ENGR 101 – Lecture 5

Logical Operations and Indexing

Laura Alford, James Juett, Rick Niciejewski 9/20/17

Lecture Goals

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

- **Previous lecture: Functions and Vectorization**
  - Reducing code duplication
  - Functions
  - Vectorization
  - Project 1 Overview
  - Suggested readings, Attaway, Chap 3.7 and 6.1-6.2

- **Today's lecture: Logic Operations and Logical Indexing**
  - Relational operators (gt, lt, not eq, ...)
  - Logical operators (and, or, not)
  - Final MATLAB precedence table
  - Indexing matrices with logical arrays
  - Suggested readings, Attaway, Chap 1.6 and 2.5

Recall: Binary Arithmetic Operations

- Essentially, doing math with two operands.

	Operator	Function	Example	Result
Addition	`+`	`plus`	`2 + 3`	5
Subtraction	`-`	`minus`	`5 - 3`	2
Multiplication	`.*`	`times`	`5 .* 3`	15
Exponentiation	`.^`	`power`	`2 .^ 3`	8
Division	`./`	`rdivide`	`11 ./ 4`	2.75
Modulo		`mod`	`mod(11,4)`	3

Asking Questions

- Consider the following expression:

```
x < 5
```

- ~~In math, this would mean that x must be less than 5.~~
- In programming, this is a **question**.
  - "Check the current value of x… Is it less than 5?"
  - The result is called a **truth value**, and is either false or true.
  - False and true are encoded as `0` and `1`, eg. disp(3 < 5) = 1

false	true
`0`	`1`

Relational Operations

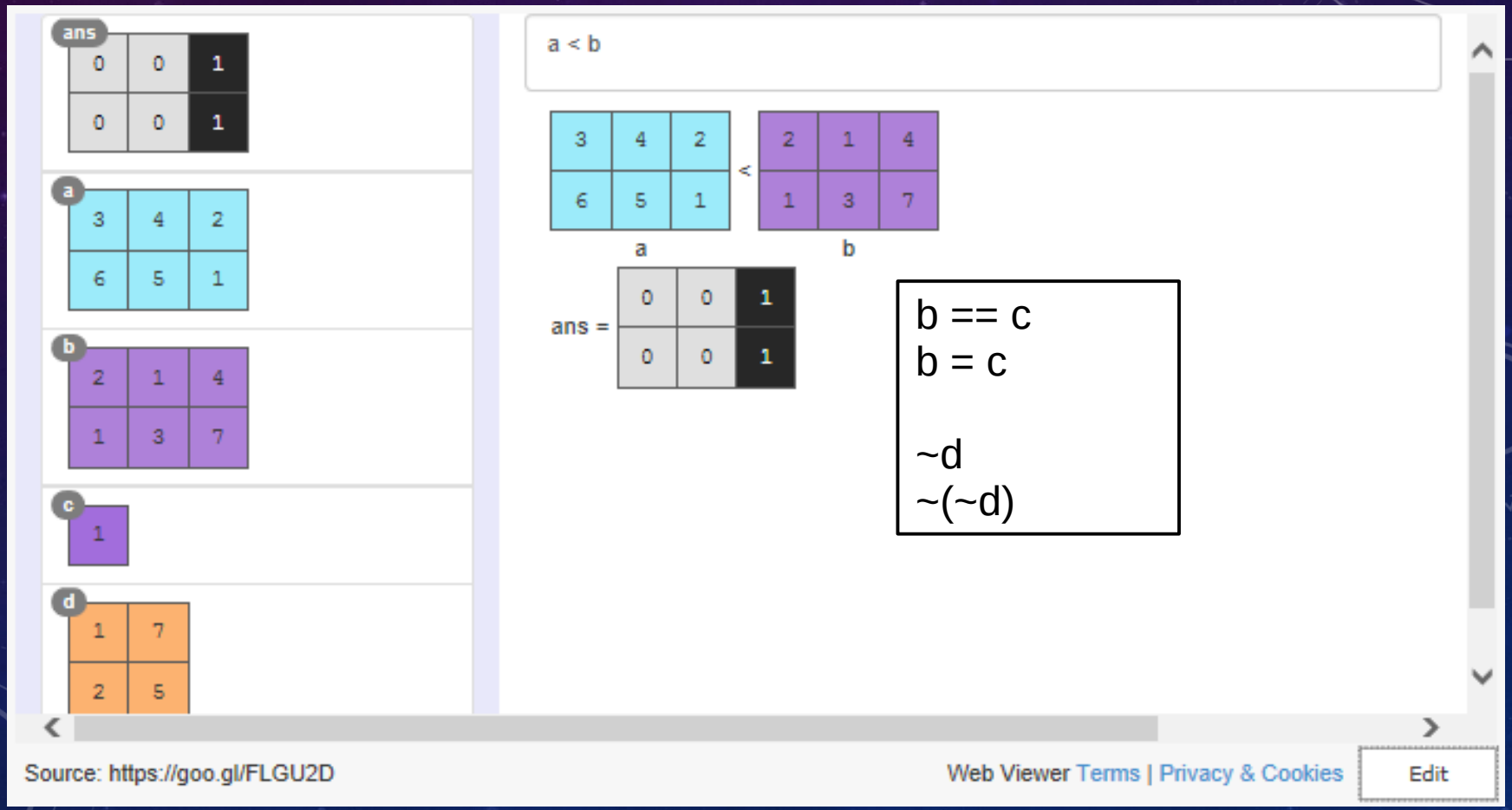
- These operations check for **equality** or perform **comparisons**.

