# 18.303 Problem Set 3

Due Wednesday, 29 September 2010.

## Problem 1: Diffusion equation

For this problem, you will be solving the diffusion equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$ for $x \in [0, L]$ with Neumann boundary conditions $u'(0) = u'(L) = 0$ and an initial condition $u(x, 0) = f(x)$. Physically, $u(x, t)$ could represent, for example, the concentration (mass per unit volume) of salt dissolved in water along a straw of length $L$ (neglecting diffusion $\perp$ to the straw), with $f(x)$ being the initial concentration. For simplicity, take $L = 1$.

Recall that, in pset 1, you found that the eigenfunctions of $d^2/dx^2$ with these boundary conditions are $\cos(n\pi x/L)$ for $n = 0, 1, 2, \ldots$ with eigenvalues $\lambda_n = -(n\pi/L)^2$. Hence, it is natural to write $f(x)$ as a cosine series

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\pi x/L),$$

where $a_n = \frac{2}{L}\int_0^L f(x)\cos(n\pi x/L)dx$ due to orthogonality. The Matlab file `cosines.m`, posted on the web page, provides a function `cosines(x, L, a)` to do this summation for a given vector `a` of the coefficients $a_n$.

(a) Suppose $f(x)$ has Fourier coefficients $a_n = \cos(n\pi/2)$ for $n = 0, 1, 2, \ldots, N$. Plot this function for $N = 10$, 100, and 1000 versus $x$. You can use the commands: `L=1; x=linspace(0,L,10000); plot(x, cosines(x, L, cos([0:N]*pi/2))` to do the plot for each $N$ (setting `N=10` etcetera first).

(b) You should be able to answer the following questions with pen and paper: How does the total initial solute mass $\int_0^L f(x)dx$ depend on $N$? What is $f(L/2)$ as a function of $N$? What does the answer to the previous two questions tell you about the width of the peak shape that you observed in the previous part, as a function of $N$?

(c) From class, $u(x, t)$ is given by the same series as $f(x)$, except that $a_n$ is multiplied by $e^{\lambda_n t}$. In Matlab, you use the commands: `L=1; x=linspace(0,L,10000); plot(x, cosines(x, L, cos([0:N]*pi/2) .* exp(-([0:N]*pi/L).^2 * t)))` to plot $u(x, t)$ for a given $t$. Do this plot for $t = 0.0001$, $t = 0.001$, $t = 0.01$, $t = 0.1$, and $t = 1$. (You can type `hold on` between plots in order to superimpose multiple plots, which might be useful for the last 3 plots where the scales are similar.) Describe physically what is happening.

(d) Check that mass is conserved: compute $\int_0^L u(x, t)dx$ approximately for a few values of $t$ via a summation: `sum(cosines(x, L, cos([0:N]*pi/2) .* exp(-([0:N]*pi/L).^2 * t))) * x(2)`. The result should equal your answer for $\int f(x)dx$ from above.

## Problem 2: Finite differences and boundary conditions

In class, we considered $d^2/dx^2$ for $x \in [0, L]$ and Dirichlet boundary conditions $u(0) = u(L) = 0$. We approximated $u(x)$ by its values $u_m \approx u(m\Delta x)$ on a grid of $M$ points, with $\Delta x = L/(M + 1)$ and $u_0 = u_{M+1} = 0$. We then approximated $d^2/dx^2$ in two center-difference steps: first, we approximated $d/dx$ by $u'_{m+1/2} \approx (u_{m+1} - u_m)/\Delta x$; then, we approximated $d^2/dx^2$ by $u''_m \approx (u'_{m+1/2} - u'_{m-1/2})/\Delta x$. This gave us the wonderful discrete Laplacian matrix $A$ (with $1, -2, 1$ on the diagonals), which we could also write as $A = -D^T D/\Delta x^2$ for the $D$ matrix given in class (and computed by the `diff1.m` file from pset 2). Now, in this problem we will do the same thing for the Neumann boundary conditions $u'(0) = u'(L) = 0$.

