

CS368 MATLAB Programming

Lecture 5

Young Wu

Based on lecture slides by Michael O'Neill and Beck Hasti

February 23, 2022

Guess Two-Thirds of the Average Game

Quiz

Q1

next time

- Enter an integer between 0 and 100 (including 0 and 100) that is the closest to $\frac{2}{3}$ of the average of everyone's integer.

Comment on Vectorization

Admin

- Please try to avoid using *for* loops and *if* conditionals in the first half of the course.
- The main difference between MATLAB and other programming languages is its very efficient matrix operation implementation.

Boolean Variables

Math

- A Boolean variable, also called *logical* variable type in MATLAB, is a variable with two possible values *true* and *false*.
- A Boolean variable is stored as either 1 for *true* or 0 for *false*.



Indicator Functions

Math

- Indicator functions, also called dummy variables, are functions that return 1 if a condition is satisfied and 0 if the condition is not satisfied.

① $x == y$ is the indicator of $x = y$, meaning $\begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$.

② $x \neq y$ is the indicator of $x \neq y$, meaning $\begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases}$,

~~$x \neq y$~~ does not work in MATLAB.

③ $x > y, x \geq y$ are indicators of $x > y$ and $x \geq y$.

④ $x < y, x \leq y$ are indicators of $x < y$ and $x \leq y$.

Other Logical Functions

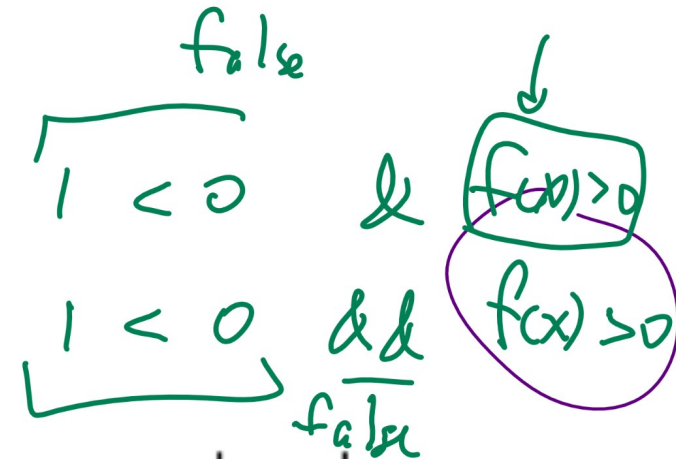
Code

- \sim is not: ~ 0 is 1 and ~ 1 is 0.
- • $\&$ is and: $0 \& 0$ is 0, $0 \& 1$ is 0, $1 \& 0$ is 0, $1 \& 1$ is 1.
- • $|$ is or: $0 | 0$ is 0, $0 | 1$ is 1, $1 | 0$ is 1, $1 | 1$ is 1.
- ↳ • $==$, $\sim =$, $<$, $<=$, $>$, $>=$, \sim , $\&$, $|$ can be applied element-wise to a vector directly.

Short Circuit Evaluation

Code

- `&&` is and, but only works on scalars.
- `||` is or, but only works on scalars.
- `&&` and `||` use short-circuit evaluation, for example, when evaluating `a && b`, if `a` is false, then `b` will not be evaluated, and when evaluating `a || b` if `a` is true, then `b` will not be evaluated.



Vector Reduction Logical Functions

Code

- *any(x)* returns whether any of the elements in the matrix or vector x is non-zero.
- *all(x)* returns whether all of the elements in the matrix or vector x is non-zero.
- *find(x)* finds the index of all the non-zero elements in the vector x .
- *find(x, 1)* finds the index of the first non-zero element in the vector x .

Other Reduction Functions

Code

- $sum(x)$ and $prod(x)$ compute the sum and product of the elements in a matrix or vector x . $(1, 1, 0, 0)$
- $sum(x, 1)$ and $prod(x, 1)$ compute the column sums and products of the elements in a matrix x , for example, $sum([1\ 2; 3\ 4], 1)$ returns the column sums $[4\ 6]$. 2
- $sum(x, 2)$ and $prod(x, 2)$ compute the row sums and products of the elements in a matrix x , for example, $sum([1\ 2; 3\ 4], 2)$ returns the row sums $\begin{bmatrix} 3 \\ 7 \end{bmatrix}$. $\begin{bmatrix} \text{---} \\ \text{---} \end{bmatrix} (:)$
 $(! ! \dots)$
- $mean(x)$ computes the average of the numbers in a matrix or vector x .
- $max(x)$ and $min(x)$ compute the maximum and minimum of the elements in a matrix or vector x .

