Computer Programming in MATLAB/OCTAVE

Chaiwoot Boonyasiriwat

March 17, 2016

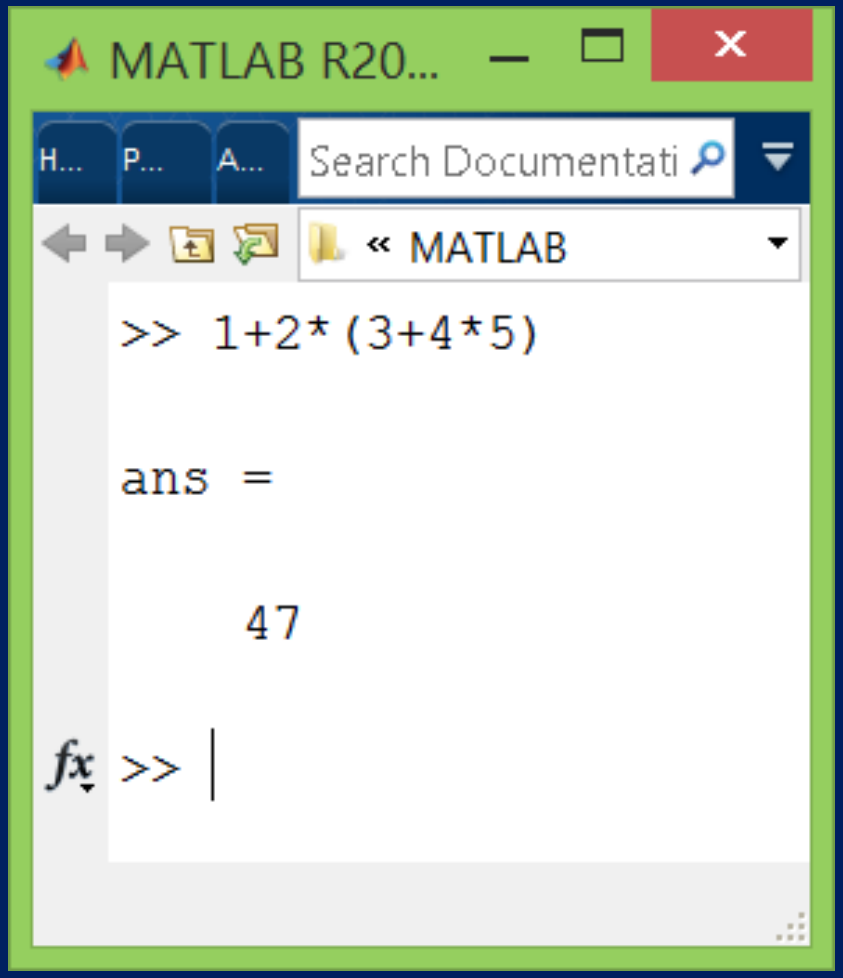
Where to Get the Software?

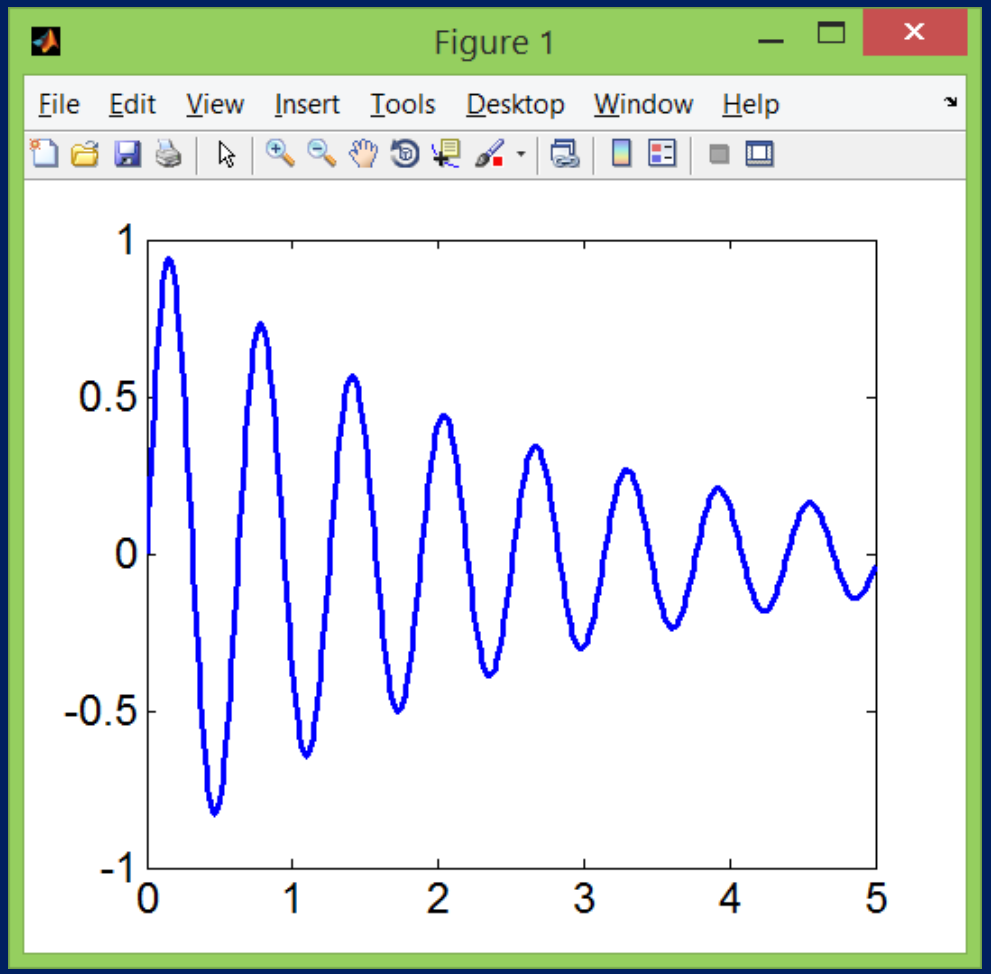
MATLAB Student ($29 or about 1000 bahts) can be purchased at http://www.mathworks.com/academia/student_version/?requestedDomain=www.mathworks.com

