-1$	`int16`
Signed 32-bit integer	-2^{31} to $2^{31}-1$	`int32`
Signed 64-bit integer	-2^{63} to $2^{63}-1$	`int64`
Unsigned 8-bit integer	0 to 2^8-1	`uint8`
Unsigned 16-bit integer	0 to $2^{16}-1$	`uint16`
Unsigned 32-bit integer	0 to $2^{32}-1$	`uint32`
Unsigned 64-bit integer	0 to $2^{64}-1$	`uint64`

Source: http://www.mathworks.com/help/matlab/matlab_prog/integers.html

Calculate, calculate, calculate

Ranges of unsigned and signed integers

- N bits unsigned:
- N bits signed:

```
>> intmax('int8')

ans =

  127

>> intmax('int64')

ans =

  9223372036854775807
```

```
>> intmin('int8')

ans =

 -128

>> intmin('int64')

ans =

 -9223372036854775808
```

The Matlab functions **intmin()** and **intmax()**

QUIZ: **intmin() intmax()**

What do the following Matlab functions return?

intmin('uint8')

intmax('uint8')

intmax('int16')

intmin('int16')

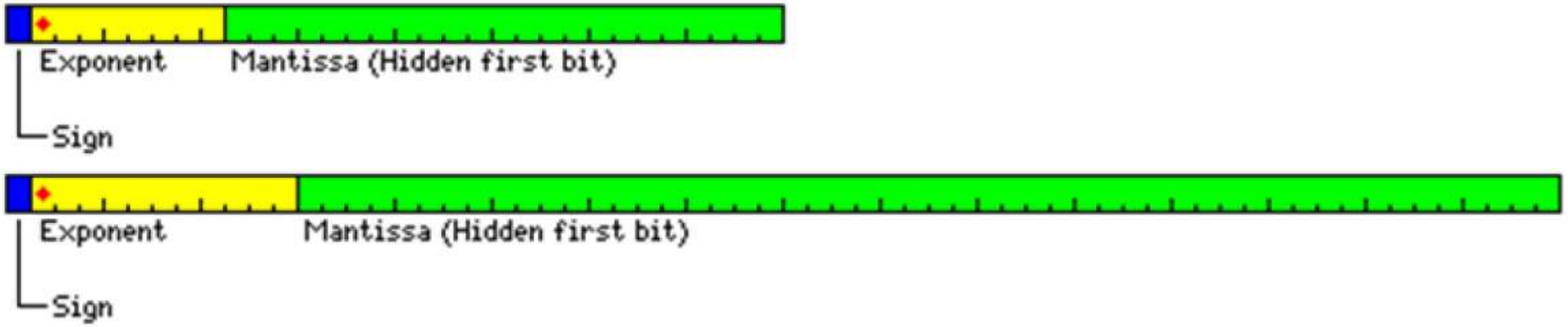
intmax('int32')

intmax('uint32')

Data Types and Operations

- Most of the time we use double() which stands for Double-Precision Floating Point Decimal (IEEE 754 standard). This is default for MATLAB numeric operations
- Logical ones and zeros along with the Boolean operations
- Characters based on the ASCII standard

IEEE 754 – Single and double precision



Source: http://www.quadibloc.com/comp/cp0201.htm

Double and Single data type

Note how much larger the range of representation over int64()!