	Operator	Function	Example	Result
Equality	==	eq	2 == 3	0
Inequality	~=	ne	2 ~= 3	1
Less Than	<	lt	5 < 5	0
Less Than or Equal	<=	le	5 <= 5	1
Greater Than	>	gt	gt(3, 4)	0
Greater Than or Equal	>=	ge	ge(4, 4)	1

goo.gl/FLGU2D

Relational Operations on Matrices

In MATLAB, anything that you can do with scalars, you can do with arrays (i.e. vectors and matrices).



goo.gl/syLgsb

Logical Operations

- We can use **logical operators** to combine two truth values according to the rules of **formal logic**.
- For example, the & operator implements **logical and**.
  - If BOTH operands are true, the whole result is true. Otherwise false.

https://goo.gl/syLgsb

```
(5<6)&(4==3)
(5<6)&(4==4)
```

Inclusive vs. Exclusive Or

- There are two kinds of "or"…

- Inclusive Or
  - One or the other, or both.
  - "Were you in sports or music in high school?"

or

- Exclusive Or
  - One or the other, but NOT both.
  - "Would you like the soup or the salad?"

xor

Logical Operations

- Essentially, combining two **truth values** in a particular way.
- Most operations have both a symbolic operator and a function.

	Operator	Function	Example	Result
Logical And	`&`	`and`	`2 < 3 & 5 > 6`	`0`
Logical Or	`|`	`or`	`2 < 3 | 5 > 6`	`1`
Exclusive Or		`xor`	`xor(0,1)`	`1`
Not	`~`	`not`	`~(1 == 2)`	`1`

The precedence of ~ is higher than ==, so we need parentheses here.

MATLAB programming – precedence

- Order of precedence
  1. (highest) parentheses
  2. exponentiation
  3. negation
  4. multiplication, division
  5. addition, subtraction
  6. colon operator (:)
  7. Relational operators
  8. Logical AND
  9. Logical OR
  10. (lowest) assignment (=)

if two or more operations have the same precedence, the expression is executed from left to right

4 min

Your turn: Logical Operator Truth Table

- The formal way to define the logical operators is with a truth table that shows the result for all combinations of operands.
- Fill in the missing pieces of the table…

A	B	~A	~B	A & B	A | B	xor(A, B)
0	0			0	0	
0	1	1				
1	0					
1	1		0			0

Solution: Logical Operator Truth Table

- A formal way to define the logical operators is with a truth table that shows the result for all combinations of operands.
- Fill in the missing pieces of the table…

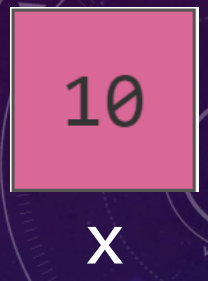
A	B	~A	~B	A & B	A | B	xor(A, B)
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Logical Operations with Matrices

- Of course, we can perform these same operations with matrices as well as scalars.

https://goo.gl/SgvMKg	a == b + 1 a(:,2) == b(:,3)

Careful!



