Assignment 3 **2.086 Fall 2014**

Due: *Monday, 3 November at 5 PM.*

Upload your solution to course website as a zip file "`YOURNAME_ASSIGNMENT_3`" which includes the script for each question *as well as* all MATLAB® functions (of your own creation) called by your scripts; both scripts and functions must conform to the formats described in **Instructions** and **Questions** below.

Instructions

Download (from the course website Assignment 3 page) the `Assignment_3_Templates` folder. This folder contains a template for the script associated with each question (`A3Qy_Template` for Question `y`), as well as a template for each function which we ask you to create (`func_Template` for a function `func`). The `Assignment_3_Templates` folder also contains the `grade_o_matic` files for Assignment 3 (please see Assignment 1 for a description of `grade_o_matic`[1]) as well as all `.mat` files which you will need for Assignment 3.

We indicate here several general format and performance requirements:

(a.) Your script for Question `y` of Assignment `x` *must* be a proper MATLAB ".m" script file and *must* be named `AxQy.m`. In some cases the script will be trivial and you may submit the template "as is" — just remove the `_Template` — in your `YOURNAME_ASSIGNMENT_3` folder. But note that you still must submit a proper `AxQy.m` script or `grade_o_matic` will not perform correctly.

(b.) In this assignment, for each question `y`, we will specify inputs and outputs both for the script `A3Qy` and (as is more traditional) any requested MATLAB functions; we shall denote the former as script inputs and script outputs and the latter as function inputs and function outputs. For each question and hence each script, and also each function, we will identify *allowable instances* for the inputs — the parameter values or "parameter domains" for which the codes must work.

(c.) Recall that for scripts, input variables must be assigned *outside* your script (of course before the script is executed) — *not* inside your script — in the workspace; all other variables required by the script must be defined *inside* the script. Hence you should test your scripts in the following fashion: `clear` the workspace; assign the input variables in the workspace; run your script. Note for MATLAB functions you need not take such precautions: all inputs and outputs are passed through the input and output argument lists; a function enjoys a private workspace.

(d.) We ask that in the submitted version of your scripts and functions you suppress all display by placing a "`;`" at the end of each line of code. (Of course during debugging you will often choose to display many intermediate and final results.) We also require that **before** you upload your solution to course website you run `grade_o_matic` (from your YOURNAME_ASSIGNMENT_3 folder) for final confirmation that all is in order.

[1]Note that, for display in verbose mode, `grade_o_matic` will "unroll" arrays and present as a row vector.

Note that, in Assignment 3, the templates now provide rather little in the way of hints: you must largely design your own code. You can start with the mathematical statement of the problem and, as appropriate, the numerical method for approximation or estimation or solution. You can then design the logic (or "flow") for your code: the reduction of your method to a sequence of steps — an algorithm. Finally, you should consider the particular MATLAB implementation: the capabilities of MATLAB which you will exploit, and the associated syntax and "built-in" functions.

Questions

1. (*20 points*) A robot takes advantage of an IR Distance Transducer to identify the shape of a room and its own position within the room, as demonstrated in the 2.086 OCW video RoboRoom. In order to relate distance to measured voltage the robot exploits a calibration curve deduced offline by a least-squares procedure.

   In this question we would like you to develop a script which, given a set of (distance, voltage) pairs, develops estimates for the calibration constants which inform the calibration relationship. Please refer to the nutshell *Fitting a Model to Data*, Section 4.2.3, for a description of the IR Distance Transducer model and the formulation of the associated *linear* least-squares fit procedure; the former is defined by the two calibration constants C and γ; the latter develops associated estimates for these calibration constants, $\hat{C}$ and $\hat{\gamma}$, respectively.

   The script takes two script inputs, D_i^{meas} and V_i, for $1 \leq i \leq m$: D_i^{meas} is an independently measured distance for which the transducer yields voltage V_i. These inputs D_i^{meas} and V_i, $1 \leq i \leq m$, must correspond in your script to MATLAB $m \times 1$ arrays `D_meas` and `V`, respectively: entry i of `D_meas` (respectively, of `V`) is the distance D_i^{meas} (respectively, voltage V_i) associated with the i^{th} (distance, voltage) calibration pair. Allowable instances must satisfy $1 \leq m \leq 10000$ and furthermore yield a design matrix X with independent columns. (Note that m is not a script input and should instead be deduced from (say) `D_meas`.)

