Mat1062: Introductory Numerical Methods for PDE

Problem Set 1

Tuesday January 19, 2016

due: by 4pm, Friday January 29

You're encouraged to work in groups; just make sure to have everyone's name on the HW when you hand it in.

1. **Eigenvalues, Eigenfunctions**

   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

$$u_x(0,t) - h_1\, u(0,t) = 0, \qquad u_x(L,t) + h_2\, u(L,t)$$

where h_1 and h_2 are real numbers, show that there are no nontrivial solutions $X(x)$ when $k > 0$. What about if $k = 0$? *Note that if h_1 and h_2 are both positive then the boundary conditions correspond to energy loss through the ends. Specifically, a quick integration by parts argument show that*

$$\frac{d}{dt}\int_0^L \frac{1}{2}\, u(x,t)^2\, dx = -\int_0^L u_x(x,t)^2\, dx - h_2\, u(L,t)^2 - h_1\, u(0,t)^2 \leq 0$$

2. **Eigenvalues and Eigenvectors**

If the IBVP has Robin boundary conditions

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

There are two nontrivial $X(x)$ which correspond to $k > 0$ solutions. Find the corresponding eigenvalues. You will need to use a computer to help you to do this — it involves finding the intersection of two curves or the zero of a single function, depending on how you do these things. I'm perfectly okay if you use maple, mathematica, or wolfram alpha for this job — I don't want you to go off and code up a zero-finder. The matlab optimization toolbox has the command fzero, if you have that toolbox. You want to find these eigenvalues up to 16 significant digits because you're going to need them for the next step. Call the eigenvalues λ_1 and λ_2 where $\lambda_1 > \lambda_2$. Find the corresponding eigenfunctions $X_1(x)$ and $X_2(x)$.

Consider the spatial operator $\mathcal{L}$ defined by

$$\begin{cases} u_{xx} & \text{on } (0,2) \\ 3\, u(0) + u_x(0) = 0 \\ -8\, u(2) + u_x(2) = 0 \end{cases}$$

(a) Write a matlab function called RobinBCsApproach1.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 analogue of the $(N-1)\times(N-1)$ matrix M_2 defined in (46) on page 27 of the lecture notes.

(b) Take $N = 7$, $L = 2$ and use your function to create a $(N-1)\times(N-1)$ matrix M. Use matlab's `eig` command to find the eigenvalues of M. You can find the largest three eigenvalues in descending order via the command: `evals = sort(eig(M),'descend');`. From your rootfinding work, you know what the two largest eigenvalues should be. Taking the rootfinding eigenvalues as the true values, you can compare the finite-difference eigenvalues to them and define an error. Define `err1(1) =` λ_1`-evals(1)` and `err2(1) =` λ_2`-evals(2)`.

(c) Repeat the above with $N = 2\cdot 7$ and define `err1(2) =` λ_1`-evals(1)` and `err2(2) =` λ_2`-evals(2)`.

(d) Repeat the above with $N = 2^2 \cdot 7$ and define `err1(3)` and `err2(3)` analogously.

(e) Repeat the above with $N = 2^3 \cdot 7$ and define `err1(4)` and `err2(4)` analogously.

(f) Repeat the above with $N = 2^4 \cdot 7$ and define `err1(5)` and `err2(5)` analogously.

(g) Repeat the above with $N = 2^5 \cdot 7$ and define `err1(6)` and `err2(6)` analogously.

(h) Give the six values of `err1` and `err2`. (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 `err1(1)/err1(2)` and `err2(1)/err2(2)` and so on. What do you see? (Note: if you type `err1(1:5)./err1(2:6)` and `err2(1:5)./err2(2:6)`. you'll get all those ratios at once.)

(i) Plot the analytical eigenfunctions $X_1(x)$ and $X_2(x)$ and compare them to the eigenvectors for correspdonding eigenvectors of the discretized problem for the first few values of N. As a sample of how to get them, try the following commands as a sample:

```
A = rand(4,4)   (create a random 4x4 matrix)
A = A + A'  (make it symmetric)
[v,d] = eig(A)  (v is a matrix of eigenvectors, d is a diagonal
                matrix with eigenvalues on the diagonal)
[evals,I] = sort(diag(d),'descend')
evals(1)  (largest eigenvalue)
v1 = v(:,I(1))  (eigenvector for largest eigenvalue)
A*v1./v1  (check that I'm not lying)
evals(2)   (next largest eigenvalue)
v2 = v(:,I(2)) (eigenvector for next largest eigenvalue)
A*v2./v2 (check that I'm not lying)
```

If you're feeling ambitious, you can find an error which is the norm of the difference between the analytical eigenfunctions, sampled appropriately, and the eigenvectors and see how those errors decrease as N increases.

(j) Repeat the above but with a new matlab function called RobinBCsApproach2.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 analogue of the $(N-1)\times(N-1)$ matrix M_3 on page 29 of the lecture notes.

(k) Repeat the above but with a new matlab function called RobinBCsApproach3.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 analogue of the $(N+1)\times(N+1)$ matrix M_4 on page 30 of the lecture notes.

