Mini-Matlab Lesson 1: Getting started

Matlab is a powerful tool for numerical computation. As its name (matrix laboratory) suggests, Matlab is designed to make it easy for you to handle and visualize matrices.

Contents

What should I use?

- **Low level programming** (e.g. C and Fortran) is computationally fast and powerful...but has steep learning curves and can be difficult to program.
- **High level programming** (e.g. Matlab, Octave, Python) is not as computationally efficient in general, but is much easier to learn and to program. A well written and efficient Matlab code can still compete quite well with Fortran or C.
- **Computer algebra systems (CAS)** (e.g. Mathematica and Maple) are designed for symbolic manipulation of mathematics (exact and/or high precision). They are much slower than numerical computations, but serve a different purpose...

Getting Started

```
% Princeton students can download a copy of Matlab here:
% <http://www.princeton.edu/software/licenses/software/matlab>
```

Basic creation and display of matrices

```
% Create a row vector
A = [1 2 3];

% You can either display output by (i) not using a semi-colon at the end of
% each command, (ii) typing the name of the variable in a new prompt and
% pressing enter, or (iii) using the disp function
disp('The matrix A = ');
disp(A)

% Create a column vector
B = [5; 6; 7];
disp('The matrix B = ');
disp(A)

% By not using a semi-colon at the end, you can output the value of the
% variable immediately. Here is a 3 x 3 matrix
C = [1 2 3;
    4 5 6;
    7 8 9]
```

```
The matrix A =
```

```
     1     2     3

The matrix B =
     1     2     3


C =

     1     2     3
     4     5     6
     7     8     9
```

Automated creation of matrices

```
% Row vector from 0 to 2 in steps of 0.25
a1 = [0:0.25:2]

% Row vector from 0 to 10 with 7 points
a2 = linspace(0, 10, 7)

% A 2x3 matrix of ones
a3 = ones(2,3)

% A 3x3 identity matrix
a4 = eye(3,3)

% A 2x3 matrix of random numbers
a5 = rand(2,3)
```

```
a1 =

         0    0.2500    0.5000    0.7500    1.0000    1.2500    1.5000    1.7500


a2 =

         0    1.6667    3.3333    5.0000    6.6667    8.3333   10.0000


a3 =

     1     1     1
     1     1     1


a4 =

     1     0     0
     0     1     0
     0     0     1


a5 =
```

```
    0.4904    0.8739    0.2085
    0.8530    0.2703    0.5650
```

Basic matrix manipulations

```
% All the basic matrix operations you learned in linear algebra are easy to
% use in Matlab.

% Determinant
a1 = det([1 2;
          3 4])

% Matrix multiplication
a2 = [1 2 3]*[4; 5; 6]

% Element-wise multiplication
a3 = 2*ones(2,2).*[1 2; 3 4]

% Notice the dot above. The dot operator means 'do things element-wise'.
% Notice that if we tried regular matrix multiplication for two
% incompatible matrices (e.g. [1 2 3]*[4 5 6]) it would not work.

% But pairwise is okay
a5 = [1 2 3].*[4 5 6]

% Regular transpose (_note the dot!_)
a6 = [1+1i 2+1i 3+1i].'

% Conjugate transpose
a7 = [1+1i 2+1i 3+1i]'
```

```
a1 =

    -2


a2 =

    32


a3 =

     2     4
     6     8


a5 =

     4    10    18
```

```
a6 =

   1.0000 + 1.0000i
   2.0000 + 1.0000i
   3.0000 + 1.0000i


a7 =

   1.0000 - 1.0000i
   2.0000 - 1.0000i
   3.0000 - 1.0000i
```

Published with MATLAB® 7.12