   The script yields two script outputs: $\hat{C}$ and $\hat{\gamma}$, your least-squares estimates for the calibration constants C^{true} and γ^{true}, respectively; $\hat{C}$ and $\hat{\gamma}$ must correspond in your script to MATLAB scalars `C_hat` and `gamma_hat`, respectively.

   A template is provided in `A3Q1_Template`. We emphasize that your script should perform correctly for any set of (real or synthetic) data. You should yourself devise several test cases for which you can anticipate the correct answers and hence test your script.

   Analysis Suggestion: Dr James Penn conducted calibration experiments for a particular transducer, a Sharp GP2Y0A02YK0F (with a 15-150 cm range). The experimental data comprises $m = 15$ measurements — calibration pairs — provided to you in the `.mat` file `IR_Transducer_Data` (available in the `Assignment_3_Templates` folder) as $m \times 1$ arrays `D_meas` and `V`; the latter conform to the script input specifications. For this set of real data, exercise your script `A3Q1` to obtain estimates $\hat{C}$ and $\hat{\gamma}$ for the calibration constants C^{true} and γ^{true}, respectively. Do you believe that the resulting calibration curve will be uniformly accurate over the entire range of distances 15 cm $\leq D \leq$ 150 cm? Do you find any evidence for model error in the assumed form $D^{\text{model}}(V)$ of the calibration curve?

2. (*20 points*) The height of a ball as a function of time is given by $z^{\text{true}}(t)$. We wish to determine the acceleration of the ball, $\ddot{z}(t)$, from noisy measurements. Note that z is measured positive upwards relative to the acceleration of gravity.

In particular, for any given time t_0 of interest, we are provided with $m = 5$ measurements, $(t_{-2}, z_{-2}^{\text{meas}}), (t_{-1}, z_{-1}^{\text{meas}}), (t_0, z_0^{\text{meas}}), (t_1, z_1^{\text{meas}}), (t_2, z_2^{\text{meas}})$. The measurement times $t_i, -2 \leq i \leq 2$, are equispaced:

$$t_i = t_0 + i\,\delta t\ , -2 \leq i \leq 2\ , \tag{1}$$

for a given positive δt.

We consider two approximations to $\ddot{z}(t_0)$, the first based on finite difference, the second based on least-squares fit. For the finite different approximation, we take

$$\ddot{z}_{\delta t}(t_0) = \frac{z_{-1}^{\text{meas}} - 2z_0^{\text{meas}} + z_1^{\text{meas}}}{(\delta t)^2}\ . \tag{2}$$

For the regression approximation, we take

$$\ddot{z}_{\text{fit}}(t_0) = 2\hat{\beta}_2\ , \tag{3}$$

where $\hat{\beta}_2$ is the approximation to β_2^{true} associated with least-squares fit of the data z_i^{meas}, $-2 \leq i \leq 2$, to the quadratic model $z^{\text{model}}(t;\beta) = \beta_0 + \beta_1 t + \beta_2 t^2$.

In this question we would like you to write a function with signature

```
[accel_fd,accel_fit] = accel_calc(time_vec, z_meas_vec)
```

which, for some given set of height measurements,

$$(t_{-2}, z_{-2}^{\text{meas}}), (t_{-1}, z_{-1}^{\text{meas}}), (t_0, z_0^{\text{meas}}), (t_1, z_1^{\text{meas}}), (t_2, z_2^{\text{meas}}),$$

evaluates $\ddot{z}_{\delta t}(t_0)$ of (2) and also $\ddot{z}_{\text{fit}}$ of (3).

The function takes two function inputs: $t_i, -2 \leq i \leq 2$, and $z_i^{\text{meas}}, -2 \leq i \leq 2$, where z_i^{meas} is the measured height (in meters) corresponding to the time t_i (in seconds). These inputs must correspond in your script to MATLAB 5×1 vectors `time_vec` and `z_meas_vec`, respectively: for $k = 1, \ldots, 5$, entry k of `time_vec` (respectively, `z_meas_vec`) is t_{k-3} (respectively, z_{k-3}^{meas}). (Note the shift of indices such that the MATLAB array first index is 1.) Allowable instances must conform to (1), and yield a design matrix X with independent columns. We do not place any requirements on $(t_i, z_i^{\text{meas}}), -2 \leq i \leq 2$, related to the probability distribution of the measurement noise.

The function yields two function outputs: acceleration estimates $\ddot{z}_{\delta t}$ and $\ddot{z}_{\text{fit}}$, which must correspond in your function to MATLAB scalar variables `accel_fd` and `accel_fit`, respectively.

