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

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.
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. \( x \sim= y \) 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\text{ is not}: \sim0 \text{ is 1 and } \sim1 \text{ is 0.}
- \& \text{ is and}: 0 \& 0 \text{ is 0, } 0 \& 1 \text{ is 0, } 1 \& 0 \text{ is 0, } 1 \& 1 \text{ is 1.}
- | \text{ is or}: 0 | 0 \text{ is 0, } 0 | 1 \text{ is 1, } 1 | 0 \text{ is 1, } 1 | 1 \text{ is 1.}
- ==, \sim=, <, \leq, >, \geq, \sim, \&, | \text{ 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

- \texttt{any(x)} returns whether any of the elements in the matrix or vector \( x \) is non-zero.
- \texttt{all(x)} returns whether all of the elements in the matrix or vector \( x \) is non-zero.
- \texttt{find(x)} finds the index of all the non-zero elements in the vector \( x \).
- \texttt{find(x, 1)} finds the index of the first non-zero element in the vector \( x \).
Other Reduction Functions

Code

- \( \text{sum}(x) \) and \( \text{prod}(x) \) compute the sum and product of the elements in a matrix or vector \( x \).
- \( \text{sum}(x, 1) \) and \( \text{prod}(x, 1) \) compute the column sums and products of the elements in a matrix \( x \), for example, \( \text{sum}([1 \ 2; \ 3 \ 4], \ 1) \) returns the column sums \([4 \ 6] \).
- \( \text{sum}(x, 2) \) and \( \text{prod}(x, 2) \) compute the row sums and products of the elements in a matrix \( x \), for example, \( \text{sum}([1 \ 2; \ 3 \ 4], \ 2) \) returns the row sums \([3 \ 7] \).
- \( \text{mean}(x) \) computes the average of the numbers in a matrix or vector \( x \).
- \( \text{max}(x) \) and \( \text{min}(x) \) compute the maximum and minimum of the elements in a matrix or vector \( x \).
(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. $a = ['B', 'C', 'D']; s = ['B', 'D', 'D'];$

   - $C : \text{sum}(a == s)$
   - $D : \text{sum}(a \sim= s)$
   - $E : \text{sum}(a \neq s)$ (this is not MATLAB)
(Compute the GPA if C is worth 1 point and N is worth 0 point for a student whose grades are C, C, N.)

0.5

```plaintext
1 g = ['C', 'C', 'N', 'N'];
```

```plaintext
C : (1 * (g == 'C') + 0 * (g == 'N')) / length(g)
```

```plaintext
D : (1 * sum(g == 'C') + 0 * sum(g == 'N')) / length(g)
```
(Compute letter grade if A corresponds to a grade $\geq 90$, B for a grade $\geq 80$, C for a grade $\geq 70$, and D otherwise.)

'C'

```latex
\texttt{g = 75; c = [101 90 80 70 0]; s = ['A' 'B' 'C' 'D'];}
\texttt{C : s(sum(g \geq c) + 1)}
\texttt{D : s(sum(g < c))}
```
(Compute letter grades if $A$ corresponds to a grade $\geq 90$, $B$ for a grade $\geq 80$, $C$ for a grade $\geq 70$, and $D$ otherwise.)

'C'

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

\[ C : s(sum(repmat(g', 1, 5) < repmat(c, 3, 1)) + 1) \]

\[ D : s(sum(repmat(g', 1, 5) < repmat(c, 3, 1), 2)) \]
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 $\text{size}$ has 1 input and 2 outputs.

New functions can be defined in \textit{.m} files and used in commands.
A function with name $f$ should be put in a file named $f.m$.

The first line of the file is \textit{function $y = f(x)$} or \textit{function $[y_1, y_2, ...] = f(x_1, x_2, ...)\),\) 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 \textit{end}, but it can be omitted.
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.
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).

```matlab
function z = add(x, y)
    z = x + y;
end
```

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

```matlab
function z = lincom(x, y, u, v)
    arguments
    x; y; u = 1; v = 1;
    end
    z = u * x + v * y;
end
```

- `arguments` block is also used for input validation. More detail in a later lecture.
Function Example, Linear Combination Too

Code

```
function z = lincom(x, y, u, v)
    arguments
    x; y; u = 1; v = 1;
    end
    z = u * x + v * y;
    end

lincom(1, 2, 3, 4) returns 11 .
lincom(1, 2, 3) returns 5 .
lincom(1, 2) returns 3 .
```
Function Example, Max and Min

Code

Multiple values can be returned from a function,

\[ [y_1, y_2, \ldots, y_n] = f(x) \]

stores the value of \( i \)-th output in \( y_i \)
for \( i = 1, 2, \ldots, n \) and \( f(x) \) only returns first output.

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

- \( mxn([1, 2, 3]) \) returns \( 3 \)
- \([a \ b] = mxn([1, 2, 3]) \) 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. \( \text{sum}(f1(2)) \)
   - \( B : 2 \)
   - \( C : 3 \)
   - \( D : 5 \)
Functions, Multiple Outputs

Quiz

1. \( \text{function } [u, \ v] = f2(x) \)
2. \( u = x; \ v = x + 1; \)
3. \( \text{end} \)

1. \( \text{sum}(f2(2)) \)

- \( B : 2 \)
- \( C : 3 \)
- \( D : 5 \)
Blank Slide