18.303 Problem Set 2

Due Wednesday, 21 September 2011.

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 "forward" finite-difference approximation of the form:

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

(It is "forward" in the sense that it only uses abscissae $\geq x$.)

(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 a, b, c you can make this approximation *second-order accurate*: that is, the errors are proportional to $(\Delta x)^2$ 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 second-order accuracy.

Problem 2: Inner products, adjoints, and definiteness

Here, we consider inner products $\langle u, v\rangle$ on some vector space V of real-valued functions and the corresponding adjoint $\hat{A}^*$ of real-valued operators $\hat{A}$, where the transpose is defined, as in class, by whatever satisfies $\langle u, \hat{A}v\rangle = \langle \hat{A}^* u, v\rangle$ for all u and v in the vector space (usually, $\hat{A}^*$ is obtained from $\hat{A}$ by some kind of integration by parts).

(a) Suppose V consists of the functions $u(x)$ on $x \in [0, L]$ with boundary conditions $u(0) = 0$ and $u'(L) = 0$, 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 self-adjoint ($\hat{A} = \hat{A}^*$). Is it positive-definite? Is this consistent with what you found in problem 2(a) of pset 1?

(b) Suppose that $\hat{A} = -\frac{d^2}{dx^2} + q(x)$ where $q(x)$ is some (integrable) function with $q(x) \geq q_0$ for some constant q_0. Show that this is self-adjoint, for the same vector space V and inner product as in the previous part. Furthermore, show that all eigenvalues λ of $\hat{A}$ are $\geq q_0$. (Hint: consider whether $\hat{A} - q_0$ is definite.)

Problem 3: Practice with eigenvectors in Matlab

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), assuming Dirichlet boundaries $u(0) = u(L) = 0$.

Now, we will construct the discrete version of the operator $\hat{A} = -\frac{d^2}{dx^2} + q(x)$, with $q(x) = 100\cos(2\pi x/L)$, 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 = D’ * D / dx^2 + diag(100*cos(2*pi*x/L));
```

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. You can sort them in ascending order and plot them, along with plotting the eigenvalues of the discrete Laplacian $-d^2/dx^2$ without $q(x)$, with the command: `plot(sort(diag(S)), ’bo’, sort(eig(D’*D/dx^2)), ’r*’); legend(’Laplacian + cosine’, ’ordinary Laplacian’)` .... Verify that they are real and > -100, as predicted in problem 2(b) above.

(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 orthogonal under the usual inner product $x^T y$: check that `u1’ * u2`, `u1’ * u3`, and `u2’ * u3` are all nearly zero., **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.”

Problem 4: Fun with symmetry—redefining transposition

For $\hat{A} = -\frac{d^2}{dx^2}$ on functions $u(x)$ with Dirichlet boundary conditions $u(0) = u(L) = 0$ on the domain $\Omega = [0, L]$, we discretized it in class, with center-difference approximations, to obtain the approximate “discrete Laplacian” matrix A:

$$A = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

operating on vectors of M points $u_1, \ldots, u_M$, where $\Delta x = L/(M+1)$ and $u(m\Delta x) \approx u_m$. Let’s say $M = 11$ and $L = 12$ (so that $\Delta x = 1$).

Now, we will restrict our vector space even further. We will consider only *mirror-symmetric* functions $u(x)$ with $u(L - x) = u(x)$. That is, $u(x)$ is even-symmetric around $x = L/2$. This will throw out half of our eigenfunctions $\sin(n\pi x/L)$ of $\hat{A}$, leaving us only with the *odd* $n = 1, 3, 5, \ldots$ eigensolutions (which are mirror-symmetric). You will now consider the finite-difference matrix version of this symmetry.

(a) Because of our mirror symmetry, five of our points are redundant: $u_1 = u_{11}$, $u_2 = u_{10}$, $u_3 = u_9$, $u_4 = u_8$, $u_5 = u_7$. Write down a new 6×6 matrix A_{sym} that acts only on the vector of 6 points $\mathbf{u}_{\text{sym}} = (u_1, u_2, \ldots, u_6)^T$, but which gives the *same* results for those 6 points as A does on the whole (mirror-symmetric) vector $\mathbf{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_5, u_4, u_3, u_2, u_1)^T$. That is, $A_{\text{sym}}\mathbf{u}_{\text{sym}}$ equals the first 6 components of $A\mathbf{u}$.

(b) You should find that A_{sym} is *not* a symmetric matrix in the usual sense ($A_{\text{sym}}^T \neq A_{\text{sym}}$). However, check in Matlab that it has real eigenvalues, and in fact that its eigenvalues are exactly 6 of the eigenvalues of A .

For example, you can define the 11×11 matrix A in Matlab by:

```
A = diag([2 2 2 2 2 2 2 2 2 2 2]) - diag([1 1 1 1 1 1 1 1 1 1], 1) - diag([1 1 1 1 1 1 1 1
1 1], -1)
```

You can get a list of the eigenvalues by `eig(A)`. Modify as needed for A_{sym}.

(c) Define a new inner product $\langle \mathbf{u}_{\text{sym}}, \mathbf{v}_{\text{sym}} \rangle$ on our real 6-component vectors, chosen so that $\langle \mathbf{u}_{\text{sym}}, \mathbf{v}_{\text{sym}} \rangle$ is *exactly* equal to the ordinary inner product $\mathbf{u}^T\mathbf{v}$ of the corresponding 11-component mirror-symmetric vectors. Hint: you should be able to write it as $\mathbf{u}_{\text{sym}}^T W \mathbf{v}_{\text{sym}}$ for some diagonal matrix W.

(d) Verify that A_{sym} *is* self-adjoint under your new inner product: i.e. check (by explicit hand calculation) that $\langle \mathbf{u}_{\text{sym}}, A_{\text{sym}}\mathbf{v}_{\text{sym}} \rangle = \langle A_{\text{sym}}\mathbf{u}_{\text{sym}}, \mathbf{v}_{\text{sym}} \rangle$ for all vectors $\mathbf{u}_{\text{sym}}$ and $\mathbf{v}_{\text{sym}}$. (This means that the eigenvalues are real, the eigenvectors are orthogonal under the new inner product, etcetera.)