MATH 235 — Mathematical Models in Science and Engineering

Instructions: Present your work in a neat and organized manner. Please **use** either the 8.5×11 size paper or the filler paper with pre-punched holes. Please do **not use** paper which has been torn from a spiral notebook. Please secure all your papers by using either a staple or a paper clip, but **not** by folding its (upper left) corner.

You **must** show **all of the essential details** of your work to get full credit. If you used *Mathematica* for some of your calculations, attach a printout showing your commands and their output. If I am forced to fill in gaps in your solution by using notrivial (at my discretion) steps, I will also be forced to reduce your score.

Please refer to the syllabus for the instructions on working on homework assignments with other students and on submitting **your own** work.

Homework Assignment # 11

1. This problem is worth **1.5 points**.
   The purpose of this Exercise is twofold. First, you will experiment with a numerical code to verify that the procedure described in Sec. 11.2 of fitting a smooth curve through jagged data indeed works. Second, you will investigate how the "quality" of such smoothing depends on the smoothing criterion (i.e., in this particular case, on which derivative of the curve you minimize).

   (a) Download the code `hw11_p1.m`. Run it for a few different values of the parameter α, whose value is set in or around line 43. Attach three plots which illustrate that in order to obtain a smooth curve that is close to the exact solution, "α must be small but not too small". That is, your plots should show situations where the solution is: (i) close to $\hat{u}$ (and hence noisy), (ii) very smooth but rather far from both $\hat{u}$ and u_{exact}; and (iii) both "sufficiently smooth" and "not too far" from the exact solution u_{exact}. The plots must come either all from Figure 1 or all from Figure 2 (preferred), but there is no need to attach plots of both figures.

   Note that near the end points, the smoothened curve may deviate noticeably from the exact one. This is a consequence of the approximation (19) stated in the notes. You will be given a chance to correct for this in the Bonus problem below.

   Technical note 1: When searching for a value of α in problems 1 and 3, use only one significant digit.

   Technical note 2: A shorthand syntax to enter a number $\#\cdot 10^{-m}$ in Matlab is `#e-m` (e.g., `3e-7`).

   Technical note 3: You should consider putting the value of α in the title of your figure; in this case, use the command `num2str`, as illustrated in the code `hw11_p1.m`.

   (b) In (or near) line 45 of the code, set `der2Bmin=1` and then repeat the steps in part (a). Read the accompanying comments in the code for the explanation of the change you have made.

   (c) This part itself has multiple questions.

   (i) Let the (nearly) optimal values of α that you have found in parts (a) and (b) be denoted by α_2 and α_1, respectively (so that the subscript indicates which derivative is minimized by the given smoothing method). Which of these α's is larger? Qualitatively explain your finding following the analysis of Sec. 11.3.

   Hint: To explain why one of the α's is larger, you will need the information about eigenvalues of matrix M. This matrix is defined inside the code for two values of `der2Bmin`. A useful command for finding just a few eigenvalues of a matrix is `eigs`.

(ii) Keeping `der2Bmin=1` and `alpha=`α_1, run the code. Click on Figure 1 and then in Matlab's command window, type `hold on`. (This will prevent the existing plot from being overwritten by the plot which you will produce at the next step.) Repeat this for Figure 2. Then, set `der2Bmin=2` and `alpha=`α_2 and run the code again. The curves obtained by both smooting methods will be plotted together. Attach a printout of Figure 2, label the curves, and write which method does the smoothing job better.

(iii) As a small variation of the previous question, state whether, in your opinion, one of the methods does a *much* better job than the other.
As a practical matter, suppose that you have to smoothen some uknown curve and are not sure which smoothness criterion to select. Further suppose that you already have a code that smoothens curves using the minimization of their first derivatives. If you decide to change the smoothing method, you will have to go inside the code and reprogram part of it. Will you do that, or just go ahead and try to do the job with the existing code? (Of course, you need to justify your opinion.)

2. This problem is worth **1.5 points**.
In Sec. 11.3, we identified the eigenvectors corresponding to a few *largest* eigenvalues of matrix H as the main culprits causing jaggedness of curve $\hat{u}$. On the other hand, in Sec. 11.4, we showed that the "jaggedness culprits" for the solution of a Fredholm equation of the first type are the eigenvectors of matrix A corresponding to the first few *smallest* eigenvalues. In this Exercise, you will verify both of these statements through numerical computation and then will be asked to explain why in these two cases, the "culprit" eigenvectors come from the opposite sides of the spectrum.

(a) (i) In the code `hw11_p1.m`, set $N = 16$ (line 13) and run the code. Close both figures, but do *not* clear the workspace. This way you will define matrix H in the current workspace.
(ii) Download the code `hw11_p2a.m`. Open it and replace all the occurrences of `??` with proper commands as directed in the comments in that code.
(iii) In Matlab's command window, type `format short e`. In this format, you will be able to view a very small number, e.g., 0.00000000123, which in Matlab's default format (which is `short` as opposed to `short e`) would appear as 0.