A script template is provided in `A3Q2_Template`: you should not modify this template; you must only remove the `_Template` and upload in your `YOURNAME_ASSIGNMENT_3` folder. We also provide a function template in `accel_calc_Template`. We emphasize that your function should perform correctly for any set of (real or synthetic) data. You should yourself devise several test cases for which you can anticipate the correct answers and hence test your script.

Analysis Suggestion: Apply your function `accel_calc` to various snippets of synthetic data with different magnitudes of noise (which you may take as normal zero-mean, homoscedastic, independent). Which approach, finite-difference or fit, would be most efficient in the case of very small noise? Which approach, finite-difference or fit, might be more accurate in the case of larger noise? (Note that finite-difference approximation is implicitly based on *interpolation* of the data rather than fitting of the data.) How might you improve the fit approach?

In the event that you would like to bolster your argument with experimental data, we provide some ammunition. Dr James Penn conducted falling ball experiments for a ball of mass 0.014 kg and diameter 0.0508 meters: a video camera records the trajectory; an image processing technique, with correction for lens distortion, yields data for height as a function of time. We provide to you in the `.mat` file `Falling_Ball_Snippet` (available in the `Assignment_3_Templates` folder) as 5×1 arrays `time_vec` and `z_meas_vec`. The `time_vec` and `z_meas_vec` arrays conform to (1) and yield a design matrix X with independent columns, and thus are allowable input instances for your function `accel_calc`. Note that in Dr Penn's data, height is defined as distance above the ground — hence positive, and decreasing as a function of time.

3. (*20 points*) As demonstrated in the 2.086 OCW video RoboFriction, the friction coefficient between a robot wheel and the ground plays a crucial role in robot navigation and performance. We would like to (*a*) predict the friction force for the materials relevant to this particular robotic application, and more generally (*b*) confirm (at least for our situation) Amontons' Law for friction force as a function of (only) normal load.

   Towards that end, we shall follow the formulation of **CYAWTP 6** of the nutshell *Fitting a Model to Data* to identify estimates $\hat{C}$, $\hat{\alpha}$, and $\hat{\eta}$ for C^{true}, α^{true}, and η^{true}, respectively, of the power law friction model proposed. We assume that we are privy to m experimental measurements of the form

$$(F_{\text{f, static}}^{\text{max, meas}})_i = (F_{\text{f, static}}^{\text{max}})^{\text{model}}((F_{\text{normal}})_i, (A_{\text{surface}})_i; \beta^{\text{true}}) + \epsilon_i \ , \ 1 \le i \le m \ . \tag{4}$$

   Here $((F_{\text{normal}})_i, (A_{\text{surface}})_i)$ is the (normal force, superficial surface area) pair associated with the i^{th} measurement of the maximum static friction force, $(F_{\text{f, static}}^{\text{max, meas}})_i$.

   More specifically, we would like you to write a script which, for some given set of data, performs a regression to determine (*i*) an estimate $\hat{C}$ for the friction coefficient μ, (*ii*) a 95% confidence-level individual two-sided confidence interval $[\text{ci}_\eta]$ for η^{true}, and (*iii*) a judgement as to whether to accept or reject the hypothesis $\mathbf{H} : \eta = 0$. Note you may develop the confidence interval under the assumptions of zero-mean, homoscedastic, independent normal noise; please consider (not the large-sample but rather) the finite-sample formula as provided in Section 4.2 of the nutshell *Regression*.

   The script takes three script inputs: $(F_{\text{f, static}}^{\text{max, meas}})_i$, $(F_{\text{normal}})_i$, and $(A_{\text{surface}})_i$, for $1 \le i \le m$, where $(F_{\text{f, static}}^{\text{max, meas}})_i$ is the maximum measured static friction force corresponding to the normal load $(F_{\text{normal}})_i$ and the superficial surface area $(A_{\text{surface}})_i$. These inputs must correspond in your script to MATLAB $m \times 1$ vectors `F_fstaticmaxmeas`, `F_normal`, and `A_surface`, respectively: the entry i of `F_fstaticmaxmeas`, `F_normal`, and `A_surface` contains, respectively, the measured friction force $F_{\text{f, static}}^{\text{max, meas}}$ (in Newtons), the prescribed normal load F_{normal} (in Newtons), and the prescribed surface area A_{surface} (in cm^2), for the i^{th} measurement. Allowable instances must satisfy $1 \le m \le 10000$ and furthermore yield a design matrix X with independent columns. (Note that m is not a script input and should instead be deduced from (say) `F_fstaticmaxmeas`.)

