Previous Lecture:
- Intro to the course, “Computational senses”
- Running a program in Matlab

Today’s Lecture:
- Anatomy of a program
- Variables, assignment, mathematical operations
- Functions for input & output

Announcements
- Due to fixed lab capacity, you must attend the discussion section in which you are enrolled
- Consulting begins Sunday in ACCEL Green Room (Carpenter Hall)
- “InstallFests” for remote MATLAB to be announced

Formula
- Surface area of a sphere: \( A = 4\pi r^2 \)
- Have the cosine of some angle and want \( \cos(\theta/2) \):
  \[
  \cos(\theta/2) = \sqrt{\frac{1 + \cos(\theta)}{2}}
  \]

A computer program

```matlab
% Example 1_1: Surface area of a sphere
% A: surface area of the sphere
% r: radius of the sphere

r = input('Enter the radius: '); % input
A = 4 * 3.14159 * r * r; % computation
fprintf('Surface area is %f\n', A); % output
```

Variable & assignment

- **Variable**: a named computer memory space for storing a value
  - \( r \) \
  - \( A \)
  - Valid names start with a letter, can contain digits
  - Use meaningful variable names!

Variable & assignment

- **Variable**: a named space for storing a value
  - \( r = 2 \times 4.5 \)
- **Assignment**: putting a value into a variable
- **Assignment operator**: `=`
- **An assignment statement**: \( r = 2 \times 4.5 \)
- **Expression** on right-hand-side (rhs) is evaluated before the assignment operation
Assignment

- **Expression on rhs** is evaluated before the assignment operation.
- Examples:
  
  ```
  x = 2 * 3.14
  y = 1 + x
  z = 4^2 - cos(y)
  ```
- **Question**: can we reverse the order of the 3 statements above?

Script execution

(A script is a sequence of statements, an "m-file")

```matlab
% Quad1
% Solves x^2 + 5x + 6 = 0

a = 1;
b = 5;c = 6;
d = sqrt(b^2 - 4*a*c);
r1 = (-b - d)/(2*a)
r2 = (-b + d)/(2*a)
```

Memory space

Statements in a program are executed in sequence

```matlab
% A program fragment ... x = 2 * 3.14
y = 1 + x
x = 5
% What is y now?
```

Input & output

- `variable = input('prompt')`
- `r = input('Enter radius: ')`

```matlab
fprintf('Increase is %f inches
', x)
fprintf('Position (%d, %d)
', x, y)
```

Substitution sequences (conversion specifications)

- `%f` fixed point (or floating point)
- `%d` decimal—whole number
- `%e` exponential
- `%g` general—Matlab chooses a format
- `%c` character
- `%s` string

Examples:

- `%f
  %15.2f

Comments

- For readability!
- A comment starts with `%` and goes to the end of the line
- Start each program (script) with a **concise** description of what it does
- Define each important variable/constant
- Top a block of code for a specific task with a **concise** comment
Example

Modify the previous program to calculate the increase in surface area in mile$^2$ given an increase in the radius of a sphere.

Note: 1 mile = 5280 feet

% Example 1_2: Surface area increase
% given an increase in the radius
r= input('Enter radius r in miles: ');
delta= input('Enter delta r in inches: ');

% Example 1_2: Surface area increase
% given an increase in the radius
r= input('Enter radius r in miles: ');
delta= input('Enter delta r in inches: ');

What's next?

- So far, all the statements in our scripts are executed in order
- We do not have a way to specify that some statements should be executed only under some condition
- We need a new language construct...

Consider the quadratic function
\[ q(x) = x^2 + bx + c \]
on the interval \([L, R]\):

- Is the function strictly increasing in \([L, R]\)?
- Which is smaller, \(q(L)\) or \(q(R)\)?
- What is the minimum value of \(q(x)\) in \([L, R]\)?

- What are the critical points?