Concepts in programming expounded via MATLAB

Dan Amrhein – amrhein@mit.edu
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- **if** statements
- **while** loops
- **for** loops

Write functions and scripts

Figure making
What is MATLAB?

- A very powerful calculator
  - Like a scientific calculator, MATLAB has all mathematic and linear algebra tools as built-in functions

- A useful tool for manipulating data
  - MATLAB can read in data and handle very large datasets
  - Most statistical and signal processing tools are already built into MATLAB

- A programming language
  - Like Python it is an interpreted language
  - Executes commands line-by-line
The MATLAB interface
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- if statements
- while loops
- for loops

Write functions and scripts

Figure making
Key concepts: variable, path, function

A variable is an object stored in the computer’s memory.

Attributes:
• Identifier
• Type
• Value
• Size

Typical variable types used in matlab are double, single, char, and logical

These variables are built into MATLAB:

▶ i and j: complex numbers
▶ pi: 3.141592653589793…
▶ ans: stores the last unassigned value (like on a calculator)
▶ Inf and -Inf: positive and negative infinity
▶ NaN: not a number
"The search path, or *path* is a subset of all the folders in the file system. MATLAB software uses the search path to efficiently locate files used with MathWorks products. MATLAB can access all files in the folders on the search path."

to see the current path, type "path"
In general:

\[ \text{[outputs]} = \text{function(inputs)} \]
Can always look up functions in the command window

```
>> help min
min   Smallest component.
For vectors, min(X) is the smallest element in X. For matrices, min(X) is a row vector containing the minimum element from each column. For N-D arrays, min(X) operates along the first non-singleton dimension.

[Y,I] = min(X) returns the indices of the minimum values in vector I. If the values along the first non-singleton dimension contain more than one minimal element, the index of the first one is returned.

min(X,Y) returns an array the same size as X and Y with the smallest elements taken from X or Y. Either one can be a scalar.

[Y,I] = min(X,[],DIM) operates along the dimension DIM.
```
If running a function (or script) causes your computer to freeze, takes too long, or you just realize you don’t want it to run, you can stop it with:

Ctrl+C (Windows/Linux)
Command + Period (Macs)
Built-in MATLAB functions (many more exist!)

- sin(x)
- cos(x)
- tan(x)
- exp(x)
- sinh(x)
- cosh(x)
- tanh(x)
- log(x)
- [Y,I] = min(x,[],dim)
- [Y,I] = max(x,[],dim)
- + (addition)
- - (subtraction)
- * (scalar/matrix multiplication)
- .* (array multiplication)
- ^ (scalar/matrix exponent)
- .^ (array exponent)
- \ (left division)
- / (right division)
- ./ or .\ (array division)
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
  - if statements
  - while loops
  - for loops

Write functions and scripts

Figure making
You can create matrices in three ways

```
[ ]   a = [1, 2]   Commas separate row elements.

:      a = 1 : 3    Semicolons separate column elements.

functions a = rand(5)
```
You can create matrices in three ways

- \[
\begin{bmatrix}
\end{bmatrix}
\]
  \[a = [1, 2]\]
  Commas separate row elements.
  Semicolons separate column elements.

- \[a = 1 : 3\]

- Functions
  \[a = \text{rand}(5)\]

Questions?
Matrix math - come back tomorrow!

+ addition
- subtraction
* scalar/matrix multiplication
^ scalar/matrix exponent
/ right division
\ left division
.* element-wise multiplication
./ element-wise array division
.^ element-wise exponent
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- if statements
- while loops
- for loops

Write functions and scripts

Figure making
Accessing matrix elements

\[
A = \begin{pmatrix}
8 & 1 & 6 \\
3 & 5 & 7 \\
4 & 9 & 2
\end{pmatrix}
\]

A single element

\[
A(3,2)
\]

\[
\begin{array}{c}
9
\end{array}
\]