   The script yields three script outputs: scalars $\hat{C}$, which must correspond in your script to MATLAB variable `C_hat`; $[\text{ci}_\eta]$, which must correspond in your MATLAB script to the 1×2

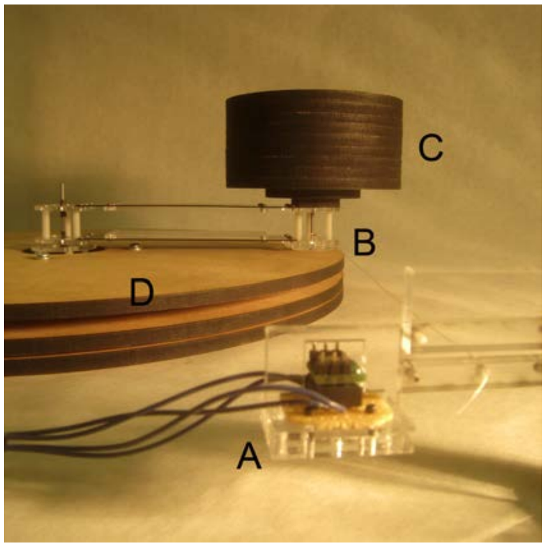

Figure 1: Experimental apparatus for friction measurement: Force transducer (A) is connected to contact area (B) by a thin wire. Normal force is exerted on the contact area by load stack (C). Tangential force is applied on turntable (D) and transmitted by friction from the turntable surface to the contact area. Apparatus and photograph courtesy of James Penn.

array `ci_eta`; logical MATLAB variable `Amontons_Law` which should be assigned the value `True` (respectively, `False`) if you accept (respectively, reject) hypothesis **H**.

A template is provided in `A3Q3_Template`. We emphasize that your script should perform correctly for any set of (real or synthetic) data. You should devise several test cases for which you can anticipate the correct answers and hence test your script.

Analysis Suggestion: Dr James Penn conducted experiments, for a particular pair of robot-wheel and ground materials, to obtain friction force measurements $F_{\mathrm{f,\,static}}^{\mathrm{max,\,meas}}$ (in Newtons) as a function of normal load F_{normal} (in Newtons) and superficial contact area A_{surface} (in cm^2). The turntable apparatus is shown in Figure 1.

The experimental data comprises $m = 50$ measurements: 2 (repetition) measurements at each of 25 points on a 5×5 "grid" in $(F_{\mathrm{normal}}, A_{\mathrm{surface}})$ space. This *real* data is provided to you in the `.mat` file `friction_data` (available in the `Assignment_3_Templates` folder) as 50×1 arrays `F_fstaticmaxmeas`, `F_normal`, and `A_surface`: entry i of `F_fstaticmaxmeas`, `F_normal`, and `A_surface` provides, respectively, the measured friction force $F_{\mathrm{f,\,static}}^{\mathrm{max,\,meas}}$ (in Newtons), the prescribed normal load F_{normal} (in Newtons), and the prescribed surface area A_{surface} (in cm^2), for the i^{th} measurement; for example, in the first measurement, $i = 1$, the measured friction force is 0.1080 Newtons, the imposed normal load is 0.9810 Newtons, and the superficial contact area is 1.2903 cm^2.

Based on Dr Penn's data, did Amontons get it right, say at the 95% confidence level, at least for the particular materials of our 2.086 experiment? In other words, would you conclude, for our pair of materials, that the static friction force is indeed independent of the superficial contact area?

4. *(20 points)* In 2.086, in the infamous Spring of 2013, a quiz — the same quiz — is administered

asynchronously across recitations which take place on the Monday, Wednesday, and Friday of a particular week. We would like to understand if, on average, the students who take the quiz later in the week perform the same as the students who take the quiz earlier in the week. In particular, we do not wish to afford any groups of students an unfair advantage.

We first postulate a model for the average grade received on the quiz as a function of time,

$$g^{\text{model}}(t;\beta) = \beta_0 + \beta_1 t \ , \tag{5}$$

where g represents the average numerical score (out of 100 points), $\beta = (\beta_0 \ \beta_1)^{\mathrm{T}}$ are the coefficients to be determined from data, and t is the time at which the quiz is administered. We shall measure time in days such that the Monday, Wednesday, and Friday of the quiz week correspond to $t = 1$, $t = 3$, and $t = 5$, respectively.

