18.303 Problem Set 2

Due Monday, 28 September 2015.

Problem 1: Hermitian vs. Self-adjoint

One of the tricky issues that we will mostly gloss over in this class is the distinction between Hermitian (or "symmetric") operators and self-adjoint operators. $\hat{A}$ Hermitian means that $\langle u, \hat{A}v\rangle = \langle \hat{A}u, v\rangle$ for all u, v in the domain of $\hat{A}$. To be truly self-adjoint, however, we also need for $\hat{A}^*$ to have the same domain as $\hat{A}$. In this problem you will explore this distinction a *little* bit, for the famously problematic example of the operator $\hat{A} = i\frac{\partial}{\partial x}$ acting on differentiable functions $u(x)$ on $\Omega = [0, L]$ with $u(0) = u(L) = 0$, for the inner product $\langle u, v\rangle = \int_0^L \overline{u(x)}v(x)dx$.

(a) Show that $\langle \hat{A}^* u, v\rangle = \langle u, \hat{A}v\rangle$ for all (differentiable) v with $v(0) = v(L) = 0$ with $\hat{A} = i\frac{\partial}{\partial x}$. However, you should be able to see that this is still true even for u that *don't* satisfy the boundary conditions, so that the domain of $\hat{A}^*$ is larger than the domain of $\hat{A}$.

The analogous property in the discretized (matrix) system is the fact that our first-derivative matrix D from class was *non-square*: it was $(N + 1) \times N$, so that D^T operated on a larger (higher-dimensional) vector space than D. Non-square matrices don't have eigenvectors!

(b) What happens if you try to find eigenfunctions $\hat{A}u = \lambda u$ satisfying these boundary conditions? (Something bad!)

(c) Describe the analogue of the 18.06 singular-value decomposition (SVD) for $\hat{A}$: orthogonal bases for the inputs (domain) and outputs (range) of $\hat{A}$.

(d) Consider instead the same operator $\hat{A}$, but for *periodic* functions $u(L) = u(0)$.

(i) In this case, show that $\hat{A}^* = \hat{A}$: $\langle \hat{A}^* u, v\rangle = \langle u, \hat{A}v\rangle$ is only true if *both* u and v are periodic.

(ii) Show that, in this case, $\hat{A}$ has perfectly okay periodic eigenfunctions (what are they?).

Problem 2: Modified inner products for column vectors

Consider the inner product $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^* B\mathbf{y}$ from class (lecture 5.5 notes), where the vectors are in $\mathbb{C}^N$ and B is an $N \times N$ Hermitian positive-definite matrix.

(a) Show that this inner product satisfies the required properties of inner products from class: $\langle \mathbf{x}, \mathbf{y}\rangle = \overline{\langle \mathbf{y}, \mathbf{x}\rangle}$, $\langle \mathbf{x}, \mathbf{x}\rangle > 0$ except for $\mathbf{x} = 0$. (Linearity $\langle \mathbf{x}, \alpha\mathbf{y} + \mathbf{z}\rangle = \alpha\langle \mathbf{x}, \mathbf{y}\rangle + \langle \mathbf{x}, \mathbf{z}\rangle$ is obvious from linearity the of matrix operations; you need not show it.)

(b) If M is an arbitrary (possibly complex) $N \times N$ matrix, define the adjoint $M^\dagger$ by $\langle \mathbf{x}, M\mathbf{y}\rangle = \langle M^\dagger \mathbf{x}, \mathbf{y}\rangle$ (for all $\mathbf{x}, \mathbf{y}$). (In this problem, we use $\dagger$ instead of $*$ for the adjoint in order to avoid confusion with the conjugate transpose: for this inner product, the adjoint $M^\dagger$ is *not* the conjugate transpose $M^* = \overline{M^T}$.) Give an explicit formula for $M^\dagger$ in terms of M and B .

(c) Using your formula from above, show that $M^\dagger = M$ (i.e., M is self-adjoint/Hermitian for *this* inner product) if $M = B^{-1}A$ for some $A = A^*$.

(d) In Julia, construct a random 5×5 Hermitian matrix A in Julia by `A=randn(5,5); A=A+A'`, and a random 5×5 Hermitian positive-definite matrix B by `B=randn(5,5); B=B'*B`, and check that $M = B^{-1}A$ has real eigenvalues by `eigvals(B\A)`. [Optional: also check that the eigenvectors, from the `eig` function, are orthogonal under your inner product.]

(e) In Julia, do the same thing except just make a random indefinite matrix B by `B=randn(5,5); B=B+B'`. Check that `eigvals(B\A)` does *not* generally give real eigenvalues (unless you get "lucky" and B happens to be definite by chance, but you can just repeat the process a few times to be sure).