All of the rows in the first column

\[
A(:,1)
\]

\[
\begin{array}{c}
8 \\
3 \\
4
\end{array}
\]

The entire matrix, strung out columnwise as a vector

\[
A(:)
\]

\[
\begin{array}{c}
9 \\
4 \\
1 \\
6 \\
7 \\
2
\end{array}
\]

Matrices can be indexed as though they are vectors!

\[
A(6)
\]

\[
\begin{array}{c}
9
\end{array}
\]

Part of the second row

\[
A(2,[1,2])
\]

\[
\begin{array}{c}
3 \\
5
\end{array}
\]

The 2x2 submatrix in the upper right

\[
A(1:2,2:end)
\]

\[
\begin{array}{cc}
1 & 6 \\
5 & 7
\end{array}
\]
Questions?
Relational operators

For comparing numbers
<    less than
>    greater than
\leq    less than or equal
\geq    greater than or equal
==    equal
\neq    not equal

'=' is not a relational operator! It is used for variable assignment.

Scalars related by these operators yield logical variables (true or false).

Matrices can be related this way too, yielding logical matrices.

Logical operations
\sim    not
&    and
|    or
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- if statements
- while loops
- for loops

Write functions and scripts

Figure making
Relational operators

For comparing numbers
<  less than
>  greater than
<= less than or equal
>= greater than or equal
== equal
~= not equal

'=' is not a relational operator! It is used for variable assignment.

Scalars related by these operators yield logical variables (true or false).

Matrices can be related this way too, yielding logical matrices.

Logical operations
~  not
&  and
|  or

Questions?
if statements

if <logical variable>
    <some sort of operation>
end

if this variable is TRUE...
then this action is performed!

tab here is not required but improves readability
(MATLAB does this automatically!)
Outline of this class

- Intro MATLAB and its interface
- Key concepts: variable, path, function
- Construct matrices
- Access and modify matrix elements
- Construct conditional statements
- Write code using
  - if statements
  - while loops
  - for loops
- Write functions and scripts
- Figure making
if statements

if <logical variable>
    <some sort of operation>
else
    <some sort of operation>
end

if this variable is FALSE...

then this action is performed!
if statements

if <logical variable>
  <some sort of operation>
elseif <another logical variable>
  <some sort of operation>
end

if this variable is FALSE...
...but this variable is TRUE...
...then this action is performed!
if <logical variable>
  <some sort of operation>
elseif <another logical variable>
  <some sort of operation>
else
  <yet another operation>
end

if this variable is FALSE...

...and so is this variable...

then this action is performed!
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- if statements
- while loops
- for loops

Write functions and scripts

Figure making
A while loop is just a repeated if statement!

```
while <logical variable>
  <some operation>
end
```

---

- `while <logical variable>` keeps evaluating the operation so long as this variable is true!

- Tab for style points (again, MATLAB does this automatically)
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
  - if statements
  - while loops
  - for loops

Write functions and scripts

Figure making
for loops repeat a block of code, and the loop “knows” which iteration it is on!

```plaintext
for ii = <some vector>
    <some statements>
end
```

in the statements, the value of the variable "ii" in the nth repetition is equal to the nth element of <some vector>.

statements are repeated a number of times equal to the length of <some vector>.
Outline of this class

Intro MATLAB and its interface

Key concepts: variable, path, function

Construct matrices

Access and modify matrix elements

Construct conditional statements

Write code using
- if statements
- while loops
- for loops

Write functions and scripts

Figure making
Scripts

The command window is good for scratch work or work you do not need saved.

If you want to save a series of commands, variables, or other functions/jobs, make a script!

% comments out a line in MATLAB
Writing your own functions

The basic syntax: save a *.m file with name foobar.m

First line must be

function  [<outputs>]  =  foobar(<inputs>)
Many different options:
• plot
• bar
• scatter
• image, imagesc, pcolor
• surf
• contour, contourf
• etc
When you get stuck

- Google!
- In the command window: “help <function>”
- Ask a friend!