Chapter 4
Execution Control

Outline

4.1 Concept: Code Blocks

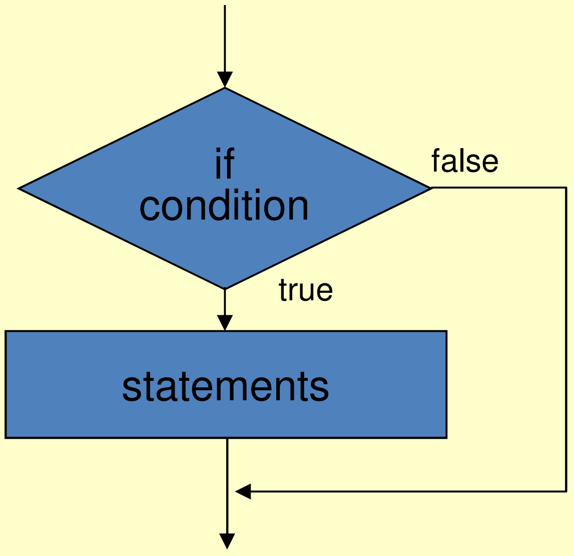
- We have noted already that a MATLAB function abstracts away the details of an operation like sin(Θ)
- We are about to study three chapters with different needs to make explicit a collection of one or more lines of MATLAB code – a code block
- Some languages enclose code blocks in braces {…}
- MATLAB defines some key words with specific functionality: **`if`**, **`else`**, **`elseif`**, **`end`** , **`case`**, **`otherwise`**, **`for`** and **`function`** , for example
- Code blocks in MATLAB are always enclosed between key words, and are usually indented although there is no significance to the indentation

4.2 Conditional Execution

There are two forms of conditional execution:

- **<u>If statements</u>** apply one or more tests to existing variables in the workspace. Only if the test produces a true result is a specific code block executed.
- **<u>Switch statements</u>** consider one variable in the workspace and offer one or more code blocks when that variable has certain exact values.
- In both forms, provision is made for a code block to be executed when none of the specified conditions are true

4.3 Simple if Statements

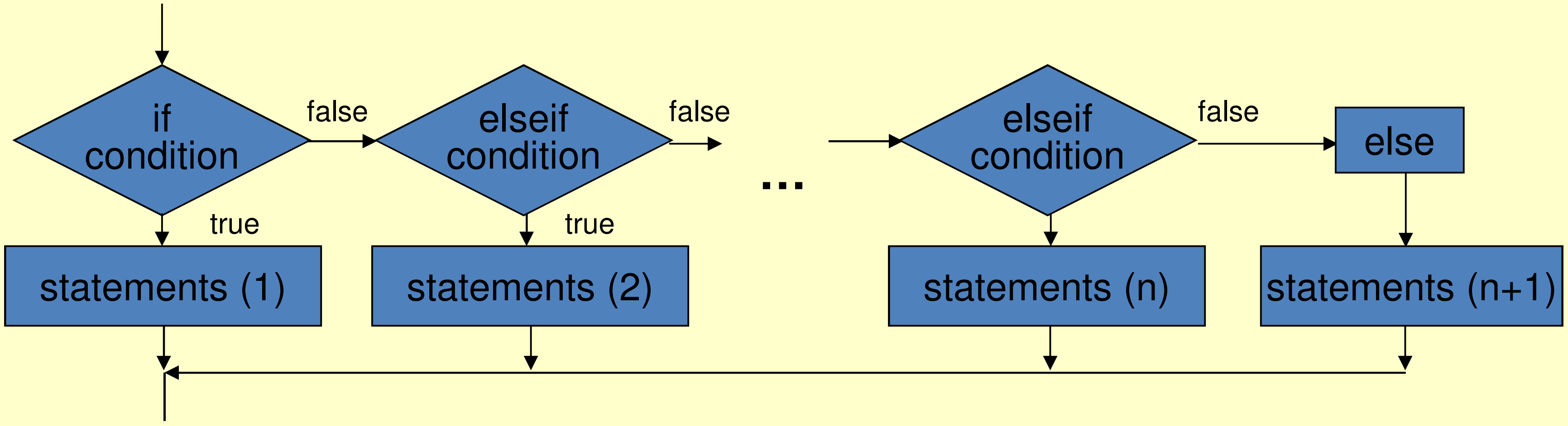


- Simple conditional execution applies a test to some data and if the result of that test is true, the statements in the code block are executed.
- Otherwise, control passes to the statement following the end of the code block.

