Numerical Methods II, Courant Institute, Spring 2016

http://www.math.nyu.edu/faculty/goodman/teaching/NumMethII2016/index.html

Always check the `classes` message board before doing any work on the assignment.

Assignment 1, due February ?

Corrections: [(1f) corrected to have $\frac{1}{\Delta x}$ instead of $\frac{1}{\Delta x^2}$, (2b) corrected to have a square root, (4d) corrected to fix typo and change notation: $e^{\mu x} \rightarrow e^{ax}$, and add more explanation. Equation (1) formatted (slightly) differently and explained.]

1. The matrix

$$L = \begin{pmatrix} -2 & 1 & 0 & & \cdots & & 0 \\ 1 & -2 & 1 & 0 & & & \\ 0 & 1 & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & & & \ddots & 0 \\ & & & & \ddots & -2 & 1 \\ 0 & & \cdots & & 0 & 1 & -2 \end{pmatrix}$$

is the three point *discrete Laplace* operator in one dimension with Dirichlet boundary conditions. This matrix is so important for numerical analysis that it is worth spending a lot of time learning about it. Suppose the components of $U \in \mathbb{R}^N$ are the values of a function u on a uniform grid in the interval $[0, R]$: $U_j = u(x_j)$, with $x_j = j\Delta x$, $(j = 1, \cdots, N)$ and $(N+1)\Delta x = R$. If

$$V = \frac{1}{\Delta x^2} L U \ ,$$

then the components of V are finite difference approximations to the second derivative of u, assuming that u satisfies the *Dirichlet* boundary conditions $u(0) = 0$ and $u(R) = 0$. If u were a function of more variables, for example $u(x, y)$, the *Laplace operator* on u is the sum of the second derivatives

$$\triangle u = \partial_x^2 u + \partial_y^2 u \ .$$

In one dimension, the Laplace operator is just the second derivative. For j in the "interior" ($j > 1$ and $j < N$),

$$\begin{aligned} V_j &= \frac{1}{\Delta x^2}\left(u(x_j + \Delta x) - 2u(x_j) + u(x_j - \Delta x)\right) \\ &= \partial_x^2 u(x_j) + O(\Delta x^2) \ . \end{aligned}$$

Note that this formula holds also for $j = 1$, because $u(x_1 - \Delta x) = u(0) = 0$, and also for $j = N$.

(a) The eigenfunctions of the second derivative operator on $[0.R]$ with Dirichlet boundary conditions are

$$u_k(x) = \sin(k\pi x/R) \; .$$

These satisfy the Dirichlet boundary condition and the eigenvalue relation

$$\triangle u_k = \mu_k u_k \;\; , \quad \mu_k = -\frac{\pi^2 k^2}{R^2} \; .$$

Show that if U_k is the vector with components $U_{j,k} = u_k(x_j)$, then U_k is an eigenvector of L. Find the N eigenvalues of L, λ_k. Show that they are all negative.

(b) Show by direct calculation that the "low lying" eigenvalues, λ_k for k not large, are consistent with the corresponding μ_k in the sense that $\lambda_k/\Delta x^2 \approx \mu_k$.

(c) Since L is a symmetric matrix, the eigenvalues are real (which we already know) and the eigenvectors corresponding to distinct eigenvalues are orthogonal (which isn't that hard to check if you wish). You may assume the orthogonality for this problem. The *quadratic form* associated with L is $F(U) = \frac{1}{2}U^t LU$. Use the sign of the eigenvalues to show that $F(U) < 0$ if $U \neq 0$. Hint: write an expression for $F(U)$ that involves the eigenvalues and eigenvectors of L.

(d) Show that $\nabla F(U) = LU$ for any symmetric matrix L.

(e) Show that

$$\begin{aligned} F(U) &= -\frac{1}{2}\left[U_1^2 + \sum_{j=2}^{N} (U_j - U_{j-1})^2 + U_n^2 \right] \\ &= -\frac{1}{2}\left[\sum_{j=1}^{N+1} (U_j - U_{j-1})^2 \right] \; . \end{aligned}$$

The second version refers to U_0 and U_{N+1}, which are implicitly taken to be zero.

(f) Suppose $u(x)$ is a differentiable function. The *Dirichlet integral* is

$$D(u) = \frac{1}{2}\int_0^R \left(u'(x)\right)^2 \, dx \; .$$

Show that if $U_j = u(x_j)$ and u is a smooth function of x, then

$$D(u) = -\frac{1}{\Delta x} F(U) + O(\Delta x^2) \; .$$

The quadratic form related to L is consistent with the Dirichlet integral (with the correct Δx pre-factor).

2. In finite dimensional linear algebra there is a theorem that any two vector norms are *equivalent* in the sense that they determine the same Cauchy sequences. The technical statement is that if $\|v\|_a$ and $\|v\|_b$ are any two vector norms on $\mathbb{R}^N$, then there is a constant C_{ab} so that

$$\frac{1}{C_{ab}} \|v\|_a \leq \|v\|_b \leq C_{ab} \|v\|_a \ .$$

Another way to say this is that the *condition number* of $\|\cdot\|_b$ relative to $\|\cdot\|_a$ is well defined and finite. This condition number is

$$\kappa = \frac{\displaystyle\max_{\|v\|_a=1} \|v\|_b}{\displaystyle\min_{\|v\|_a=1} \|v\|_b} \ . \tag{1}$$

Here,

$$\max_{\|v\|_a=1} \|v\|_b$$

is the maximum value of $\|v\|_b$ over all vectors v with $\|v\|_a = 1$.

