CS368 MATLAB Programming

Lecture 10

Young Wu

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

April 6, 2022

Multiple Choice

Quiz

- Which answer(s) is correct?
- A : All of the below.
- B : None of the below.
- C : All of the above.
- D : One of the above.
- E : None of the above.

Indicator Variables

Math

- If the same task is performed for different values of a variable, use an indicator variable and vectorize.
- If different tasks are performed for different values of a variable, use a *switch* conditional.
- If different tasks are performed under different conditions, use an *if* conditional.

Switch

Code

- Different tasks are performed for $x = v_1$, for $x = v_2$ or v_3, and for every other value of x.

```
switch x
    case v1
        ...
    case {v2, v3}
        ...
    otherwise
        ...
end
```

If Else

Code

- Different tasks are performed if $x \neq 0$, if $x = 0$ but $y \neq 0$, and if $x = 0$ and $y = 0$.

```
if x
    ...
elseif y
    ...
else
    ...
end
```

Condition for If

Code

- *if x* and *if x ~= 0* represent the same condition. The expression *x ~= 0* should be treated as a variable whose value is $\begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$.
- *while x* and *while x ~= 0* represent the same loop for the same reason.

Conditionals, Switch

Quiz

```
x = 10; switch mod(x, 4)
    case 0
        x + 1
    case {1, 2}
        x * 2
    otherwise
        x ^ 3
end
```

- A : 11
- B : 20
- C : 1000

Conditionals, If

Quiz

```
x = 10;
 if  x < 10 && ~mod(x, 2)
    x + 1
 elseif  ~mod(x , 3)
    x * 2
 else
    x ^ 3
end
```

- A : 11
- B : 20
- C : 1000

Conditionals, Variable as Condition

Quiz

```
x = 0; y = 1; z = 2;
 if  x && ~y && z
    x
 elseif  x  ||  ~y  ||  z
    y
 else
    z
end
```

- $A : 0$
- $B : 1$
- $C : 2$

Number of Input Arguments

Code

- When the function *function* $z = f(x, y)$ is called, $0, 1$ or 2 arguments can be provided.
- *switch* can be used here to perform different tasks when different number of arguments are given.
- *nargin* is the number of input arguments provided when the function is called.

Log with Optional Input Arguments

Code

- For example, a new log function can be defined by *log()* returns 1, *log(x)* returns natural $\log(x)$, and *log(n, x)* returns $\log_n(x)$.

```
function z = log(x, y)
  switch nargin
    case 1
      z = log(x);
    case 2
      z = log(y) / log(x);
    otherwise
      z = 1;
  end; end
```

Variable Length Input Argument

Code

- $varargin$ represents an arbitrary number of input variables.
- It can only be used as the last argument of a function, for example, $function\ \ y = f(x1,\ x2,\ x3,\ varagin)$.
- The i-th argument can be accessed by $varargin\{i\}$.

Log with Variable Length Argument

Code

- For example, a new log function can be defined so that it returns a vector if more than one input is provided.

```
function z = log(x, varagin)
   if nargin == 1
     z = log(x);
   else
     z = [log(x) zeros(1, nargin - 1)];
     for t = 2:nargin
       z(t) = log(varagin{t - 1});
     end; end; end
```

Output Arguments

Code

- *varargout* represents an arbitrary number of output variables.
- *nargout* represents the number of output variables assigned when the function is called.
- For example, $x = size([1\ \ 2;\ 3\ \ 4])$ assigns x the value **2 2** and $[x,\ y] = size([1\ \ 2;\ 3\ \ 4])$ assigns x the value **2** .

Recursion

Math

- A function that uses itself in the body is called a recursive function.

```
function z = f(x)
if x ... % base case
    z = ...
else % recursion
    z = ... f(x') ...
end
```

Recursion Example, Factorial

Code

- To compute the factorial of $n \geqslant 0$:

```
function z = f(x)
    if ~x
      z = 1;
    else
      z = x * f(x - 1);
    end
end
```

Recursion Example, Vector Sum

Code

- To compute the sum of the values in a vector *v*:

```
function z = f(x, t)
   if nargin == 1
     z = f(x, 1);
   elseif t > length(x)
     z = 0;
   else
     z = x(t) + f(x, t + 1);
   end
end
```

Recursion, Fibonacci

Quiz

```
function z = fib(x)
   if x < 3
      z = 1;
   else
      z = fib(x - 1) + fib(x - 2);
   end
end
fib (5)
```

- A : 5
- B : 7

Recursion, Binomial

Quiz

```
function  z = combin(x, y)
    if  y == 0 || y == x
      z = 1;
    else
      z = combin(x − 1, y) + combin(x − 1, y − 1);
    end
end
combin(3, 2)
```

- $A : 3$
- $B : 6$

Recursion, Greatest Common Divisor

Quiz

```
function z = gcd(x, y)
    if y == 0
        z = x;
    else
        z = gcd(y, mod(x, y));
    end
end
gcd(10, 6)
```

- A : 2
- B : 6

Recursion, Greatest Common Divisor Again

Quiz

```
function z = gcd(x, y)
    if y == 0
        z = x;
    else
        z = gcd(y, mod(x, y));
    end
end
gcd(9, 16)
```

- $A : 1$
- $C : 9$

Blank Slide