- Let's say we wanted to check whether a number is in a particular interval. It's tempting to try something like this:

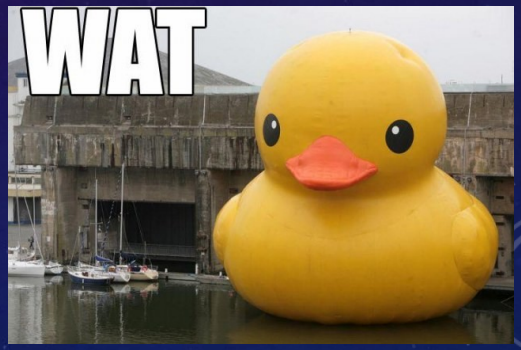
```
1 < x < 5
```

MATLAB claims for x=10 this is true.

- MATLAB sees this as:

```
(1 < x) < 5
(1 < 10) < 5
(1) < 5
1
```

`1 < 10` is true, so it evaluates to 1.



- Instead, use & to specify the lower and upper bounds separately:

```
1 < x & x < 5
```

Your turn: Logical Operations

5 min

goo.gl/5AQbNV

- Given these variables:

1	2	4
3	3	2

a

0	2	9
3	3	6

b

2

c

- Find the result of the following five expressions:

```
a ~= b          b - a == c + 3          a > c

a(2,:) == 3 .* [1,1,1]          a + 1 > 3 & b > 2
```

Solution: Logical Operations

- Given these variables:

1	2	4
3	3	2

a

0	2	9
3	3	6

b

2

c

- Find the result of the following expressions:

```
a ~= b
```

1	0	1
0	0	1

```
b - a == c + 3
```

0	0	1
0	0	0

```
a > c
```

0	0	1
1	1	0

```
a(2,:) == 3 .* [1,1,1]
```

1	1	0

```
a + 1 > 3 & b > 2
```

0	0	1
1	1	0

Logical Arrays

- The result of a relational or logical operation is a **logical array**.
- The 1s and 0s in a logical array are a completely different type of data than regular numbers.
- In many contexts, an expression will behave differently with logical vs. regular numbers.
  - Indexing is an important example.