(a) Find the *Neumann* discrete Laplacian matrix $\tilde{A}$. Use the same center-difference steps that we used to find $A$ in class, except now impose the boundary condition $u'_{1/2} = u'_{M+1/2} = 0$ to approximate the Neumann condition.

(b) Show that $\tilde{A} = -\tilde{D}^T\tilde{D}/\Delta x^2$ for a matrix $\tilde{D}$ that can be obtained from $D$ just by deleting rows and/or columns. Modify the `diff1.m` file to compute $\tilde{D}$ instead of $D$, making a new file `diff1n.m`. [For example, given a matrix $M$, in Matlab you can make a submatrix of rows 3 to 5 and columns 8 to 17 by `M(3:5, 8:17)`, or columns 8 to 17 and all the rows by `M(:, 8:17)`.]

(c) One of the nice theorems from 18.06 was that, for matrices of the form $\tilde{A} = \alpha\tilde{D}^T\tilde{D}$ for $\alpha \neq 0$, $\text{rank}(\tilde{A}) = \text{rank}(\tilde{D}) = \text{rank}(\tilde{D}^T)$. By inspection of either $\tilde{D}$ or $\tilde{D}^T$, say what the rank of $\tilde{A}$ is and the dimension of the nullspace $N(\tilde{A})$. Give a basis for the nullspace of $\tilde{A}$. (This should be all possible by inspection, little computation required!) Compare to the nullspace of $d^2/dx^2$ with these boundary conditions, from pset 1.

(d) Using Matlab code similar to problem 3 of pset 2, compute the eigenvalues and eigenvectors of $\tilde{A}$, and compare the solutions with the smallest three $|\lambda|$ to the corresponding exact eigensolutions for $d^2/dx^2$ with these boundary conditions. Roughly how fast does the error in $\lambda_1 \approx -(\pi/L)^2$ decrease with $\Delta x$? This is because $u'_{1/2}$ is a _________ difference approximation for $u'(0)$ and $u'_{M+1/2}$ is a __________ difference approximation for $u'(L)$.

(e) The distance from the $u'_{1/2}$ to $u'_{M+1/2}$ grid points is $L - \Delta x$, not $L$. Fix this by changing $\Delta x$ to $\Delta x = L/M$. With this new $\Delta x$, roughly how fast does the error in $\lambda_1 = -(\pi/L)^2$ decrease with $\Delta x$?

## Problem 3: Bessel, Bessel, toil and mess...el

In class, we solved for the eigenfunctions of $\nabla^2$ in two dimensions, in a cylindrical region $r \in [0, R]$, $\theta \in [0, 2\pi]$ using separation of variables, and obtained Bessel's equation and Bessel-function solutions. Although Bessel's equation has two solutions $J_m(kr)$ and $Y_m(kr)$ (the *Bessel functions*), the second solution ($Y_m$) blows up as $r \to 0$ and so for that problem we could only have $J_m(kr)$ solutions (although we still needed to solve a transcendental equation to obtain $k$).

In this problem, you will solve for the 2d eigenfunctions of $\nabla^2$ in an *annular* region $\Omega$ that does *not contain the origin*, as depicted schematically in Fig. 1, between radii $R_1$ and $R_2$, so that you will need *both* the $J_m$ and $Y_m$ solutions. We will impose Dirichlet boundary conditions $u|_{d\Omega} = 0$; i.e. for a function $u(r, \theta)$ in cylindrical coordinates, $u(R_1, \theta) = u(R_2, \theta) = 0$. Exactly as in class, the separation of variables ansatz $u(r, \theta) = \rho(r)\tau(\theta)$ leads to functions $\tau(\theta)$ spanned by $\sin(m\theta)$ and $\cos(m\theta)$ for integers $m$, and functions $\rho(r)$ that satisfy Bessel's equation. Thus, the eigenfunctions are of the form:

$$u(r, \theta) = [\alpha J_m(kr) + \beta Y_m(kr)] \times [A\cos(m\theta) + B\sin(m\theta)]$$

for arbitrary constants $A$ and $B$, for integers $m = 0, 1, 2, \ldots$, and for constants $\alpha$, $\beta$, and $k$ to be determined.

