Mat1062: Introductory Numerical Methods for PDE
Problem Set 1
Tuesday January 13, 2009
due: Tuesday January 24

To save figures you can use the command "print -dps figure1.ps".

1. **Do not hand in this problem. This is a problem for you to solve on your own; please see me if you need help.** Given the heat equation $u_t = \alpha^2 u_{xx}$ on $[0, L]$ with linear homogeneous boundary conditions, we seek exact solutions of the form

$$u(x,t) = T(t)X(x)$$

Plugging this assumption into the PDE, one finds

$$\frac{T'(t)}{\alpha^2 T(t)} = \frac{X''(x)}{X(x)} = k.$$

for some real number k. This implies that $T(t)$ and $X(x)$ satisfy the ODEs

$$T'(t) = k\alpha^2 T(t), \qquad X''(x) = kX(x).$$

If the constant k is a negative number then one can write k as $-\lambda^2$ and the ODEs become

$$T'(t) = -\lambda^2\alpha^2 T(t), \qquad X''(x) = -\lambda^2 X(x),$$

resulting in the solution $X(x) = A\cos(\lambda x) + B\sin(\lambda x)$ for constants A and B and a $T(t)$ that decays to zero as $t \to \infty$.

If the constant k is zero then the ODEs become

$$T'(t) = 0, \qquad X''(x) = 0,$$

resulting in the solution $X(x) = Ax + B$ for constants A and B and a $T(t)$ that is constant.

If the constant k is a positive number then one can write k as λ^2 and the ODEs become

$$T'(t) = \lambda^2\alpha^2 T(t), \qquad X''(x) = \lambda^2 X(x),$$

resulting in the solution $X(x) = A\cosh(\lambda x) + B\sinh(\lambda x)$ for constants A and B and a $T(t)$ that goes to infinity as $t \to \infty$.

   (a) If the IBVP has Dirichlet boundary conditions

$$u(0,t) = 0, \qquad u(L,t) = 0$$

   prove that there are no nontrivial solutions $X(x)$ when $k \geq 0$. That is, all separable solutions decay in time.

(b) If the IBVP has the Neumann boundary conditions

$$u_x(0,t) = 0, \qquad u_x(L,t) = 0$$

prove that there are no nontrivial solutions $X(x)$ when $k > 0$.

(c) If the IBVP has Robin boundary conditions

$$-u_x(0,t) = -\frac{h_1}{k}u(0,t), \qquad u_x(L,t) = -\frac{h_2}{k}u(L,t)$$

where h_1, h_2, and k are all positive numbers, show that there are no nontrivial solutions $X(x)$ when $k > 0$. What about if $k = 0$?

(d) If the IBVP has Robin boundary conditions

$$u_x(0,t) + 3u(0,t) = 0, \qquad u_x(2,t) - 8u(2,t) = 0$$

There are two nontrivial $X(x)$ which correspond to $k > 0$ solutions. Find them. (You will need to use a computer to help you to do this.)

2. We now want to ask the same questions as above except for the discrete problem.

(a) Write a matlab function called DirichletMatrix.m which has has two inputs: the length of the interval $[x_L, x_R]$ and the number of subintervals. It has one output: the matrix M_1 from the Jan 13 lecture notes.

(b) Take $N = 8$, $L = 1$ and call the output of your function DirichletMatrix. Use matlab's "eig" command to find the eigenvalues of M_1. You can find the largest three eigenvalues via the commands: $v = sort(eig(M_1)); v(N-3:N-1)$. Analytically, you know what the true eigenvalues are for the PDE on the interval with Dirichlet boundary conditions. If V_2 is the value for the second largest eigenvalue of the PDE problem, let err(1) = v(2)-V_2 be the difference between it and the second largest eigenvalue of the matrix.

(c) Repeat the above with $N = 16$ and define err(2) via v(2)-V_2 where v is the sorted vector of eigenvectors from the $N = 16$ computation.

(d) Repeat the above with $N = 32$ and define err(3) analogously.

(e) Repeat the above with $N = 64$ and define err(4) analogously.

(f) Repeat the above with $N = 128$ and define err(5) analogously.

(g) Repeat the above with $N = 256$ and define err(6) analogously.

(h) Give the six values of err. (Note: if you type "format short e" you'll get a better display of the error. You should notice they are decreasing. Now give the subsequent ratios err(1)/err(2) and so on. What do you see? (Note: if you type "err(1:5)./err(2:6)" you'll get all those ratios at once.)

(i) What are the eigenvalues and eigenvectors of the matrix M_1?

3. We want to solve the partial differential equation for $u(x,t)$

$$u_t \;=\; D\,u_{xx} \qquad \text{for } 0 \le x \le L \text{ and } t \ge 0,$$

with initial and boundary data (we suppose $f(0) = f(L) = 0$)

$$u(x,0) = f(x) \;\text{ for } 0 \le x \le L; \qquad u(0,t) = u(L,t) = 0 \;\text{ for } t \ge 0.$$

(a) Write a Matlab program to compute u_j^n for $n = 0, 1, \ldots$.
How you visualize the solution is up to you. If you are motivated and have time, an excellent way to see what is going on is to use the `mesh` command to plot the surface $u(x,t)$. This is, however, no substitute for making quantitative comparisons as in the next item.

(b) For $f(x) = \sin k_0 \pi x/L$, with k_0 an integer, you can easily compute the exact solution to this PDE. (What is it?) Take $D = 1$; take $L = 2$ and $N = 10, 20, 40, 80, \ldots$. Fix $\lambda = 1/4$. The time-step, k, is then determined via $k = \frac{1}{4}h^2$. Measure the maximum difference between the solution of your difference formula at $t = 1$ and the exact solution. I.e. at $t = 1$ compute the maximum difference over x. Let e_N be this maximum error at time $t = 1$ and make a log-log plot of e_N *vs.* N, and show that $e_N \to 0$ as $N \to \infty$.