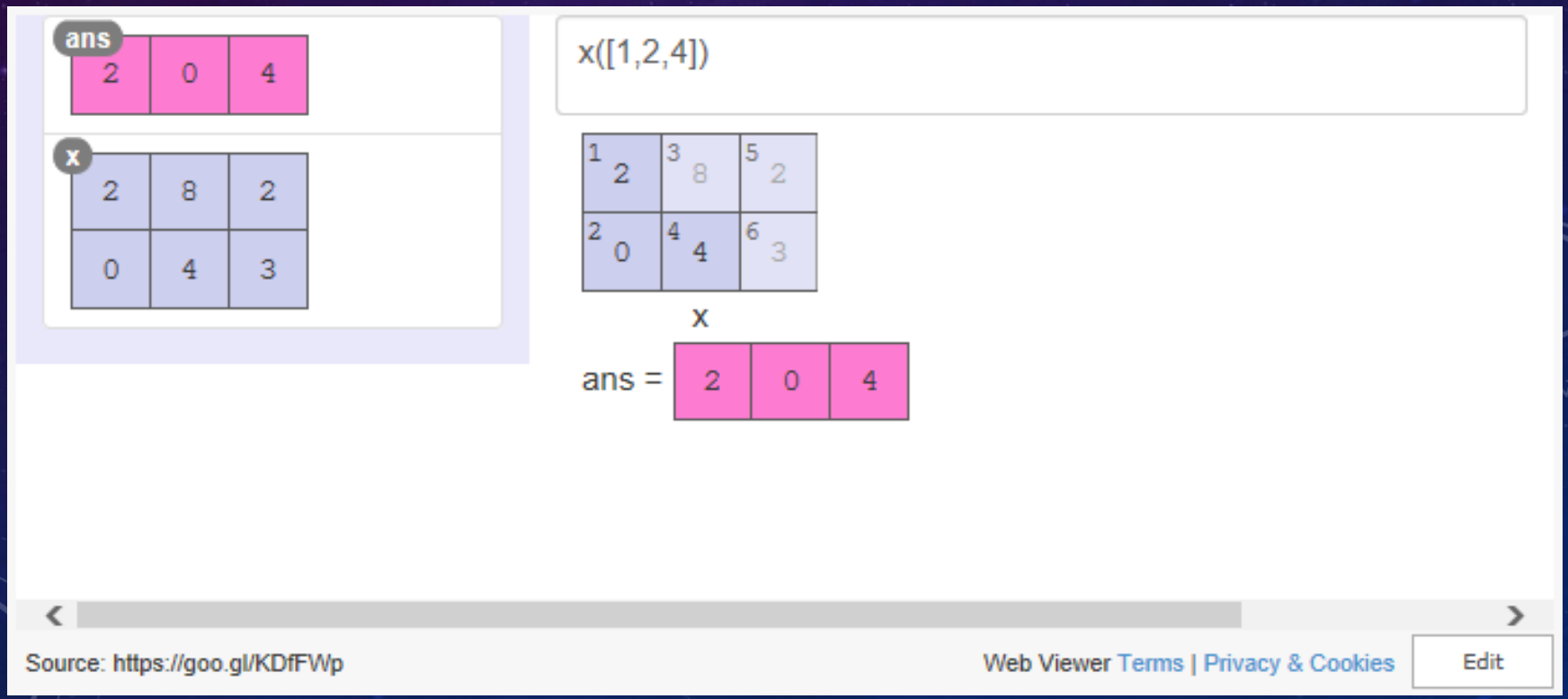
```
>> x = [1,2;3,4]

x =

     1     2
     3     4

>> which = x > 2

which =

   0   0
   1   1

>> whos which

  2×2 logical array
```

Recall: Matrix Indexing

- Each element has a sequential index.
- When we put a **[regular array]** between the ( ) in an indexing expression, we select those indices.

1 2	3 8	5 2
2 0	4 4	6 3

x



goo.gl/YRFAER

Logical Indexing

- We can also use a **logical array** to index.
- In this case, we select all the elements from the source matrix that correspond to positions with a 1 in the logical matrix.

1 2	3 8	5 2
2 0	4 4	6 3

x

x

2	8	2
0	4	3

```
a = x
which = x < 3
x = x(which)
 x = a
x = x(x < 3)
```

Source: https://goo.gl/YRFAER

Web Viewer Terms | Privacy & Cookies

Edit

A few notes

- If you want to hardcode a logical matrix, you need to wrap it up in the `logical` function (otherwise it's just regular numeric `1`s and `0`s).

`[1,0;0,1]`

1	0
0	1

`logical([1,0;0,1])`

1	0
0	1

- You can use “whos” in the command window to check the data types

goo.gl/YRFAER

Writing Into a Logically Indexed Matrix

- One can assign elements in a logically indexed expression.
- The RHS must either be a scalar or have the same number of elements as there are 1s in the logical index matrix.

x

2	8	2
0	4	3

```
a = x
which = x < 3
x(which) = 99
  x = a
x(x < 3) = 88
  x = a
x(x == 2) = -1
x(x+5 == 9) = 22
```

