Fall 2005 Math 151
Hand/MATLAB Nomenclature
and the “The Cycle of Life”
Tue, 30/Aug ©2005, Art Belmonte

Hand Examples

Nomenclature

References to problems in our textbook [Stewart] done “by hand” have the form **ppp/nn**; i.e., page / exercise #. For example, we refer to the 10th problem on page 53 as 53/10.

In our class, we interpret the phrase “by hand” according to the *Rules of Engagement*. Sometimes this will mean to just use pencil-and-paper. At other times, a TI-89 calculator together with my TAMUCALC package may be employed. (This often speeds up computations considerably but does not show intermediate steps.) You ought to practice doing a given problem one way or the other—or both ways. In the fullness of time, you’ll get the feel of things.

MATLAB Examples

Nomenclature

References to problems in our textbook [Stewart] done with MATLAB have the form **spppxnn**. Those from from your lab manual [Gilat] have the form **gpppxnn**. For example, the directory (folder) that I would use to store files associated with Page 53, Exercise 4 [in Stewart] is **s053x04**.

This naming convention was chosen so as to have fixed-length directory [folder] names that work on all platforms (Linux, Mac OS X, and Windows).

Here are some types of MATLAB files you’ll encounter. All except *.fig* files may be printed via **xprint** from your CalcLab account.

.eps	Encapsulated PostScript file
.fig	figure file
.m	M-file (function or script)
.txt	diary file

“The Cycle of Life” demonstrated [an examination of s053x04]

Find a vector **a** with representation given by the directed line segment $\overrightarrow{AB}$ connecting points $A(4, -1)$ and $B(1, 2)$. Draw $\overrightarrow{AB}$ and the equivalent representation that starts at the origin.

Solution

We have $\mathbf{a} = \overrightarrow{AB} = \overrightarrow{B} - \overrightarrow{A} = [1, 2] - [4, -1] = [-3, 3]$. Ultimately we’ll list a diary file and produce a diagram. But let’s first see how we organize our work in MATLAB.

Supplemental weekend MATLAB sessions at the beginning of the term will introduce you to the “Cycle of Life,” in which you iteratively flesh out the solution to a given problem worked via MATLAB. We start off with a skeleton file in the MATLAB Editor that looks like this. There is a preliminary part, a body, and a closing part.

```
%
delete s053x04.txt; diary s053x04.txt
clear; clc; close all; echo on
%
% Stewart 53/4
%
% COMPUTATIONS GO HERE!
%
echo off; diary off
```

In the preliminary part, clear the deck for the next run of the code.

- Throw away the old diary file (**delete**).
- Start a new diary file (**diary**). Its root name matches that of the M-file.
- Free up all MATLAB variables (**clear**).
- Clear MATLAB’s Command Window (**clc**).
- Close all figure windows (**close all**).
- Cause input to be included in the diary file (**echo on**).

Next come the mathematical computations and graphics commands (if any) in the body.

Finally, we shut down the party.

- Turn off the echoing of input (**echo off**).
- Write out the diary file (**diary off**).

Every problem has the same structure. Only the computations change. Working out of the MATLAB Editor window saves us work. MATLAB cleans up our mess (and our mistakes!) after each run. When we have finished solving the problem, we have a perfect transcript of our work that we may print out.

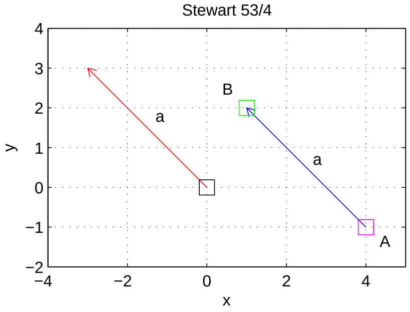
Here is a finished M-file. I didn’t type this in all at once. Instead, I did it interactively bit-by-bit: Edit, save-and-run, fix errors, repeat.

```
%
delete s053x04.txt; diary s053x04.txt
clear; clc; close all; echo on
%
% Stewart 53/4     [Percent signs begin comments.]
%
% Both semicolons and commas separate commands.
% Semicolons suppress output, commas show it.
%
origin = [0 0]; A = [4,-1]; B = [1,2];
a = B - A
arrow(A, a); hold on; axis equal
arrow(origin, a, ’r’); grid on
```

```
axis([-4 5 -2 4])
plot(A(1), A(2), ’ms’, ’MarkerSize’, 12);
plot(B(1), B(2), ’gs’, ’MarkerSize’, 12)
plot(0, 0, ’ks’, ’MarkerSize’, 12)
xlabel(’x’); ylabel(’y’)
title(’Stewart 53/4’)
% Points and vectors labeled interactively via text boxes
%
echo off; diary off
```

And here is the diary file with input and output interspersed. The **arrow** command was written by Professor Jeffery Cooper.

```
% Stewart 53/4
%
origin = [0 0]; A = [4,-1]; B = [1,2];
a = B - A
a =
    -3     3
arrow(A, a); hold on; axis equal
arrow(origin, a, ’r’); grid on
axis([-4 5 -2 4])
plot(A(1), A(2), ’ms’, ’MarkerSize’, 12);
plot(B(1), B(2), ’gs’, ’MarkerSize’, 12)
plot(0, 0, ’ks’, ’MarkerSize’, 12)
xlabel(’x’); ylabel(’y’)
title(’Stewart 53/4’)
% Points and vectors labeled interactively via text boxes
%
echo off; diary off
```



Math 151 MATLAB Environment (on CalcLab)

Here is how to set up your MATLAB environment on CalcLab so as to automatically have access to additional MATLAB routines written by Belmonte, Cooper, and Polking.

Open a terminal window by clicking once on the shell/monitor button near the lower left of the screen. Type the following at a CalcLab terminal window command prompt. That’s the tilde (˜) character.

```
cp  -r  ˜belmonte/matlab       ˜
cp  -r  ˜belmonte/EngrMath  ˜
```

You are recursively copying two directory structures from my home directory to yours. Once you have done this you’re set for the semester on CalcLab!

Auxiliary routines

These MATLAB codes were written by Art Belmonte, with the exception of **arrow**, written by Professor Jeffery Cooper at the University of Maryland. To learn how to use a given routine, type **“help command”** in the MATLAB Command Window, where **command** is the name of one of the routines. This gives a short description of its use. To actually see the underlying MATLAB code, type **“type command”** instead. (Naturally, you do NOT type the quotation marks!)

angvecdg	angle between vectors in degrees
angvecrd	angle between vectors in radians
arrow	plots a 2-D arrow (vector / directed line segment) [Cooper]
cline2pt	Cartesian equation of the line through 2 points
comp	scalar projection of two vectors
d2r	converts degrees to radians
idiff	implicit derivative
len	Euclidean length [magnitude] of a vector
linptdir	parametric equation of line in given direction
orthvec	orthogonal projection of two vectors
perp	orthogonal complement of a 2-D vector
proj	vector projection of two vectors
r2d	converts radians to degrees
unitvec	unit vector parallel to a given vector

Other useful commands

These were gleaned from other lab manuals over the years.

diary	saves the text of a MATLAB session.
echo	displays commands (ordinarily suppressed).
eval	causes execution of a string; used with **vectorize**.
vectorize	renders element-by-element mulitiplication, division, and exponentiation. (You’ll never forget a dot again!)