Run `hw11_p2a.m` and print out the figures. Label the graph in each panel with the numeric value of the corresponding eigenvalue.[1] A simple but important quantitative measure of an eigenvector's profile is the number of zero crossings. Write your conclusion of how this measure of the eigenvector profile changes with the increase of the corresponding eigenvalues.

Note: Pay attention to the scale along the vertical axis, especially for the eigenvectors corresponding to the smallest eigenvalues.

(b) (i) Download and run the code `hw11_p3.m`. Close both figures, but do *not* clear the workspace. This will define matrix A in the current workspace.
(ii) Save the code `hw11_p2a.m` as `hw11_p2b.m`: click on *File* in the top-left corner of the Matlab <u>editor</u> (*not* command!) window and select *Save as*. Modify the code so as to be able to obtain the eigenvectors and eigenvalues of matrix A and then plot the first and last three of those eigenvectors.
(iii) Repeat Step (iii) of part (a) for `hw11_p2b.m` instead of `hw11_p2a.m`.

(c) Why does the fastest-changing eigenvector of A correspond to its smallest eigenvalue, while the fastest-changing eigenvector of M corresponds to its largest eigenvalue? (I.e., why are the orders of the eigenvectors of these two matrices *opposite*?)

[1] See Technical note 3 in Problem 1(a).

Hint: Which continuous operation was matrix A obtained from? Same question about matrix M with the H given by Eq. (16) of the notes.

3. This problem is worth **1.5 points**. Of these, parts (a)–(f) combined are worth 1 point, and part (g) is worth 0.5 point.

   The purpose of this exercise is to demonstrate that the sensitivity of the regularization of Fredholm integral equations of the first kind to the choice of the smoothing matrix B is greater than it is for the curve-smoothing problem considered in Problems 1 and 2 above. This will be addressed in parts (a)–(d).

   In addition, in parts (e) and (f), you will be also asked to show that if the noise in the function on the r.h.s. of a Fredholm integral equation of the first kind is "too large", the reconstruction of a "decent-looking" data $w(x)$ may be quite problematic, at least for a given choice of the smoothing matrix B.

   (a) Download the code `hw11_p3.m`. The amplitude of noise of the measurements collected by the gold prospector (recall Lecture 10) should be set to `ep=0.001*max(abs(mu))` in line 24. (The prospector has done a very careful measurement job.) Also, the parameter `whichB` in line 34 should be set to 1 (read the explanation on lines 34-36 as to what this option means). Experiment with the parameter α (line 31) and find its near-optimum value that produces a reconstructed data vector `wep` that is reasonably close to the exact data vector `w`. (Pay attention to the *scale of the vertical axis*!) The *technical notes* for Problem 1(a) apply here as well. Attach the plot of your "best" `wep` (Figure 1) and mark the value of α on it.

   Note 1: To get a ballpark for the optimal value of α, examine Eq. (42) in Lecture 11. In this regard, think what you could say about the size of the first few coefficients m_j, $j \gtrsim 1$, given the value of `ep`. Also, recall what you know from Problem 2 about the size of the first few eigenvalues λ_j, $j \gtrsim 1$ of A.

   Note 2: Unlike in the curve-smoothing problem, here we do *not* need to care much about the reconstructed $w(x)$ being smooth. What we rather have is that this $w(x)$ be as close as possible to the exact one. Indeed, the prospector of Lecture 10 probably wouldn't mind if his estimate of the gold density has fast but tiny noise. On the other hand, he would be most interested in the overall shape of that estimate being close to the true density profile, so as to know where he needs to dig to get as much of the gold as quickly as possible.

   Note 3: In light of these two Notes, you may estimate the minimum size of the noise in the reconstructed gold density that one can obtain the for the optimal α. Use this estimate as a sanity check of your results. Namely, you know that for the α that you have observed to be optimal, your noise has to be in the ballpark of your estimate. Of course, you can decrease the noise by increasing α, but then you will be distorting the shape of your reconstructed solution. This is because the exact solution, $w(x) = 1$, can be shown to be a superposition of mostly the first two largest and even (in x) eigenvectors:

   $$w(x) \approx m_{20}\, \underline{v}_{20} + m_{18}\, \underline{v}_{18}\,, \tag{HW11.1}$$

   where $\underline{v}_{20}$ and $\underline{v}_{18}$ are practically the same as $\underline{v}_{16}$ and $\underline{v}_{14}$ in Problem 2(b). Moreover, one can show that $|m_{20}| \approx 10|m_{18}|$; i.e. $\underline{v}_{20}$ contributes much more to the reconstructed solution than $\underline{v}_{18}$. This observation, along with the magnitudes of λ_{20} and λ_{18}, will also help you explain the magnitude of the optimal α that you have found by experimentation.

   (b) Repeat part (a) for `whichB=2`. Be prepared that the magnitude of α in this case will be *much* different from that in case (a). (See *Note 1* below.)