Source: https://goo.gl/YRFAER

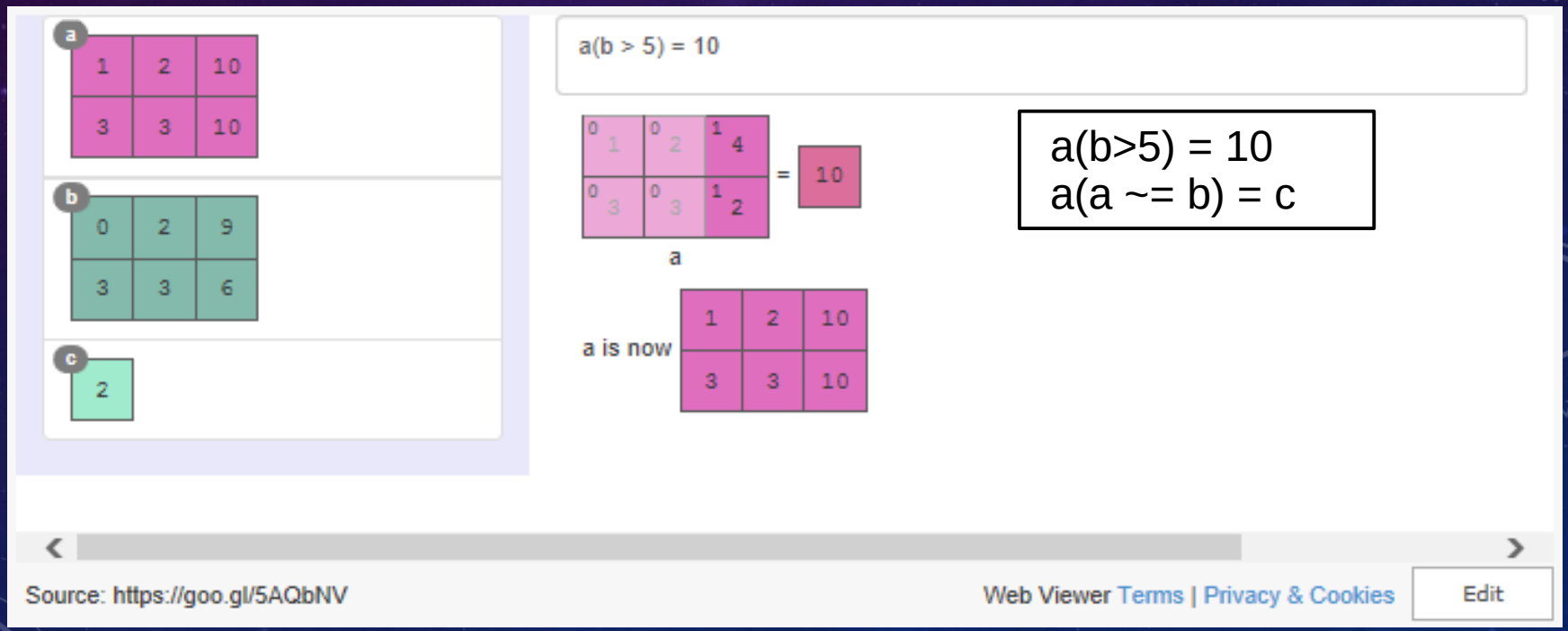
Web Viewer Terms | Privacy & Cookies

Edit

goo.gl/5AQbNV

Logical Indexing with Parallel Matrices

- You can use a logical matrix derived from a different matrix than the source into which you are indexing.
- The two "parallel" matrices should be the same size.



Your turn: Logical Indexing

5 min

goo.gl/VEfrMK

- Write code for each of the following:
  1. Double all elements in a that are between 5 and 10 (inclusive).
  2. Set all matrix elements in b to their absolute value.

https://goo.gl/VEfrMK

Solution: Logical Indexing

- Write code for each of the following:

Double all elements in a that are between 5 and 10 (inclusive).

```
a(5 <= a & a <= 10) = 2 .* a(5 <= a & a <= 10)
```

Set all matrix elements in b to their absolute value.

```
b(b < 0) = -1 .* b(b < 0)
```

I will stay for about one hour
to answer questions (Project 1). Will be behind Stamps Auditorium

See you next time !!!