Lec04 IAM550 J.Raeder 9/5/2019 MATLAB scripts, more plotting

Some notes on file systems:

- Disks, USB drives etc. are block devices: 512b blocks. They need a 'file system' to work, i.e., 'formatting, partitioning'
- MS/Windows → FAT (8.3 file names), FAT32, NTFS (Windows 10)
- Mac → HFS, HFS+ but supports FAT, FAT32 USB drives
- Linux → ext3, ext4, xfs, and basically everything else
- File systems are hierarchical → tree starting from root '/' in Unix, 'drive numbers' a: etc in Windows.
- Buffering → data get lost in a crash, a 'journaled' file system prevents that
- Disks fail! → make frequent backups!
- RAID → Redundant Array of Independent Disks → adds redundancy vs space → raid1 (mirroring); raid5, raid6 (error correction codes)

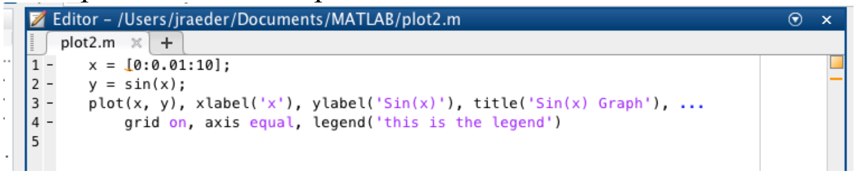
Some notes on file names:

- Old DOS/Windows days: 8.3 file names
- Can contain basically any character (even some non printable ones), but better stick to letters, digits, "_-." And no blanks.
- Convention: name.extension, where the extension designates the file format. This is not enforced! .pdf → Adobe Portable Document Format, .docx → MS/Word, .jpg or .jpeg → portable graphics format, .htm, .html → hypertext markup language, .txt → unformatted (ascii) text, etc.
- Better: file headers, 'magic number' (every postscript file must start with '%!PS', etc.
- MATLAB uses .m for scripts
- Access rights: r/w/x for world/group/user. More sophisticated (secure Linux only): Access Control Lists (ACL)
- Tips: make file names as descriptive as possible, BUT do not use spaces in file names (not forbidden, but they can break command line or scripts).

MATLAB scripts (also M-files):

- Avoid repeated typing
- The only way to do really complex calculations
- Self-documenting (more or less)
- Repeatable!! Try to repeat exactly what you did with an interactive drawing program ☹
- Re-usable, no reinventing wheels.
- Use old scripts (or examples from the Internet) to create new ones

- In the command window type → edit *filename.m* → opens new edit window, filename may already exist
- OR from menu New → Script OR → Open *filename*
- Pops up a new window → simple ascii editor
- You can as well edit m-files with a text editor, but NO FORMATTING → notepad, vi, emacs, xedit, but NOT MS/WORD.
- The script is just MATLAB commands as you would type them. Be careful to use ';' at command end. Otherwise you may get too much output when you execute it.
- When done → Save OR Save as
- To run the script → menu → Run
- OR, after you saved the script (or if the m-file already exists) just type the file name in the command window.
- M-files can have 1000s of lines
- When writing scripts, always include plenty of comments (% Comment)
- A simple one to make a plot:



```
x = [0:0.01:10];
y = sin(x);
plot(x, y), xlabel('x'), ylabel('Sin(x)'), title('Sin(x) Graph'), ...
    grid on, axis equal, legend('this is the legend')
```

MATLAB advanced line plots:

Look at: https://www.mathworks.com/help/matlab/ref/plot.html#btzi1ef-2

Multiple plots with annotations:

```
ax1 = subplot(2,1,1); % top subplot
x = linspace(0,3);
y1 = sin(5*x);
plot(ax1,x,y1)
title(ax1,'Top Subplot')
ylabel(ax1,'sin(5x)')

ax2 = subplot(2,1,2); % bottom subplot
y2 = sin(15*x);
plot(ax2,x,y2)
title(ax2,'Bottom Subplot')
ylabel(ax2,'sin(15x)')
%  immediately save the figure as a pdf file
print(gcf, 'Fig.pdf', '-dpdf', '-fillpage')
%   'gcf'  means 'get current figure'
```

Complex figure:

```
% Create a set of values for the damping factor
zeta = [0.01 .02 0.05 0.1 .2 .5 1];
% Define a color for each damping factor
colors = ['r' 'g' 'b' 'c' 'm' 'y' 'k'];
% Create a range of frequency values equally spaced logarithmically
w = logspace(-1, 1, 1000);
% Plot the gain vs. frequency for each of the seven damping factors
figure
for i = 1:7
    a = w.^2 - 1;
    b = 2*w*zeta(i);
    gain = sqrt(1./(a.^2 + b.^2));
    loglog(w, gain, 'color', colors(i))
    hold on
end
% Set the axis limits
axis([0.1 10 0.01 100])
% Add a title and axis labels
title('|G|(\omega) vs \omega')
xlabel('\omega')
ylabel('|G|(\omega)')
% Turn the grid on
grid on
```