- Previous topics:
  - Branching
  - Boolean expressions
- Now:
  - Introduction to **for**-loop

## Question

A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?

Physical experiment? ◆

Thought experiment? → analysis

Computational experiment! → simulation◆

◆Need to repeat many trials!

2

```
% one trial of the experiment
breakPt= rand;
if  breakPt<0.5
   shortPiece= breakPt;
else
   shortPiece= 1-breakPt;
end
```

3

```
% one trial of the experiment
breakPt= rand;
shortPiece= min(breakPt, 1-breakPt);
```

Want to do many trials, add up the lengths of the short pieces, and then divide by the number of trials to get the average length.

4

Repeat n times

```
% one trial of the experiment
breakPt= rand;
shortPiece= min(breakPt, 1-breakPt);
```

Take average

Print result

5

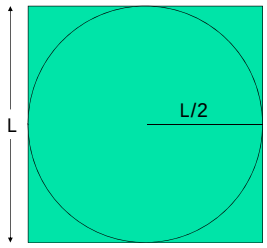
```
n= 10000;  % number of trials
total= 0;  % accumulated length so far

for  k= 1:n
   % one trial of the experiment
   breakPt= rand;
   shortPiece= min(breakPt, 1-breakPt);
   total= total + shortPiece;
end

aveLength= total/n
fprintf('Average length is %f\n', ...
                              aveLength)
```

6

## Monte Carlo Estimation of π




Throw *N* darts

Sq. area = $N = L \times L$

Circle area = $N_{in}$
$= \pi L^2 / 4$

$\pi = 4\ N_{in} / N$

8

## Monte Carlo Approximation of Pi

For each of N trials
  Throw a dart
  If it lands in circle
    add 1 to total # of hits

Pi is 4*hits/N

9

## Monte Carlo Pi with N darts on L-by-L board

```
hits = 0;
for k = 1:N
   % Throw kth dart
   x = rand*L - L/2;
   y = rand*L - L/2;
   % Is it in the circle?
   if sqrt(x^2+y^2) <= L/2
      hits = hits + 1;
   end
end
myPi = 4*hits/N;
```

14

## Syntax of the `for` loop

```
for <var>= <start value>:<incr>:<end bound>
    statements to be executed repeatedly
end
```

Loop body

15

## Syntax of the `for` loop

```
for <var>= <start value>:<incr>:<end bound>
    statements to be executed repeatedly
end
```

Loop header specifies all the values that the index variable will take on, one for each pass of the loop.

E.g, `k= 3:1:7` means `k` will take on the values 3, 4, 5, 6, 7, one at a time.

16

## Pattern for doing something *n* times

```
n= _____
for k= 1:n
    % code to do
    % that something
end
```

17

## `for` loop examples

```
for k= 2:0.5:3
     disp(k)
end
```
k takes on the values _____
Non-integer increment is OK

```
for k= 1:4
     disp(k)
end
```
k takes on the values _____
Default increment is 1

```
for k= 0:-2:-6
     disp(k)
end
```
k takes on the values _____
"Increment" may be negative

```
for k= 0:-2:-7
     disp(k)
end
```
k takes on the values _____
Colon expression specifies a *bound*

```
for k= 5:2:1
     disp(k)
end
```



## Example: count down to blast off

- Write a script **`countDown`** that produces this screen output after asking user to input starting value (4 in this example):

```
T = 4 seconds
T = 3 seconds
T = 2 seconds
T = 1 second
Blast off!!!!
```



```
% What will be printed?
for k= 1:2:6
    fprintf('%d ', k)
end
```

- A: 1 2 3 4 5 6
- B: 1 3 5 6
- C: 1 3 5
- D: *error (incorrect bounds)*



```
% What will be printed?
for k= 10:-1:14
    fprintf('%d ', k)
end
fprintf('!')
```

- A: *error (incorrect bounds)*
- B: 10 *(then error)*
- C: 10 !
- D: 14 !
- E: !