(a) Using the boundary conditions, write down two equations for $\alpha$, $\beta$ and $k$, of the form $E\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0$ for some $2 \times 2$ matrix $E$. This only has a solution when $\det E = 0$, and from this fact obtain a single equation for $k$ of the form $f_m(k) = 0$ for some function $f_m$ that depends on $m$. This is a transcendental equation; you can't solve it by hand for $k$. In terms of $k$ (which is still unknown), write down a possible expression for $\alpha$ and $\beta$, i.e. a basis for $N(E)$.

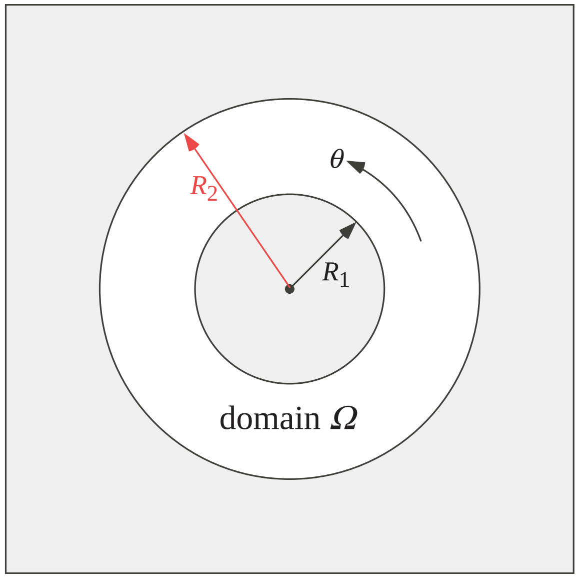


Figure 1: Schematic of the domain $\Omega$ for problem 3: an annular region in two dimensions, with radii $r \in [R_1, R_2]$ and angles $\theta \in [0, 2\pi]$.

(b) Assuming $R_1 = 1$, $R_2 = 2$, plot your function $f_m(k)$ versus $k \in [0, 20]$ for $m = 0, 1, 2$. Note that Matlab provides the Bessel functions built-in: $J_m(x)$ is `besselj(m,x)` and $Y_m(x)$ is `bessely(m,x)`. You can plot a function with the `fplot` command. For example, to plot $J_m(k) \cdot Y_m(3k)$ as a blue line, you would do `fplot(@(k) besselj(m, k) .* bessely(m, 3*k), [0,20], 'b')`, assuming you have assigned `m` to a value. [`@(k)_____` defines a function of $k$.] It might be helpful to use `hold on` between plots so that you can plot $f_0$, $f_1$, and $f_2$ on the same plot (labelled, of course).

(c) For $m = 0$, find the first three (smallest $k > 0$) solutions $k_1$, $k_2$, and $k_3$ to $f_0(k) = 0$. Get a rough estimate first from your graph (zooming if necessary), and then get an accurate answer by calling the `fzero` function (Matlab's nonlinear root-finding function). For example, to find a root of $\cos(x) - x$ near $x = 1$, you would call `fzero(@(x) cos(x) - x, 1)` in Matlab. Plot the corresponding functions $\alpha J_0(kr) + \beta Y_0(kr)$.

(d) Because $\nabla^2$ is real-symmetric, we know that the eigenfunctions must be orthogonal. From class, this implies that the radial parts must also be orthogonal when integrated via $\int r\,dr$. Check that your Bessel solutions for $k_1$ and $k_2$ are indeed orthogonal, by numerically integrating them via the `quadl` function in Matlab. In particular, if you have assigned your $\alpha, \beta, k$ solutions to variables `a1`,`b1`,`k1` and `a2`,`b2`,`k2` in Matlab, then the integral for $r \in [1, 2]$ is given to at least six digits of accuracy by `quadl(@(r) (a1*besselj(0,k1*r) + b1*bessely(0,k1*r)) .* (a2*besselj(0,k2*r) + b2*bessely(0,k2*r)) .* r, 1,2, 1e-6)` in Matlab.