e.g.
```
if a == 4
    b = 7;
end
```

Compound if Statements

- If one test fails, subsequent tests can be applied using the `elseif` key word.
- When a code block is executed, control is passed to the statement following the end of the `if` statement
- The `else` key word defines a code block to be executed when none of the conditions are true
- In a single `if` statement, only one code block will ever execute



If statements

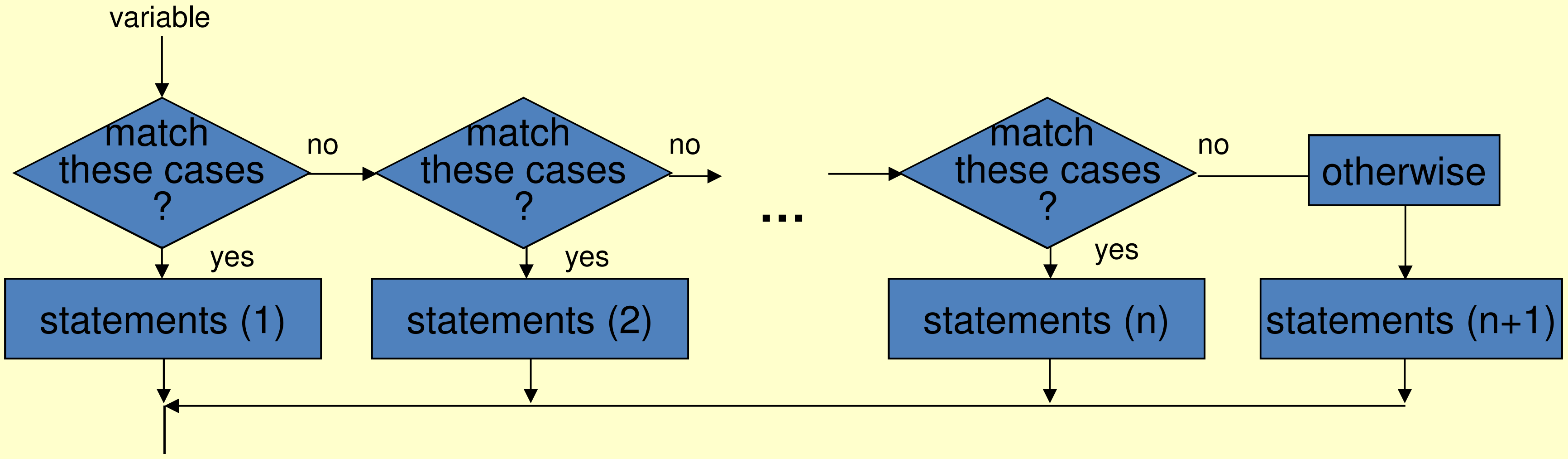
```
if <logical expression 1>
  <code block 1>
elseif <logical expression 2>
  <code block 2>
  . . .
elseif <logical expression n>
  <code block n>
else
  <default code block>
end
```

If statement example

```
grade = input('what grade? ');
if grade >= 90
  letter = 'A'
elseif grade >= 80
  letter = 'B'
elseif grade >= 70
  letter = 'C'
elseif grade >= 60
  letter = 'D'
else
  letter = 'F'
end
```

4.4 switch Statements

- Rather that applying a general test, switch statements examine one variable and select one code block based on specified values of that variable
- Specified values are called cases, and multiple values may be enclosed in braces, {…}



switch statements

```
switch <parameter>
   case <case specification 1>
        <code block 1>
   case <case specification 2>
        <code block 2>
   . . .
   case <case specification n>
        <code block n>
   otherwise
        <default code block>
end
```

Switch example

```
switch day
  case {2, 3, 4, 5, 6}
     weekend = false
  case {1, 7}
     weekend = true
  else
     error('bad day value')
end
```

4.5 Iteration in General

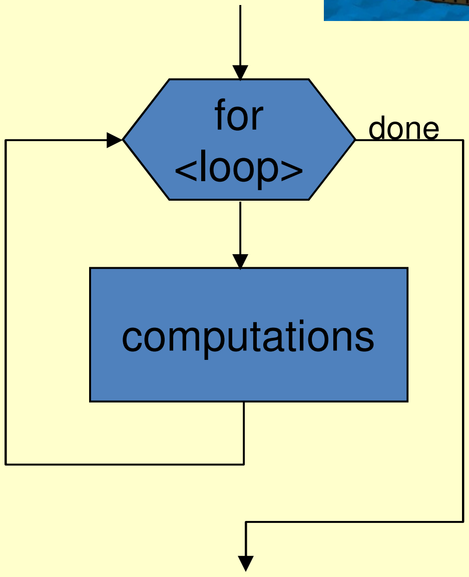
The second purpose for code blocks is to define sets of instructions that need to be repeated 0 or more times. Iteration in MATLAB comes in two forms:

- For loops where the number of repetitions is essentially fixed
- While loops where the number of repetitions is under the control of the data being processed

Under special circumstances, one can program an early exit from iteration with the **`break`** statement, and you can skip to the next repetition of the iteration with the **`continue`** statement.

4.6 for Loops

Although there is only one style of for loop, it can be used in two ways.



- The general form is:

```
for <value> = <vector>
    <code block>
 end
```

- The number of repetitions is the length of the vector and the code block is executed with the variable set to the value of each element of the vector in order.
- Changing the value of the variable within the code block has no effect on the iteration
- This form does not allow the values in the vector to be changed

Indexed for Loops

- The indexed form of the for loop is:

```
for <index> = 1:length(<vector>)
    <code block>
 end
```

- The number of repetitions is the length of the vector and the code block is executed with the index set to the position of each element of the vector in order.
- Within the code block, values of the vector are fetched by:

```
value = vector(index)
```

  and changed by

```
vector(index) = new_value
```

For loop example

```
for ndx = 1:length(grades)
   if grades(ndx) >= 70 && grades(ndx) <=
     100
     passes = passes + 1
   elseif grades(ndx) >= 0
     fails = fails + 1
   else
     grades(ndx) = -1
   end
end
```

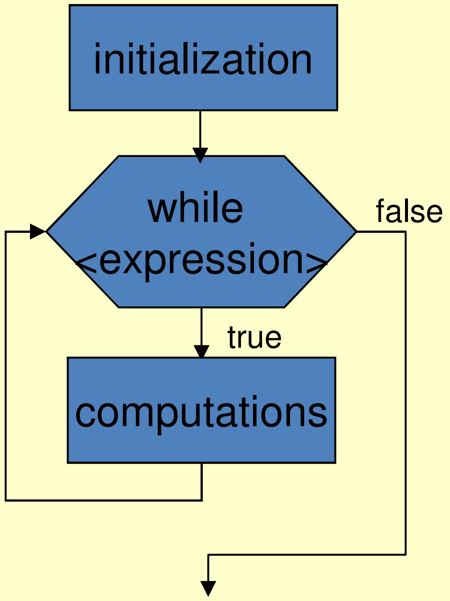
For loop example

```
A = [6 12 6 91 13 6]
theMax = A(1);
for x = A
   if x > theMax
     theMax = x;
   end
end
fprintf(‘max(A) is %d\n’,theMax);
```

4.7 while Loops

In some sense, a while loop is a do-it-yourself for loop. The general form is:

```
<initialize the condition>
while <condition>
     <code block>
end
```



The loop will continue as long as the condition returns **`true`**.

Clearly, this will execute indefinitely unless something happens in the code block to change the outcome of the test condition.

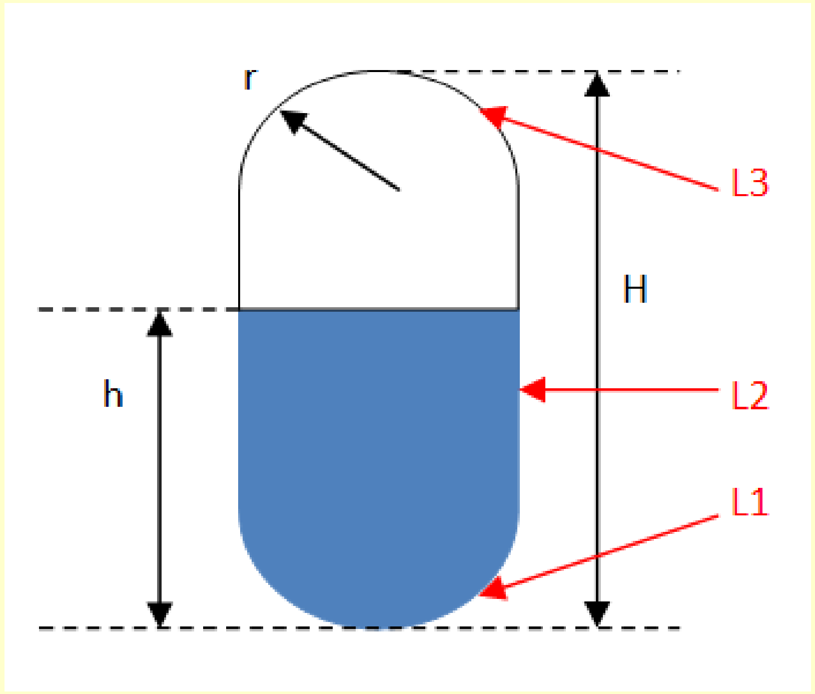
While loop example

```
day = 1
while day > 0 && day <= 7
   day = input('please enter a day: ' );
   switch day
     case {2, 3, 4, 5, 6}
        weekend = false
     case {1, 7}
        weekend = true
     else
        error('bad day value')
   end
end
```