(c) Consider the initial data (suppose N is even)

$$u_j^0 \;=\; \begin{cases} 1/h, & j = N/2 \\ 0, & \text{else.} \end{cases}$$

What "function" $f(x)$ does this approximate? What will the solution look like for short times? For long times? Run your code for this initial data and compare the solution to the approximate solution of the PDE computed as though the spatial domain were infinite.
As a measure of how much the boundary conditions are affecting your solution, plot the quantity

$$I^n \;=\; h\left(\tfrac{1}{2}\big(u_0 + u_N\big) + \sum_{j=1}^{N-1} u_j^n\right) \quad\approx\quad I(t) \;=\; \int_0^L u(x,t)\,dx$$

as a function of time. How does this behave for short times, and why does it help you distinguish between short and long times?

(d) Make a copy of your code and modify it to take *homogeneous Neumann* boundary conditions

$$u_x(0,t) \;=\; u_x(L,t) \;=\; 0, \qquad t \ge 0$$

(assume now that the initial data $f(x)$ satisfies this condition).
Repeat a), b), and c) above, taking $f(x) = \cos k\pi x/L$. How does the profile of $I(t)$ change? (How does $I(t)$ evolve in your discrete model?)

4. Consider the following initial boundary value problem

$$\begin{cases} u_t = \frac{1}{2} u_{xx} & 0 < x < 2\pi \quad t > 0 \\ u(x,0) = 1 - \cos(x) & 0 \le x \le 2\pi \\ u_x(0,t) = u(0,t)^2 & t > 0 \\ u(2\pi, t) = 0 & t > 0 \end{cases}$$

Note that the boundary condition at the point $x = 0$ is *nonlinear*: $u_x = u^2$ at $x = 0$ for all $t > 0$. We have not discussed such boundary conditions before. In this problem, you will develop a finite-difference scheme to approximate solutions of the problem.

(a) Discretize space using $h = 2\pi/N$. Let $x_j = jh$. The left-hand endpoint is x_0 and the right-hand endpoint is x_N. If $U_j(t) \approx u(x_j, t)$ then the PDE, can be approximated using $N-1$ ODEs. Using centered differences to approximate u_{xx}, what are these $N-1$ ODEs? By and large, these ODEs should look familiar; the thing that will be different will be the ODE for dU_1/dt. If you don't know immediately what to do, reread section the January 13 notes and see if this helps to inspire you.

(b) If you now discretize in space with time step, $k = T/M$, and $t_n = nk$, how do you use the current values $u(x_0, t_n)$, $u(x_1, t_n)$, $\ldots u(x_{N-1}, t_n)$, $u(x_N, t_n)$ to create the values at the next time step: $u(x_0, t_{n+1})$, $u(x_1, t_{n+1})$, $\ldots u(x_{N-1}, t_{n+1})$, $u(x_N, t_{n+1})$? Use explicit Euler time-stepping for the ODEs. (The only new thing here is to figure out how to update the value for $u(x_0, t)$ — you don't have an ODE here.)

(c) Code up your scheme. Take the initial time $t = 0$ and the final time $T = 1$. Take space step $h = 2\pi/10$ and time step $k = 1/250$. Compute the approximate solution and call it u_0. Now take $h = 2\pi/20$ and $k = 1/1000$ and compute the approximate solution. Call it u_1. Finally, take $h = 2\pi/40$ and $k = 1/4000$ and compute the approximate solution. Call it u_2. Before answering the following, type the matlab command `format short e`

i. What are the values $u_0(x_0, 1)$, $u_1(x_0, 1)$, and $u_2(x_0, 1)$? What is

$$\frac{u_0(x_0,1) - u_1(x_0,1)}{u_1(x_0,1) - u_2(x_0,1)}?$$

ii. What are

$$\|u_0(:,251) - u_1(:,1001)\|_{L^\infty} \quad \text{and} \quad \|u_0(:,1001) - u_1(:,4001)\|_{L^\infty}?$$

What is the ratio of these two numbers? *The array* $u_0(:,251)$ *has length 11,* $u_1(:,1001)$ *has length 21, and* $u_2(:,4001)$ *has length 41. You want to create vectors* $u_0(:,251) - u_1(:,1001)$ *and* $u_0(:,1001) - u_1(:,4001)$ *then find their* L^∞ *norms. Find these vectors by looking only at the meshpoints that all three have in common. There are 11 such points.*

iii. On the same graph, plot $u_0(x_0, t)$, $u_1(x_0, t)$, and $u_2(x_0, t)$. Use `plot(...,'o-')`, `plot(...,'o--')`, and `plot(...,'o:')` when making the plots so that each datapoint in time is clearly marked. Set the axes of your plot so that you're looking at the short-time behaviour close to $t = 0$. What do you observe?

(d) Now consider the more general nonlinear boundary condition at the left-hand endpoint

$$u_x(0, t) = f(u(0, t), t) \qquad t > 0.$$

Discretize space using $h = 2\pi/N$. Let $x_j = jh$. The left-hand endpoint is x_0 and the right-hand endpoint is x_N. If $U_j(t) \approx u(x_j, t)$ then the PDE, can be approximated using $N - 1$ ODEs. Using centered differences to approximate u_{xx}, what are these $N - 1$ ODEs? (Again, the thing that will require thought is the ODE for dU_1/dt.)