(a) Find the condition number of the discrete L^2 norm and the discrete L^∞ norm with respect to the discrete L^1 norm. In each case, the condition number is a power of N (the dimension). What are the extremal vectors that give the min and max in (1)?

(b) Suppose M is a positive definite symmetric $N \times N$ matrix. Show that $\|v\|_M = \sqrt{v^t M v}$ is a norm. Show that the condition number of the norm relative to L^2 is the square root of the condition number of M, which is $\lambda_{\max}(M)/\lambda_{\min}(M)$.

3. Consider the *advection diffusion* equation

$$\partial_t u + s\partial_x u = D\partial_x^2 u \ . \tag{2}$$

This equation models a substance that is is being carried (advected) with speed s while it diffuses with diffusion coefficient D. We may call s the "wind speed", if we think u representing the concentration of a chemical in moving air. If $s > 0$, the chemical is being carried from the left to the right. For this reason, left is the *upwind* direction and right is the *downwind* direction. If $s < 0$, then the downwind direction is to the left and upwind is to the left. You can see this by choosing a *moving coordinate*, y, so that if $x = C + st$ (x moves to the right with speed s), then $y = C$. This is $y = x - st$.

(a) Show that (2) in the moving coordinate system becomes $\partial_t u = D\partial_y^2 u$. If you move to the right with speed s and watch u, you see only diffusion, not advection. For this part, assume that u is defined for all x, not on a finite domain.

(b) Construct a finite difference approximation to (2) that builds on the finite difference approximation given in class for the heat equation. Use the second order centered difference approximation

$$\partial_x u(x,t) \approx \frac{u(x+\Delta x,t) - u(x-\Delta x,t)}{2\Delta x} \; .$$

Show that the resulting approximation is formally second order accurate as we did in class for the heat equation.

(c) Show that this method is stable, as long as $D > 0$ and Δx is small enough. The stability is in L^∞, L^2 and L^1.

4. For intuition and numerical work below, we record come exact solutions of the diffusion/heat equation and the advection/diffusion equation.

(a) The formula

$$u_b(x,t) = \frac{A}{\sqrt{t}}\, e^{-\frac{x^2}{4Dt}} \; , \tag{3}$$

represents a spreading blob. Show that this satisfies the diffusion equation (2) with $s = 0$, without boundary conditions.

(b) A blob that spreads, and moves moves with speed s, has the form

$$u_b(x - st, t) \; .$$

Show that this satisfies the advection-diffusion equation (2) for any s, if there are no boundaries. Hint: this is the same as exercise (3a) above.

(c) A *separation of variables* solution has the form $u(x,t) = f(x)g(t)$. Consider the diffusion equation (2) with $s = 0$. Apply *Dirichlet* boundary conditions $u = 0$ when $x = 0$ or $x = L$. Find the separation of variables solutions of the diffusion equation with $f(x) = \sin(\pi x/L)$. Hint: Use notation $\dot{g}(t) = \frac{dg(t)}{dt}$ and $f' = \frac{df}{dx}$. Some algebra gives $\frac{\dot{g}}{g} = \frac{Df''}{f}$. The left side depends only on t while the right side depends only on x, so both must be constant and equal to the *separation constant* λ: $\frac{\dot{g}}{g} = \lambda$ and $\frac{Df''}{f} = \lambda$.

(d) Find the separation of variables solutions of the advection-diffusion equation (2) with $f(x) = e^{ax}\sin(\pi x/L)$. Here, a is a constant that depends on s. Show that, particularly for large s, u is larger near the downwind boundary than the upwind boundary.

5. For this exercise you will need access to Python 2.7 and the packages `numpy` and `matplotlib`. Python 2.7 is available from many sources for free download. If you have trouble, please post on the class message board. Download the Python script `HeatEquationAnimation.py` that goes with this assignment. If you type `python HeatEquationAnimation.py`, it should create a movie file `HeatEquation.mp4` in the same directory. The movie

should be identical to `HeatEquationCheck.mp4` posted with this assignment. If it takes more than a few minutes to run on your computer, try making the parameter `n` smaller.

(a) Modify `HeatEquationAnimation.py` so that the initial data are $f(x) = \sin(\pi x/L)$. Check that the solution is the one you derived in exercise (4c). Of course, it won't be exactly the same, so you have to allow some error. The approximation should get better if you keep T the same and increase n.

(b) Modify `HeatEquationAnimation.py` to solve the advection-diffusion equation (2). There should be a parameter `s` in the code. Make a movie using $s = 10$ with the blob initial data and see that it moves to the right as it spreads.

(c) Check that your code gets the separation of variables solution from exercise (4d) correctly.

(d) Experiment with the code in some way that you find interesting. For example, you may see that the solution with blob initial data comes to resemble the separation of variables solution as it starts to decay. You may try blob solution that start more narrow. You may need a smaller n for the solution to be computed accurately. You may wish to violate the time step stability condition $\Delta t \leq .5 \cdot \Delta x^2$. If you do, the solution should look nothing like the true solution. If you feel like hacking the code instead, you may modify the movie code to plot the exact and approximate solution in each frame.