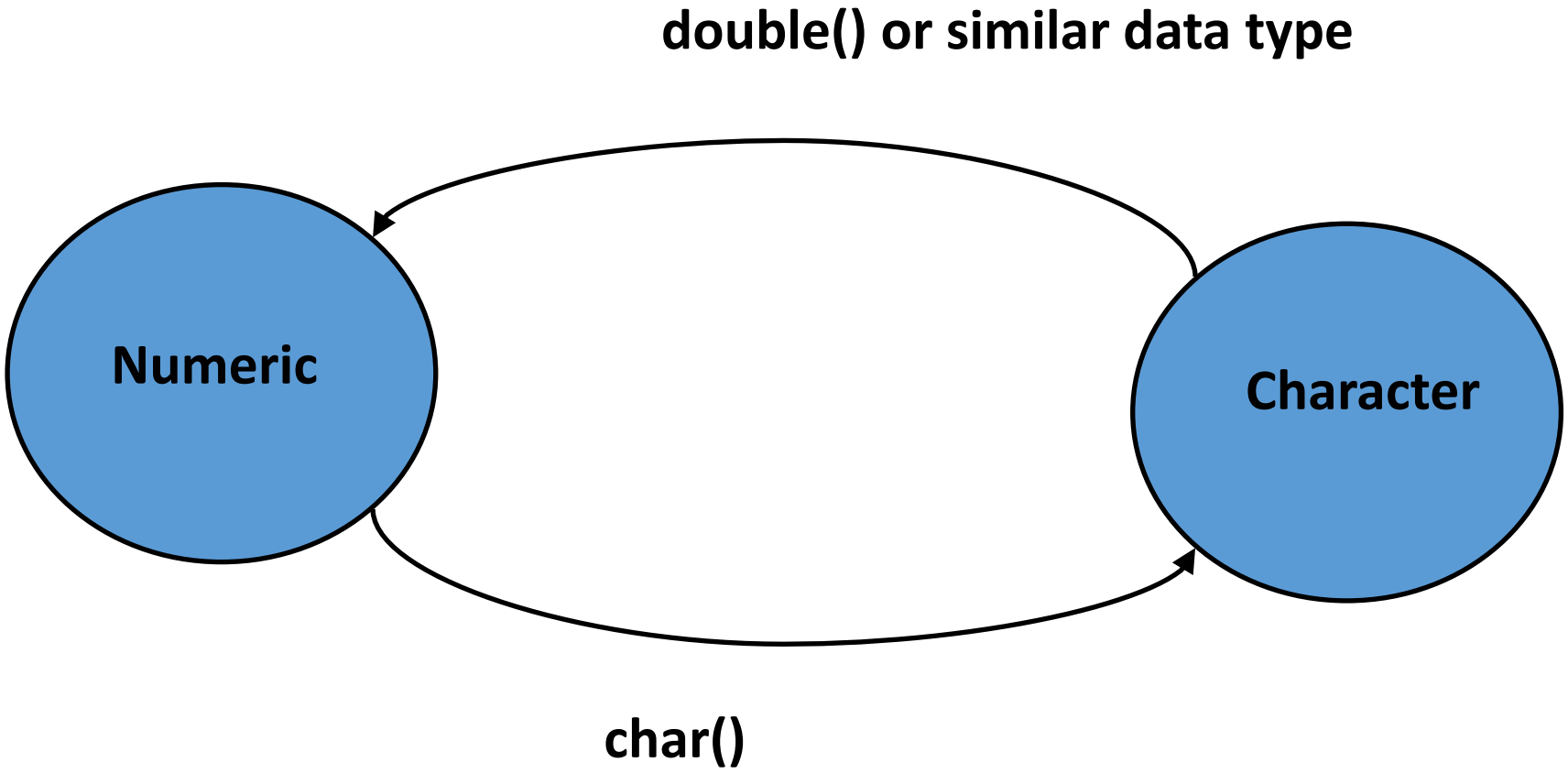
```
>> realmin('single')
ans =
   1.1755e-38
>> realmax('single')
ans =
   3.4028e+38
```

```
>> realmax('double')
ans =
   1.7977e+308
>> realmin('double')
ans =
   2.2251e-308
```

Class	Max value	Min Value	Bytes	Smallest difference
logical	1	0	1 (yes, Matlab wastes 7 bits here)	1
int8	127	-128	1	1
int16	32767	-32768	2	1
int32	2.14e+09	-2.14e+09	4	1
int64	9.22e+18	-9.22e+18	8	1
uint8	255	0	1	1
uint16	65535	0	2	1
uint32	4.29e+09	0	4	1
uint64	1.84e+19	0	8	1
single	3.40e+038	-3.40e+038	4	1.1755e-38
double	1.79e+308	-1.79e+308	8	2.2251e-308

Double is a memory-hog! We use it because it is safe, not always efficient.

How to go between characters and numerics



The inputs to char() are typically found on an ASCII table which can easily be found online for reference.

Arrays in MATLAB

1-D Arrays in MATLAB

Vectors

Name of the game: MATLAB

- v = [1 2 3 4 5]
- v = [1, 2, 3, 4, 5]
- v = 1:100

- v = 1:1:100
- v = 1:2:100

Using the iteration operator (colon) allows you to control the step size between adjacent elements in the vector.

Note step value defaults to 1 if not specified.

Your turn!

- v = 1:2:6
- v = 1:15:30
- v = 100:-15:3
- v = 1:-15:30

Referencing vector elements

```
>>r=[1 67 81 165 0];

>>r(4)=??
```

How would you create these vectors?

- 2 9 16 23 30
- 9 7 5 3 1

```
>> v = linspace(3, 15, 5)
v =
     3     6     9    12    15
```

- Write a command to create a vector of points spaced at 0.01 intervals between 0 and 50

```
>> length(v)
```

Another way to create vectors: linspace

Why might this method be preferred sometimes?

In MATLAB indices start at **1**

```
>> v = linspace(3, 15, 5)
v =
     3     6     9    12    15

>> v(3)
ans
     ????
```

```
>> v = linspace(3, 15, 5)
v =
     3     6     9    12    15

>> v(2:4)
ans
```

????

```
>> v = linspace(3, 15, 5)
v =
     3     6     9    12    15

>> v([1 2 6])
ans
????
```

In this context the vector [1 2 6] is called the index vector because it indexes elements of v.

Explain what happens here. How to fix?

The old to the new

The Column Vector – New Operator

‘ transpose operator

Example:

```
>>v=[1 2 3 4 5];
>>v’
```

What is the output?