Let’s Write some Code …

4.8 Engineering Example – Computing Liquid Levels

A cylindrical tank of height H and radius r with a spherical cap on each end (also of radius, r). If the height of the liquid is h, what is the volume of liquid in the tank? Clearly, the calculation of the volume of liquid in the tank depends upon the relationship between h, H, and r:



4.8 Engineering Example – Computing Liquid Levels

- L1: we need the volume, v, of a partially filled sphere given by:

$$v = \frac{1}{3}\pi h^2(3r - h)$$

- L2: we need the volume of a fully filled hemisphere plus the volume of a cylinder:

$$v = \frac{2}{3}\pi r^3 + \pi r^2(h - r)$$

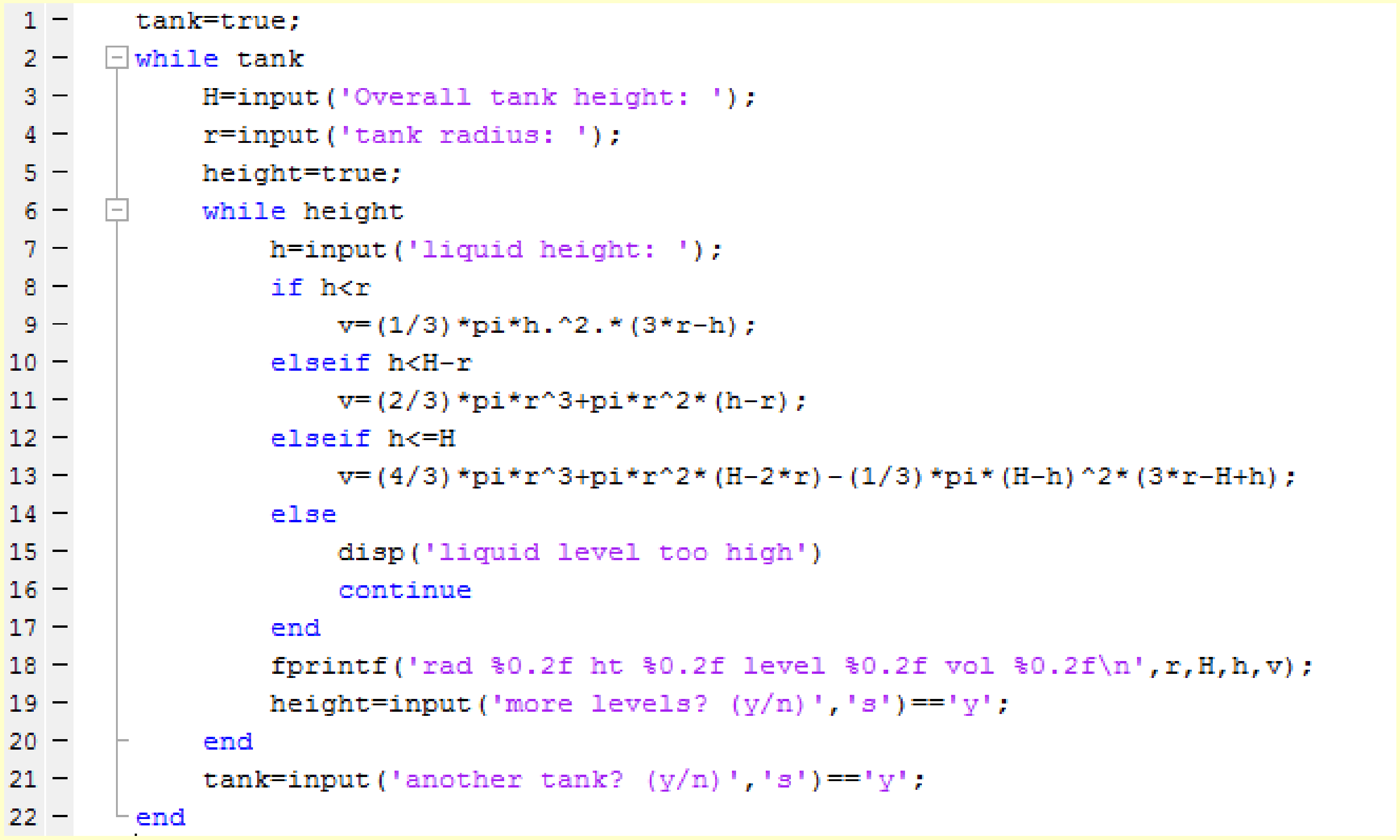
4.8 Engineering Example – Computing Liquid Levels

