CS319: Scientific Computing (with MATLAB)

CS319 Lab 3: Writing MATLAB Live Scripts

Week 6 , 2023

Niall Madden (Niall.Madden@UniversityOfGalway.ie)

Goal

Get experience of writing a MATLAB Live Script, where you combine code, functions and examples. This will include use of advanced features.

Submit your Live script on Blackboard (Labs... Lab 3) by 17:00, Friday 17 Feb. As ever, collaboration is encouraged, but must be acknowledged.

1. First steps

1. In MATLAB, create a new Live Script.
2. In the `LIVE EDITOR` menu, click on `Text`, and type a suitable title for your document. Change this from `Normal` to `Title` (see menu above the text formatting buttons.
3. In the `INSERT` menu, click on `Table of Contents`, then click on `Section Break`.
4. Give a title to this section, such as "Introduction". Do this by typing the text of the title, and then convert that text to `Heading 1` (same menu as for `Title`. Notice that this title now appears in the table of contents.
5. Add some formatted text to this section, including your name, ID number and email address. Add these as bullet points. Put your name in **bold**, your ID number in *italics*, and your email address as `monospaced`.
6. Also add bullet points with a hyperlink to some website, and one with your favourite equation. The latter can be accessed in the `Equation Menu`. However, if you know LaTeX, just type
   `$a^n + b^n = c^n$`

2. Editing Figures

Insert another section break, and add a title for the second section.

(Note: section breaks are important in Live Scripts, since you can run all the code in a single section at once).

In Week 5, (starting Slide 24), we looked at approximating a function by Taylor polynomials of various degrees.

Review the `TaylorPoly.m` script for plotting Taylor Polynomials.

Create a box for inserting code (by clicking on `Code` in the `LIVE EDITOR` menu. Copy all the code from the `TaylorPoly.m` script.

Check that it works, by clicking `Run Section`.

Tip: Don't copy the code from the lecture slides: it is likely that some of the characters won't copy properly.

Now click on one of the figures created. Using tools in the `FIGURE` menu, add a title, a grid, and x- and y-labels.

3. Data fitting

In this example, we'll compute a linear (for now) fit to some noisy data. A simple example is given in the script NoisyData.m, which you can download from Bitbucket at https://bitbucket.org/niallmadden/2223-cs319/src/main/lab3/NoisyFit.m

NoisyFit.m

```
e = 0.1; % magnitude of the noise
d = 1; % degree of the polynomial fit.
f = @(x) 1+2*x - 3.^x^2;

x = linspace(0,1,100);
y = f(x) + e*randn(size(x));
p = polyfit(x , y , d) ;

plot(x,y, '.', x, polyval(p, x), 'LineWidth',3)
```

Create a new section in your Live Script, with a suitable title. Don't forget the **section break** – it is *especially important here*. Add the code, and check that it runs OK.

3. Data fitting

That code computes the polynomial least squares fit, $p(x)$, to the vector

$$y = f(x) + en(x),$$

on the interval $[0, 1]$, and where e a chosen scalar (`0.1`, by default) and n is vector, if the same size as x, of normally distributed random numbers ($\mu = 0$, $\sigma - 1$). The degree of p is $d = 1$, by default (i.e., linear).

It then plots y and $p(x)$ against x.

Experiment with this, by trying different values of e and d.

4. Adding controls

In a MATLAB Live script, you can add "controls" such as slides, drop-down menus, etc, to variables or expressions. This allows the a user to vary parameters in your script, in an intuitive fashion.

To see how this works, select the text0.1 in the line `e=0.1`. Now click on `Control...Add slider`. Now right-click on the slider, and select `Configure Control`. Set

- the `min` value to `0`
- the `max` value to something positive, such as `1`
- the `step` to something small, such as `0.05`.

Now every time you adjust the slider:

- a new value is given to e
- all the code in that section is re-run.

Try adding other controls, such as:

- a "spinner" for the choice of d;
- a "drop-down" menu for the choice of f.