The Column Vector

Direct Approach - More Keystrokes ☹

; column operator

Example:

>> t=[0; 0.1; 0.01; 0.001]

Concatenation

```
>> nv=1:2:9
nv=
        1       3       5       7       9
>> ls=linspace(3,15,5)
ls=
        3       6       9       12      15
>>newvec=[nv ls]
newvec=
        ???
```

Reassignment

```
>> nv=1:2:9
nv=
     1     3     5     7     9
>> nv(3)=0;
>> nv
nv=
     1     3     0     7     9
```

2-D Arrays in MATLAB

Matrices

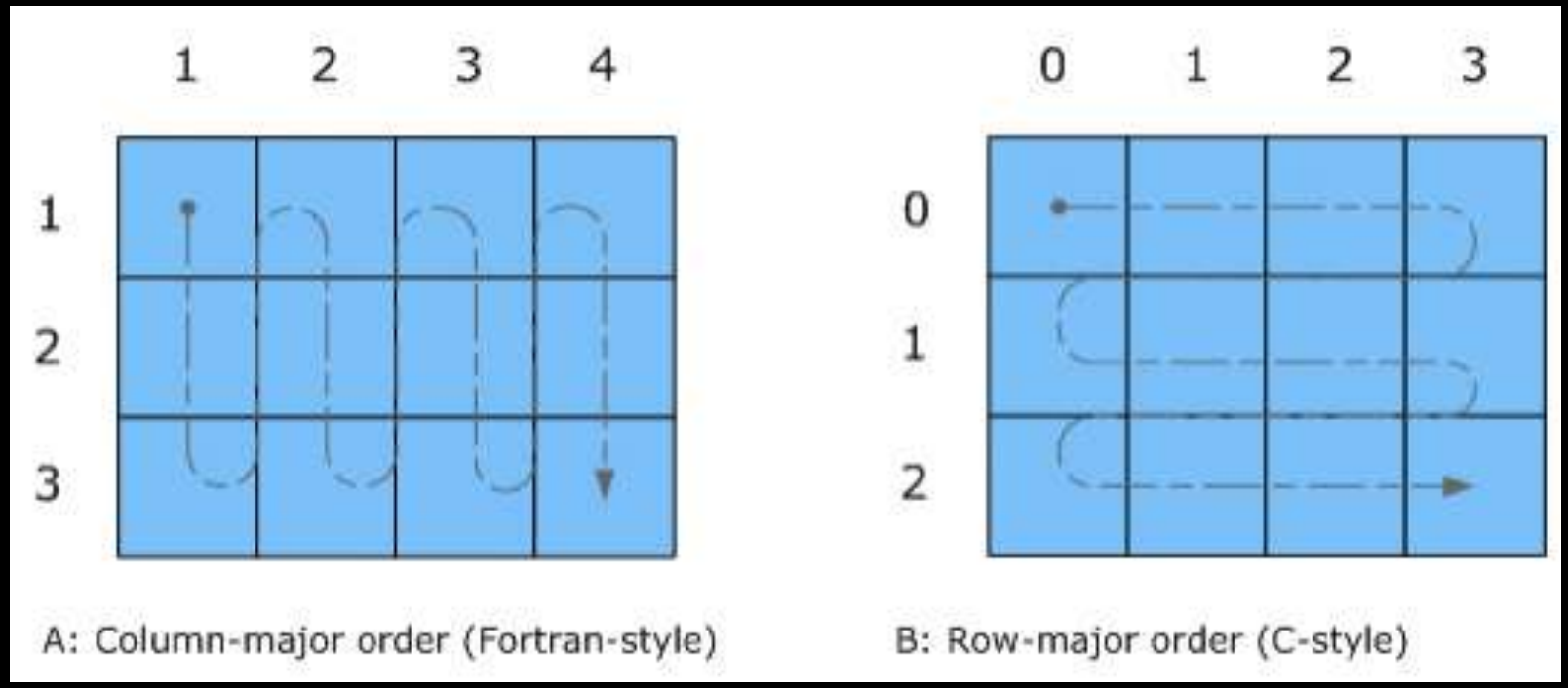
Special Matrices

ones(n,m)

zeros(n,m)

eye(n)

Column-major Order vs. Row-major Order



Note: MATLAB's code was originally written in FORTRAN .

Source:
https://software.intel.com/sites/products/documentation/hpc/mkl/lin/MKL_UG_coding_calls/Calling_LAPACK_BLAS_and_CBLAS_Routines_from_C_Language.htm

What if I don’t know how large the matrix is?

Example:

```
>>v=1:6;
>>v(end/2+1:end)
v=
        4        5        6
```

Note this works for m,n matrices also!

Homework

Handout, due at beginning of next class.

Sources

Agapie, M. (2013), *CS 344 Class Notes* [used with permission]

Attaway, S. (2012). *MATLAB a practical introduction to programming and problem solving* (2nd ed.). Waltham, MA: Butterworth-Heinemann.