- L3: we need the volume of a fully filled sphere plus the volume of a cylinder minus partially empty upper hemisphere:

$$v = \frac{4}{3}\pi r^3 + \pi r^2(H - 2r) - \frac{1}{3}\pi(H - h)^2(3r - H + h)$$

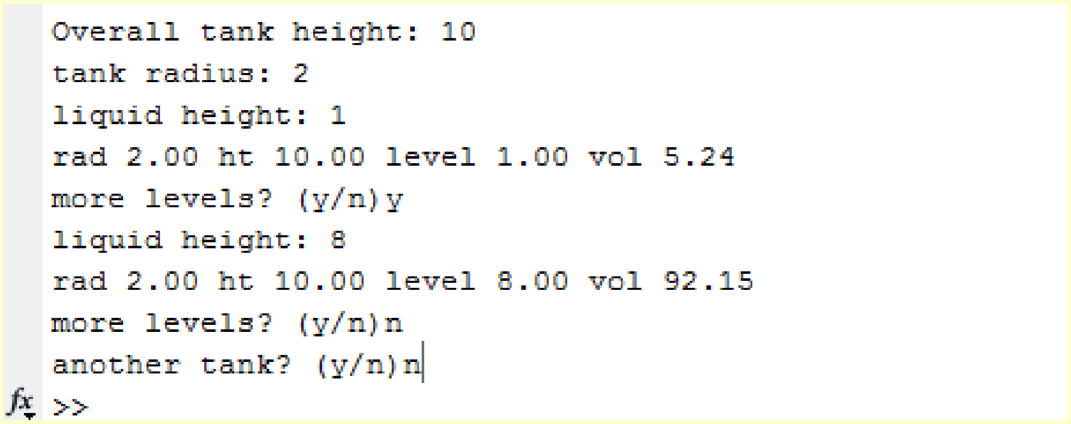
We should write the script so that we continue to consider tanks of different dimensions and different liquid heights for each tank until the user indicates that he/she needs no more results.

4.8 Engineering Example – Computing Liquid Levels



```
tank=true;
while tank
    H=input('Overall tank height: ');
    r=input('tank radius: ');
    height=true;
    while height
        h=input('liquid height: ');
        if h<r
            v=(1/3)*pi*h.^2.*(3*r-h);
        elseif h<H-r
            v=(2/3)*pi*r^3+pi*r^2*(h-r);
        elseif h<=H
            v=(4/3)*pi*r^3+pi*r^2*(H-2*r)-(1/3)*pi*(H-h)^2*(3*r-H+h);
        else
            disp('liquid level too high')
            continue
        end
        fprintf('rad %0.2f ht %0.2f level %0.2f vol %0.2f\n',r,H,h,v);
        height=input('more levels? (y/n)','s')=='y';
    end
    tank=input('another tank? (y/n)','s')=='y';
end
```

4.8 Engineering Example – Computing Liquid Levels



```
Overall tank height: 10
tank radius: 2
liquid height: 1
rad 2.00 ht 10.00 level 1.00 vol 5.24
more levels? (y/n)y
liquid height: 8
rad 2.00 ht 10.00 level 8.00 vol 92.15
more levels? (y/n)n
another tank? (y/n)n
>>
```

Homework on Chapter 4 is posted on the website:

http://www.ee.nmt.edu/~erives/289_F12/EE289.html

Homework is due in a week