We next assume that the scores for the m students who shall take the quiz can be represented as

$$g_i^{\text{meas}} = g^{\text{model}}(t_i;\beta^{\text{true}}) + \epsilon_i \ , \tag{6}$$

where g_i^{meas} is the grade received by student i, β^{true} is the true value of the parameter vector β in (5),[2], t_i is the time — day 1, 3, or 5 — at which student i takes the quiz, and ϵ_i reflects variations in individual student performance.

In this question we would like you to write a script which, for some given set of student grades, performs a regression to determine (*i*) a 95% confidence-level individual two-sided confidence interval $[\text{ci}_{\beta_1^{\text{true}}}]$ for β_1^{true},the "slope" of our grade model (5), and (*ii*) a judgement as to whether to accept or reject the hypothesis $\mathbf{H} : \beta_1^{\text{true}} = 0$. Note for your confidence interval and your hypothesis test, you may assume that $\epsilon_i, 1 \le i \le m$, is zero-mean normal, homoscedastic, and independent. (Certainly due to the upper limit on the score — 100 — the ϵ_i will not be truly normal. However, our sample will be relatively large, in which case strict adherence to normality, and homoscedasticity, is less crucial. In general, time series can prove problematic, but our window here is very short.) Please consider for your confidence interval (not the large-sample but rather) the finite-sample formula as provided in Section 4.2 of the nutshell *Regression*.

The script takes three script inputs: a $m_1 \times 1$ array of the grades of the students who take the quiz on Monday (hence the g_i^{meas} for all students i for whom $t_i = 1$), which must correspond in your script to MATLAB variable `Mon_grades`, a $m_3 \times 1$ array of the grades of the students who take the quiz on Wednesday (hence the g_i^{meas} for all students i for whom $t_i = 3$), which must correspond in your script to MATLAB variable `Wed_grades`, and a $m_5 \times 1$ array of the grades of all students who take the quiz on Friday (hence the g_i^{meas} for all students i for whom $t_i = 5$), which must correspond to MATLAB variable `Fri_grades`. Allowable instances of each of these three arrays must satisfy $0 \le g_{.} \le 100$ and furthermore $m_1 \ge 1, m_3 \ge 1$, and $m_5 \ge 1$. (Note that m_1, m_3, and m_5 are not inputs: these variables should instead be deduced from `Mon_grades`, `Wed_grades`, and `Fri_grades`, respectively.)

The script yields two outputs: $[\text{ci}_{\beta_1^{\text{true}}}]$, which must correspond in your MATLAB script to the 1×2 array `ci_beta_1`; logical MATLAB variable `no_unfair_advantage` which should be assigned the value `True` (respectively, `False`) if you accept (respectively, reject) the hypothesis $\mathbf{H}$.

[2]We may think of β^{true} as the value of the parameter vector in the hypothetical limit of an infinite number of students.

A template is provided in `A3Q4_Templates`. We emphasize that your script should perform correctly for any set of (real or synthetic) data. You should devise several test cases for which you can anticipate the correct answers and hence test your script.

Analysis Suggestion: We conducted this asynchronous quiz "experiment" in Spring of 2013. The experimental data is summarized in the files `Mon_grades`, `Wed_grades`, and `Fri_grades` in the `.mat` file `Quiz_1_S2013_Grades` (available in the `Assignment_3_Templates` folder); these three files conform to the script input prescriptions for `A3Q4.m`. Based on this data, can we conclude, at the 95% confidence level, that the students who take the quiz later in the week perform the same as the students who take the quiz earlier in the week?

5. *(20 points)* Develop a 1-page essay and a single figure which addresses one of the **Analysis Suggestion** topics posed in Questions 1, 2, 3, and 4. (Note you are required to treat in your essay only *one* of the **Analysis Suggestion** topics; you may choose whichever topic you find most interesting.)

   The essay should be in .pdf format (please, no Word documents) with filename `EssayText.pdf`; the figure should be in MATLAB `.fig` format with name `EssayFigure.fig`. Include both of these files in your folder `YOURNAME_ASSIGNMENT_3` uploaded to course website.

   Note the figure should not be just a pretty picture. The figure, which should be explicitly referenced in your essay, should provide substantive supporting evidence for some central claim or conclusion of your response. Your figure must include axis labels, a title, and a legend so that the reader — 2.086 staff — can comprehend the content.