Problem 3: More Hermitian operators

Consider

$$\hat{A}u = a\frac{d}{dx}\left[b\frac{d(cu)}{dx}\right] + d$$

for some real-valued functions $a(x) > 0$, $b(x)$, $c(x) > 0$, and $d(x)$, acting on functions $u(x)$ defined on $\Omega = [0, L]$.

(a) Show that $\hat{A}$ is Hermitian for an appropriate choice of inner product $\langle u, v\rangle$ when:

  (i) You have Dirichlet boundary conditions $u(0) = u(L) = 0$

  (ii) You have "Neumann" boundary conditions $(cu)'(0) = (cu)'(L) = 0$

  (iii) You have periodic boundary conditions $u(0) = u(L)$, and the coefficients are also periodic.

(b) Under what conditions on b and d, and for which of the above boundary conditions, is $\hat{A}$ positive-definite?

Problem 4: Finite-difference approximations

For this question you may find it helpful to refer to the notes and readings from lecture 3. Suppose that we want to compute the operation

$$\hat{A}u = \frac{d}{dx}\left[c\frac{du}{dx}\right]$$

for some smooth function $c(x)$ (you can assume c has a convergent Taylor series everywhere). Now, we want to construct a finite-difference approximation for $\hat{A}$ with $u(x)$ on $\Omega = [0, L]$ and Dirichlet boundary conditions $u(0) = u(L) = 0$, similar to class, approximating $u(m\Delta x) \approx u_m$ for M equally spaced points $m = 1, 2, \ldots, M$, $u_0 = u_{M+1} = 0$, and $\Delta x = \frac{L}{M+1}$.

(a) Using center-difference operations, construct a finite-difference approximation for $\hat{A}u$ evaluated at $m\Delta x$. (Hint: use a centered first-derivative evaluated at grid points $m + 0.5$, as in class, followed by multiplication by c, followed by another centered first derivative. Do *not* separate $\hat{A}u$ by the product rule into $c'u' + cu''$ first, as that will make the factorization in part (d) more difficult.)

(b) Show that your finite-difference expressions correspond to approximating $\hat{A}u$ by $A\mathbf{u}$ where $\mathbf{u}$ is the column vector of the M points u_m and A is a real-symmetric matrix of the form $A = -D^TCD$ (give C, and show that D is the same as the 1st-derivative matrix from lecture).

(c) In Julia, the `diagm(c)` command will create a diagonal matrix from a vector `c`. The function `diff1(M) = [ [1.0 zeros(1,M-1)]; diagm(ones(M-1),1) - eye(M) ]` will allow you to create the $(M+1)\times M$ matrix D from class via `D = diff1(M)` for any given value of M. Using these two commands, construct the matrix A from part (d) for $M = 100$ and $L = 1$ and $c(x) = e^{3x}$ via

```
L = 1
M = 100
D = diff1(M)
```

```
dx = L / (M+1)
x = dx*0.5:dx:L # sequence of x values from 0.5*dx to <= L in steps of dx
C = ....something from c(x)...hint:  use diagm...
A = -D' * C * D / dx^2
```

You can now get the eigenvalues and eigenvectors by `λ, U = eig(A)`, where λ is an array of eigenvalues and U is a matrix whose columns are the corresponding eigenvectors (notice that all the λ are < 0 since A is negative-definite).

(i) Plot the eigenvectors for the smallest-magnitude four eigenvalues. Since the eigenvalues are negative and are sorted in increasing order, these are the *last* four columns of U. You can plot them with:

```
using PyPlot
plot(dx:dx:L-dx, U[:,end-3:end])
xlabel("x"); ylabel("eigenfunctions")
legend(["fourth", "third", "second", "first"])
```

(ii) Verify that the first two eigenfunctions are indeed orthogonal with `dot(U[:,end], U[:,end-1])` in Julia, which should be zero up to roundoff errors $\lesssim 10^{-15}$.

(iii) Verify that you are getting second-order convergence of the eigenvalues: compute the smallest-magnitude eigenvalue λ_M`[end]` for $M = 100, 200, 400, 800$ and check that the *differences* are decreasing by roughly a factor of 4 (i.e. $|\lambda_{100} - \lambda_{200}|$ should be about 4 times larger than $|\lambda_{200} - \lambda_{400}|$, and so on), since doubling the resolution should multiply errors by $1/4$.

(d) For $c(x) = 1$, we saw in class that the eigenfunctions are $\sin(n\pi x/L)$. How do these compare to the eigenvectors you plotted in the previous part? Try changing $c(x)$ to some other function (note: still needs to be real and > 0), and see how different you can make the eigenfunctions from $\sin(n\pi x/L)$. Is there some feature that always remains similar, no matter how much you change c?

(e) How could you change your code to handle $\hat{A} = c\frac{d^2}{dx^2}$?