   Write your conclusion as to which smoothing matrix (i.e. that of part (a) or that of part (b)) produces better results for this problem.

Note 1: An explanation as to why the magnitude of the optimal α's in parts (a) and (b) is so different can be deduced from discussions in Sec. 11.4, given the known form of the smoothing matrices B.[2] To obtain the largest eigenvalue[3] of B in part (b), you can again use the command `eigs`.

Note 2: The above Note explains the size of the optimal α relative to that in part (a). That is, it explains how small (or large) α can be to prevent the reconstructed solution from being contaminated by noise. However, it does not explain the shape of the optimally reconstructed $w(x)$. This shape (which is determined by the "highest" eigenvectors of A) cannot be explained as easily as (briefly sketched) in Note 3 of part (a) because here, the eigenvectors of B are not the same as eigenvectors of A. Consequently, the solution of the regularized problem (26) of Lecture 11 will no longer be given by a simple expression (39). Rather, *all* of the coefficients m_j will contribute to the shape of the solution of problem (26). In fact, if you decide to look at the shape of the 5th highest eigenvector of A, you will notice its relation to the shape of your optimally reconstructed solution.

(c) Repeat part (a) while making the following modifications to the code: (i) set `whichB=1;` (ii) in line 17, define `w=(sin(pi*xx)).^2;`.

Note: The results that you observe here, compared to those in part (a), can be explained by the fact that while $w(x)$ is still approximately given by Eq. (HW11.1), the relative contribution of the two eigenvectors is quite different: $|m_{20}|$ is only about two times greater than $|m_{18}|$. Thus, any distortions introduced by α to the 18th eigenvector are reflected much more prominently in the shape of the reconstructed solution than it was the case in part (a).

(d) Repeat part (c) for `whichB=2`. Also, write your conclusion as to which smoothing matrix produces better results *for this* $w(x)$.

Note: At a qualitative level, the difference between the results that you observe here and in part (b) comes from the fact that the exact $w(x)$ in this part happens to be close to the 1st eigenvector of B. This can be shown (probably through a very involved calculation) to result in the next few "higher" eigenvectors of B to contribute less than in part (b) to a distortion caused by the regularization to the shape of the reconstructed solution.

(e) Now suppose that another prospector had thought that the first prospector (of part (a)) has wasted too much effort on collecting very precise measurements. (Indeed, in the real world, there are so many sources of error that striving for an accuracy of 0.1% can be thought of as being an overkill.) So, this other prospector decided that he would be content with the error of 2%. Repeat part (c) for `ep=0.02*max(abs(mu))` (and `whichB=1`).

(f) Repeat part (e) for `whichB=2`. State your observations regarding parts (e) and (f).

(g) Finally, summarize your findings by answering the following questions.

(i) Does one of the two smoothing matrices considered here *always* produce better results than the other one? If 'yes', which one performs better? (Support your answer by quoting specific observations.) If 'no', what does the performance of the particular smoothing matrix seem to depend on?

(ii) Does the success of the Tikhonov regularization seem to depend on the amplitude of the noise? Provide an explanation for your answer based on a discussion found in Sec. 11.4 (you may also review Notes 1 and 3 at the end of part (a)). In your explanation, explicitly state whether for a higher magnitude

[2] According to *Note 2* at the end of part (a), a better name for this matrix could be, e.g., the "regularizing matrix". However, we will continue using the term "smoothing matrix" for the sake of keeping our notations uniform.

[3] Why "largest"? In Note 1 of part (a) we said that the noise is due to the *smallest* eigenvalues of A...

of noise in the measured data $\mu(x)$, it is amplified more by solving problem (26) or it cannot be suppressed sufficiently.

Bonus **(1.5 points) Credit will be given only if the problem is done mostly correctly.**
When doing Problem 1, you probably noticed that the smoothened curve deviated most significantly from u_{exact} at the end points of the time interval. The reason for that is the approximation made in Eq. (19) of the notes where we neglected the last two terms in Eq. (18). Therefore, to improve the behavior of the smoothened curve near the end points, you need to repeat the analysis of Sec. 11.2 while correctly accounting for those terms.

(a) Modify that analysis and obtain a corrected version of Eq. (24b).

(b) Modify the code `hw11_p1.m` to account for the corrections you made in part (a). (Do this only for `der2Bmin=2`.) Attach printouts of plots that prove that the smoothened curve obtained by your modified code has a better endpoint behavior than the curve obtained for the same parameters by `hw11_p1.m`. (Do not change `tmax`, `N`, or `ep`; I will *not* give any credit for an "improved" curve obtained by tweaking those parameters.) Also, attach a printout of your modified code.
Note: Improved curves produced by my own modified version of `hw11_p1.m` have a good behavior at the left end point but still bend away from u_{exact} at the right endpoint. I do not know what causes this behavior, but am simply alerting you of the fact that you may run into a similar effect.