Indicator, Quiz Grade

Quiz

- (Compute the number of questions a student gets incorrect if the student's answers are *B, C, D* and the correct answers are *B, D, D*?)

• 2

① $a = ['B', 'C', 'D']; s = ['B', 'D', 'D'];$

• C: $sum(a == s)$

• D: $sum(a \neq s)$

• E: ~~$sum(a \neq s)$~~ (this is not MATLAB)

→ correct

Q2

$\Rightarrow a == s$

$$\sum [1, 0, 1] = 2$$

Indicator, Letter Grade

Quiz

$$\text{sum}(0, 1, 1, 1, 1) + 1 = 5$$

Q4
s(5) Error.

- (Compute letter grade if A corresponds to a grade ≥ 90 , B for a grade ≥ 80 , C for a grade ≥ 70 , and D otherwise.)

- 'C' 95 $\text{sum}[1, 0, 0, 0, 0] = 1$

```
1 g = 75; c = [101 90 80 70 0]; s = ['A' 'B' 'C' 'D'];
```

- C: $s(\text{sum}(g \geq c) + 1)$

- D: $s(\text{sum}(g < c))$

GPA.

0	0	0	1	1
---	---	---	---	---

$$\text{sum}(C \ 1 \ 1 \ 1 \ 0 \ 0) = 3$$

$$2 + 1 = 3$$

Indicator, Letter Grades

Quiz

- (Compute letter grades if *A* corresponds to a grade ≥ 90 , *B* for a grade ≥ 80 , *C* for a grade ≥ 70 , and *D* otherwise.)

Q5

'ACD'

1

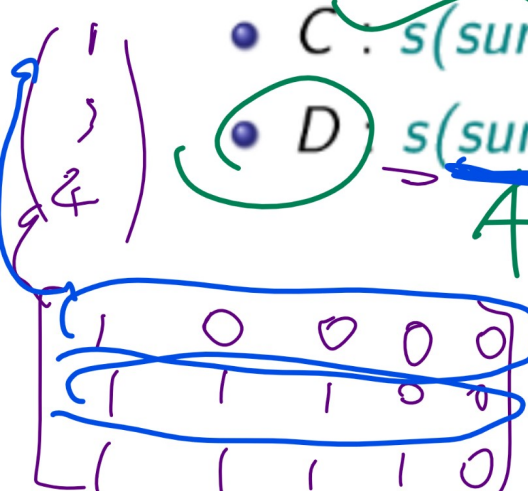
$g = [95 \ 75 \ 65]; c = [101 \ 90 \ 80 \ 70 \ 0]; s = ['A' \ 'B' \ 'C' \ 'D'];$

$s(x) \ s(2)$

• $C : s(\text{sum}(\text{repmat}(g', 1, 5) < \text{repmat}(c, 3, 1)) \neq 1)$

• $D : s(\text{sum}(\text{repmat}(g', 1, 5) < \text{repmat}(c, 3, 1)) \neq 2)$

sum over rows.



$$= \begin{bmatrix} 95 & 95 & 95 & 95 & 95 \\ 75 & 75 & 75 & 75 & 75 \\ 65 & 65 & 65 & 65 & 65 \end{bmatrix}$$

$$\begin{bmatrix} 101 & 90 & 80 & 70 & 0 \\ 101 & 90 & 80 & 70 & 0 \\ 101 & 90 & 80 & 70 & 0 \end{bmatrix}$$

matrix

Functions

Math

- A function $y = f(x)$ is a mapping from a list of inputs x , also called arguments or parameters, to a list of outputs y .
- The previous lectures covered many built-in functions in MATLAB, for example, log has 1 input and 1 output, + has 2 inputs and 1 output, and size has 1 input and 2 outputs.
- New functions can be defined in .m files and used in commands.

$$1+1 \Rightarrow + (1, 1)$$

$$[2 \ 2] \leftarrow \text{size}([1 \ 2 \ 3 \ 4])$$

Function Definition

Code

- A function with name f should be put in a file named $f.m$.
- The first line of the file is `function y = f(x)` or `function [y1, y2, ...] = f(x1, x2, ...)`, where y is the name or names of the variables to return, and x is the list of arguments of the function.
- The second line of the file is usually comments describing what the function does. Comments start with `%` the line after `%` is not executed by the program.
- The last line of the file should be `end`, but it can be omitted.

