CS 1112
Introduction to Computing Using MATLAB
DR. CURRAN MUHLBERGER
WWW.CS.CORNELL.EDU/COURSES/CS1112
Lecture 2: Programming basics

- Previous lecture & lab:
  - Intro to the course
  - “Computational senses”
  - Running commands and programs in Matlab

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

- Announcements:
  - Set up folders on your PC, flash drive, or cloud storage to store code for class (see website)
  - See website for office hours and consulting hours
  - See Discussions for partner-finding tips (including WICC social tonight!)
  - First exercise due Sun evening
  - First project will be posted after Tue lecture
Formula

- Surface area of a sphere?

- Have the cosine of some angle $\theta$ in $[0, \pi/2]$ and want $\cos(\theta/2)$?

$$A = 4\pi r^2$$

$$\cos(\theta/2) = \sqrt[2]{\frac{1 + \cos(\theta)}{2}}$$
Interactive computation in *Command Window*

```
>> r = 6
r =
  6
>> a = 4*pi*r^2
a =
   452.3893
>> v = 4/3*pi*r^3
v =
   904.7787
```
% Example 1_1: Surface area of a sphere
% r: radius of the sphere [unit]
% A: surface area of the sphere [unit^2]

r = input('Enter the radius: ');  
A = 4*pi*r^2;  
fprintf('Surface area is %f units^2!\n', A)
A computer program

input → computation → output
Where does computation happen?

- Code lives on a disk (hard drive)
  - Matlab: Folder pane
- Variables live in memory (RAM)
  - Matlab: Workspace pane
Variable & assignment

- **Variable**: a named computer memory space for storing a value

- Valid names start with a letter, can contain digits

- **Use meaningful variable names**!

- Create a variable by assigning a value to it

- By default, a number has the type (class) `double`, for “double precision floating point number”
Variable & assignment

- **Variable**: a named space for storing a value

- **Assignment**: putting a value into a variable

- **Assignment operator**: =

- An assignment statement, e.g., \( r = 2 \times 4.5 \)

- **Expression** on right-hand-side (rhs) is evaluated before the assignment operation

- Update variable’s value with another assignment statement, e.g., \( r = 7 \)
Assignment

- **Expression on rhs** is evaluated before the assignment operation.

- **Examples:**
  
  \[
  \begin{align*}
  x &= 2 \times 3.14 \\
  y &= 1 + x \\
  z &= 4^2 - \cos(y)
  \end{align*}
  \]

- **Question:** can we reverse the order of the 3 statements above?
  
  **NO!** Any variable on the rhs must be initialized.
Assignment

- Expression on rhs is evaluated before the assignment operation

- Examples:
  
  \[
  x = 2 \times 3.14 \\
  y = 1 + x \\
  z = 4^2 - \cos(y)
  \]

- Question: can we reverse the order of the 3 statements above?

- NO! Any variable on the rhs must be initialized.
Matlab’s built-in functions

- Expression on rhs is evaluated before the assignment operation.
- Examples:
  
  \[
  \begin{align*}
  x &= 2 \times 3.14 \\
  y &= 1 + x \\
  z &= 4^2 - \cos(y)
  \end{align*}
  \]

- Question: can we reverse the order of the 3 statements above?
- NO! Any variable on the rhs must be initialized.
Statements in a program are executed in sequence

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

A: 6  B: 7.28  C: some other value  D: error
% Quad1
% Solves $x^2 + 5x + 6 = 0$

```matlab
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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>5</td>
</tr>
<tr>
<td>c</td>
<td>6</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
</tr>
<tr>
<td>r1</td>
<td>-3</td>
</tr>
<tr>
<td>r2</td>
<td>-2</td>
</tr>
</tbody>
</table>
```
% Example 1_1: Surface area of a sphere
% r: radius of the sphere [unit]
% A: surface area of the sphere [unit^2]

r = input('Enter the radius: ');  
A = 4*pi*r^2;  
fprintf('Surface area is %f units^2!\n', A)
Input & output

- \texttt{variable = input('prompt ')}

\begin{verbatim}
> \texttt{r = input('Enter radius: ')}
\end{verbatim}

- \texttt{fprintf('message to print ')}

\begin{verbatim}
> \texttt{fprintf('Increase ')}
fprintf('is %f inches
', x)
fprintf('Position (\%d,\%d)\n', x, y)
\end{verbatim}
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

During discussion: Found out how to control the number of decimal places shown with %e
% Example 1_1: Surface area of a sphere
% r: radius of the sphere [unit]
% A: surface area of the sphere [unit^2]

r = input('Enter the radius: ');
A = 4*pi*r^2;
fprintf('Surface area is %f
', A)
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
  - Units, assumptions/constraints
- Top a block of code for a specific task with a **concise** comment
  - Comment: "What we are trying to do"
  - Code: "How we are doing it"
Example

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

Note: 1 mile = 5280 feet
% Example 1_2: Print surface area increase in miles^2 given an increase in the radius

r= input('Enter radius r in miles: ');
delta= input('Enter delta r in inches: ');

1 mile = 5280 feet
Tips for writing a program

- Check that you know what is given (or is input, or is assumed)

- Be goal-oriented: start by writing the last statement(s) for the program output
  - What is the program supposed to produce? You know this from the problem statement
  - Allows you to work backwards from the results

- Name as a variable what you don’t know
  - Helps you break down the steps
  - Allows you to temporarily skip over any part that you don’t know yet how to do
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...