3. **Coding up the vanilla diffusion equation**

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 data

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

and homogeneous Dirichlet boundary conditions

$$u(0,t) = u(L,t) = 0 \;\; \text{for } t > 0.$$

(a) Write a Matlab program to compute u_j^n for $n = 0, 1, \ldots$. Use explicit time-stepping. The inputs to your code should be: x_L and x_R (the endpoints of your interval), T (the time that you want to compute up to), N (the number of subintervals you want to use in space: $h = (x_R - x_L)/N$), and M (the number of timesteps to be taken: $k = T/M$).
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 "delta function" initial data (suppose N is even)

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

where h is the distance between meshpoints on the interval $[0,1]$. Based on this initial data, what solution of the diffusion equation do you expect your solution to

approximate, at least for short times? Thinking about the boundary conditions, what do you expect your solution to do for long times? Run your code for this initial data and your solution to your guessed-at analytical solution (sampled at the same x_j and t_n as your solution).
As a measure of how the boundary conditions are affecting your solution, plot the total mass:

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

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

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

(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. **Diffusion in a two-material domain**

We want to consider the diffusion equation on an interval with non-constant diffusivity. For simplicity, we'll take Dirichlet boundary conditions:

$$\begin{cases} u_t = D_1\,u_{xx} & \forall x\in(0,1) \quad \forall t>0 \\ u_t = D_2\,u_{xx} & \forall x\in(1,2) \quad \forall t>0 \\ u(x,0)=u_0(x) & \forall x\in[0,2] \\ u(0,t)=u_L & \forall t>0 \\ u(2,t)=u_R & \forall t>0 \\ \lim_{x\uparrow 1}u(x,t)=\lim_{x\downarrow 1}u(x,t) & \forall t>0 \\ \lim_{x\uparrow 1}\;D_1u_x(x,t)=\lim_{x\downarrow 1}\;D_2u_x(x,t) & \forall t>0 \end{cases}$$

The "boundary" conditions at $x=1$ correspond to "both the temperature and the temperature flux are continuous at the material interface". Really, they're interface conditions.

(a) A steady state satisfies the PDE and the boundary/interface conditions. Find the steady state. (It'll depend on D_1, D_2, u_L, and u_R. Make sure your answer makes sense to you by considering the four cases 1) $u_L=u_R$, 2) $D_1=D_2$, 3) $D_1\ll D_2$, and 4) $D_1\gg D_2$.

(b) Consider a mesh where $N = 2N_0$ and $h = 2/N$ and $x_j = jh$ with $j = 0, \ldots N$. In this case, $x_0 = 0$, $X_{N_0} = 1$, and $X_N = 2$. That is, there's a meshpoint, X_{N_0}, at the interface. Find the ODEs for

$$dU_1/dt, \quad dU_{N_0-1}/dt, \quad dU_{N_0+1}/dt, \quad dU_{N-1}/dt.$$

In doing so, make it clear how you used *both* of the boundary conditions at $x = 1$. (The ODEs at the other meshpoints are the usual ones, just modified to take into account whether you're using D_1 or D_2.)

(c) Take $D_1 = 1/10$, $D_2 = 2$, $u_L = 1$, and $u_R = 5$ and consider the initial data $u_0(x) = 1 + 2x$. Modify your code from problem 3 to solve the initial data problem, computing up to time $T = 1$.

You're doing explicit time-stepping and so you'll need to choose the time-step k by fixing a value for λ. Let's take $\lambda = 1/4$. But how should you *define* λ? Should you take $\lambda = D_1 k/h^2$ or $\lambda = D_2 k/h^2$? Try both and see what happens.

Now that you've chosen how to define λ, compute up to time $T = 1$ using $(N, M) = (10, 200)$, $(N, M) = (20, 800)$, $(N, M) = (40, 3200)$, and $(N, M) = (80, 12800)$ to create approximate solutions u_1, u_2, u_3, and u_4. From part a) you know what the steady state is and so you know the value of $u_\infty(1)$: the value of the steady state at the interface.

i. On one graph, plot $u_1(1,t) - u_\infty(1)$, $u_2(1,t) - u_\infty(1)$, $u_3(1,t) - u_\infty(1)$, and $u_4(1,t) - u_\infty(1)$ as functions of time.

ii. In a table, present the values for $u_1(1,T) - u_\infty(1)$, $u_2(1,T) - u_\infty(1)$, $u_3(1,T) - u_\infty(1)$, and $u_4(1,T) - u_\infty(1)$.

iii. Also present the values for $(u_1(1,T) - u_\infty(1))/(u_2(1,T) - u_\infty(1))$, $(u_2(1,T) - u_\infty(1))/(u_3(1,T) - u_\infty(1))$, and $(u_3(1,T) - u_\infty(1))/(u_4(1,T) - u_\infty(1))$.

iv. Finally, present the values for $(u_1(1,T) - u_2(1,T))/(u_2(1,T) - u_3(1,T))$ and $(u_2(1,T) - u_3(1,T))/(u_3(1,T) - u_4(1,T))$.