Helper Functions

Code

- Multiple functions can be defined in the same file *f.m*, but only *f* can be used outside the file in commands.
- The functions in *f.m* that is not *f* are helper functions.

Function Example, Addition

Code

- The addition function $x + y$ is usually written in infix notation (argument 1, then function name, then argument 2).
- The following function is the addition function in prefix notation (function name, then argument 1, then argument 2).

```

1 function z = add(x, y)
2   z = x + y;
3 end

```

add(1, 2) returns **3**.

~~return z~~

add.m

Function Example, Linear Combination

Code

- The linear combination of x and y with coefficients u and v is $ux + vy$.
- Sometimes, u, v are not specified, so the default value $u = v = 1$ is used.

$$u \ x + v \ y$$

1 `function z = lincom(x, y, u, v)`

2 `arguments`

3 `x; y; u = 1; v = 1;`

4 `end`

5 `z = u * x + v * y;`

6 `end`

- `arguments` block is also used for input validation. More detail in a later lecture.

Function Example, Linear Combination Too

Code

```

1 function z = lincom(x, y, u, v)
2     arguments
3     x; y; u = 1; v = 1;
4 end
5     z = u * x + v * y;
6 end
    
```

lincom.m

- lincom(1, 2, 3, 4) returns 11. $3 \cdot 1 + 4 \cdot 2 = 11$
- lincom(1, 2, 3) returns 5. $1 \cdot 2 + 3 \cdot 1 = 5$
- lincom(1, 2) returns 3. $1 \cdot 2 + 1 \cdot 1 = 3$

lincom(1) → Error.

Function Example, Max and Min

Code

- Multiple values can be returned from a function,
 $[y_1, y_2, \dots, y_n] = f(x)$ stores the value of i -th output in y_i for $i = 1, 2, \dots, n$ and $f(x)$ only returns first output.

```

1 function [mx, mn] = mxn(x)
2     mx = max(x);
3     mn = min(x);
4 end
    
```

range ?

ans → mxn([1, 2, 3]) returns 3 ⇒ ans = 3.

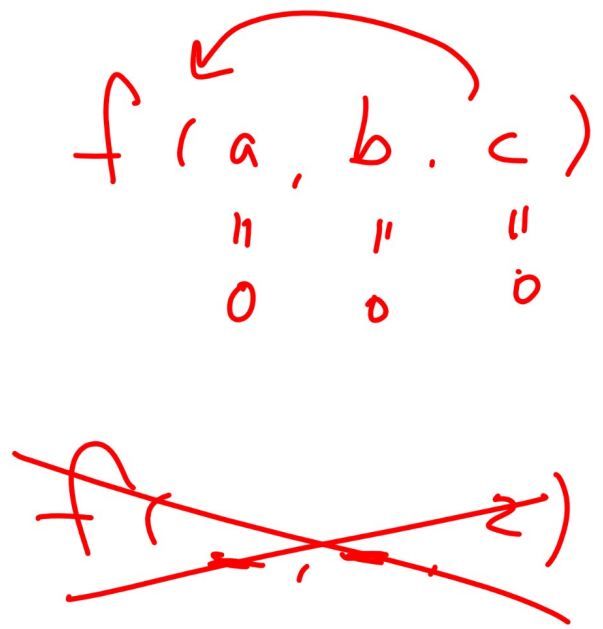
[a, b] = mxn([1, 2, 3]) sets a to 3 and b to 1.

Functions, Vector Output

Quiz

- 1 `function v = f1(x)`
 - 2 `v = [x, x + 1];`
 - 3 `end`
 - 1 `sum(f1(2))`
 - B : 2
 - C : 3
 - D : 5
-

Q6



Functions, Multiple Outputs

Quiz

Q7

```
1 function [u, v] = f2(x)  
2   u = x; v = x + 1;  
3 end
```

```
1 sum(f2(2))
```

- B : 2
- C : 3
- D : 5

return [x, x+1]

return ~~x~~, x+1

sum(2)

update P2, P3 temporary after midnight.

Blank Slide