Octave can be freely downloaded at http://octave.sourceforge.net/

MATLAB R2014a

You can use MATLAB as a simple calculator and graph plotter.





MATLAB

- MATLAB stands for MATrix LABoratory.
- MATLAB has its own a programming language which is case sensitive and row-major.
- Incorrectly written MATLAB codes can run much slower, e.g. 10x, than the correct ones.

Code for Plotting Previous Graph

plot_graph.m (program script has .m extension)

```
clear;clc
x = 0:0.01:5;
y = sin(10*x);
z = exp(-0.4*x);
plot(x,y.*z,'LineWidth',2)
set(gcf,'Color','white');
set(gca,'FontSize',15);
```

Variables and Matrix (Tensor)

- You can use a variable without declaration.
- Variables with numerical values are of type double precision by default.

Examples:

```
x = 1;          <-> x = 1.0
y = 1:5;        <-> y = [1,2,3,4,5]
z = 1:2:10;     <-> z = [1,3,5,7,9]

A = [1,2;3,4];  ⇨  A = [1 2; 3 4]
I = eye(2,2);
Z = zeros(2,3); a = ones(3,1);
```

Element-by-Element Ops.

- Element-by-element operations are usually needed.
- These operations include multiplication, division, exponential.

Examples:

```
x = a.*b;
y = a./b;
z = 2.^[1:5];
```

Script and Function

- Both Scripts and Functions are sequence of Matlab statements in file.
- Variables defined in Scripts reside in the Workspace while data defined in Functions are only temporary variables which are not in the workspace.

my_script.m

```
a = 1;
b = 2;
```

my_function.m

```
function my_function
a = 1;
b = 2;
```

Function

- Function name must be used as the file name.
- Functions normally cannot access variables defined in the workspace.
- In MATLAB, variables are passed as argument by value to a function.
- Built-in variables `nargin`, `nargout` are the number of input/output arguments of function.
- Example:

```
function c = add(a,b)
c = a+b;
```

Program Control

- Conditional control: `if`, `switch`
- Loop control: `for`, `while`, `continue`, `break`
- Error control: `try`, `catch`
- Program/function termination: `return`

if elseif else

```
a = 5;
if a < 0
   b = 1;
elseif a >= 0 && a < 10
   b = 2;
else
   b = 3;
end
disp(['b = ' num2str(b)]);
```

switch case

```
a = 1;
switch a
   case 1
     b = 1;
   case 2
     b = 2;
   otherwise
     b = 3;
end
disp(['b = ' num2str(b)]);
```

for loop

```
n = 10;
x = zeros(n,1);
y = x;
for i=1:n
   x(i) = i;
   y(i) = x(i)^2;
end
plot(x,y);
```

while loop

```
n = 10;
x = zeros(n,1);
y = x; i = 1;
while i < n
   x(i) = i;
   y(i) = x(i)^2;
   i = i+1;
end
plot(x,y);
```

continue, break

```
a = [0 1 1 2 3 4];
b = a(1);
for i=1:length(a)
   if b == a(i)
      continue;
   elseif b == 3
      break;
   else
      b = a(i);
   end
end
```

try, catch

```
a = [1 2]; b = [3;4];
try
   c = [a b];
catch exception
   disp(['Error type: ' exception.identifier]);
   disp(['Error message: ' exception.message]);
end;
```

return

```
a = 0;
for i=1:10000
   a = a + i;
   if a > 100
      return
   end
end
```