18.303 Problem Set 2

Due Wednesday, 22 September 2010.

Note: for Matlab problems in the 18.303 problem sets, you should turn in with your solutions a printout of any commands you used and their results, if any (please edit out extraneous/irrelevant stuff), and a printout of any graphs requested (please label the axes with the `xlabel` and YLABEL commands, add a legend with the `legend` command, and add a title with the `title` command—you should get into the habit of *always* labelling graphs properly).

Note: you can run Matlab on Athena with the command `add matlab && matlab &` at the Athena prompt. See also `http://web.mit.edu/matlab/www/` for more info about Matlab at MIT. As an alternative, if you want to use your own computer, you can also download the free GNU Octave program (google it), which accepts exactly the same commands as Matlab, although Octave's plots aren't quite as pretty and don't have Matlab's nice menu of commands. See also the Matlab cheat-sheet handout from class for a few Matlab commands to get you started (and there are numerous Matlab help pages and tutorials online that you can find by googling).

Problem 1: Finite-difference approximations

For this question, you may find it helpful to refer to the notes and reading from lecture 3. Consider a finite-difference approximation of the form:

$$u'(x) \approx \frac{-u(x+2\Delta x) + c \cdot u(x+\Delta x) - c \cdot u(x-\Delta x) + u(x-2\Delta x)}{d \cdot \Delta x}.$$

(a) Substituting the Taylor series for $u(x+\Delta x)$ etcetera (assuming u is a smooth function with a convergent Taylor series, blah blah), show that by an appropriate choice of the constants c and d you can make this approximation *fourth-order accurate*: that is, the errors are proportional to $(\Delta x)^4$ for small Δx.

(b) Check your answer to the previous part by numerically computing $u'(1)$ for $u(x) = \sin(x)$, as a function of Δx, exactly as in the handout from class (refer to the handout for the relevant Matlab commands, and adapt them as needed). Verify from your log-log plot of the |errors| versus Δx that you obtained the expected fourth-order accuracy.

Problem 2: Inner products, transposes, and symmetry

Here, we consider inner products $\langle u, v\rangle$ on some vector space V of real-valued functions and the corresponding transpose $\hat{A}^T$ of real-valued operators $\hat{A}$, where the transpose is defined, as in class, by whatever satisfies $\langle u, \hat{A}v\rangle = \langle \hat{A}^T u, v\rangle$ for all u and v in the vector space (usually, $\hat{A}^T$ is obtained from $\hat{A}$ by some kind of integration by parts). All of the proofs below should only require one or two lines each.

(a) Suppose V consists of the functions $u(x)$ on $x \in [0, L]$ with boundary conditions $u'(0) = u'(L) = 0$, and the inner product is $\langle u, v\rangle = \int_0^L u(x)v(x)dx$. Show that $\hat{A} = -d^2/dx^2$ is symmetric. Is it positive-definite?

(b) Suppose that V consists of functions $u(x)$ on $x \in [0, L]$ with boundary conditions $u(0) = u(L) = 0$, and the inner product is $\langle u, v\rangle = \int_0^L u(x)v(x)dx$. Show that $\hat{A} = -\frac{1}{w(x)}\frac{d^2}{dx^2}$, for some $w(x) > 0$, is *not* symmetric unless $w(x) = \text{constant}$.

(c) Suppose that V consists of functions with boundary conditions $u(0) = u(L) = 0$, and the inner product is $\langle u, v\rangle = \int_0^L w(x)u(x)v(x)dx$ for some function $w(x) > 0$.

(i) Show that this still satisfies the key properties of an inner product: $\langle u, v\rangle = \langle v, u\rangle$, $\langle \alpha u_1 + \beta u_2, v\rangle = \alpha\langle u_1, v\rangle + \beta\langle u_2, v\rangle$, $\langle u, u\rangle \geq 0$ and $= 0$ only if $u = 0$ (not counting isolated points).

(ii) Show that $\hat{A} = -\frac{1}{w(x)}\frac{d^2}{dx^2}$, for some function $w(x) > 0$, *is* symmetric positive-definite with this inner product on this space. What can you conclude about the eigenvalues and eigenfunctions of $\hat{A}$? (Don't try to prove diagonalizability, it's too hard.) Moral: sometimes symmetry is non-obvious, and requires you to choose the "right" inner product.

Problem 3: Asymmetrically Weighted Discrete Laplacians

For this problem, the Matlab function `diff1.m` linked on the web site will be helpful. If you download this file and put it in your directory (the one you run Matlab from), then the command

```
D = diff1(n);
```

will return the $(n+1) \times n$ first-derivative matrix from class, with $\Delta x = 1$, so that the discrete 1d Laplacian is $D^T D$ (`D'D` in Matlab).

Now, we will construct the discrete version of the operator $\hat{A} = -\frac{1}{w(x)}\frac{d^2}{dx^2}$, with $w(x) = e^{-x}$ on the vector space of functions $u(x)$ on $x \in [0, L]$ with $u(0) = u(L) = 0$. Set $L = 1$ with $n = 100$ grid points so that $\Delta x = L/(n+1) = 1/101$. Then, the corresponding center-difference matrix A is constructed by the Matlab commands:

```
L = 1; n = 100; dx = L / (n + 1);
x = [1:n]' * dx;
D = diff1(n);
A = diag(1 ./ exp(-x)) * D' * D / dx^2;
```

Compute the eigenvalues and eigenvectors of A via the Matlab command `[V,S] = eig(A)`. Here, the columns of V are the eigenvectors and S is a diagonal matrix of the eigenvalues.

(a) You can get a list of the eigenvalues by typing `diag(S)` in Matlab. Verify that they are real, as predicted in problem 2(c) above. You can sort them in ascending order and plot them, along with plotting the eigenvalues of the discrete Laplacian $-d^2/dx^2$ without $w(x)$, with the command: `plot(sort(diag(S)), 'bo', sort(eig(D'*D/dx^2)), 'r*'); legend('weighted Laplacian', 'ordinary Laplacian')` .... does $w(x)$ make the eigenvalues bigger or smaller?

(b) Get the first three eigenvectors corresponding to the smallest three eigenvalues with `[s,i]=sort(diag(S)); u1 = V(:,i(1)); u2 = V(:,i(2)); u3 = V(:,i(3));` from the matrix V, and plot them: `plot(x, u1, 'r-', x, u2, 'b--', x, u3, 'k:'); legend('u_1', 'u_2', 'u_3')` ... how do these compare to the eigenfunctions $\sin(n\pi x/L)$ of $-d^2/dx^2$?

(c) Check that the vector are *not* orthogonal under the usual inner product $x^T y$: check that `u1' * u2`, `u1' * u3`, and `u2' * u3` are all $\neq 0$, nor would we expect them to be since A is not symmetric.

(d) Problem 2(c) above predicts that the eigenvectors *should* be orthogonal under a modified inner product. What is this inner product? Check that is indeed the case with your three eigenvectors by implementing this inner product in Matlab. (Hint: you can multiply a vector v pointwise by e^{-x} using `v .* exp(-x)` in Matlab.) **Note:** because of rounding errors, even "orthogonal" vectors will have slightly nonzero dot products, so for our purposes anything smaller than 10^{-13} or so counts as "zero."