MATH 333: Partial Differential Equations

Problem Set 10, Final version

Due Date: Wed., Nov. 15, 2006

1. Note: This is a scaled-back version of parts (a)–(c) from Project 10. I feel secure including this problem here since no one has expressed an interest in Project 10. Should anyone want to do Project 10 now, it will be worth at least half a point less than the (≥ 2) originally indicated.

   There are infinitely many monomials of the form x^n, $n = 0, 1, 2, \ldots$. Likewise, there are infinitely many *Legendre polynomials* $P_n(x)$, $n = 0, 1, 2, \ldots$. The first five Legendre polynomials are

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= \frac{1}{2}(3x^2 - 1), \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x), \\ P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3). \end{aligned}$$

   The Legendre polynomials have the advantage over basic monomials that they are mutually orthogonal under the $L^2(-1,1)$ inner product. (You may take this mutual orthogonality as fact. For the assertion that the monomials are not, in general, orthogonal under this inner product, see Exam 1.)

   (a) Show that $\text{span}\{P_0, P_1, P_2, P_3, P_4\} = \text{span}\{1, x, x^2, x^3, x^4\}$. As a guide to getting started, this entails showing that every function that can be written as a linear combination of the $P_0, \ldots P_4$ may also be written as a linear combination of 1, x, $\ldots$, x^4, and vice versa.

   (b) Find the polynomial that is "closest" among all 4th degree polynomials, as measured by the $L^2(-1,1)$-norm, to the function $1/(1+x^2)$ on $[-1,1]$. Also, write out the 4th degree Taylor polynomial of $1/(1+x^2)$ at $x = 0$, and graph all three functions in $[-1,1]$. Give a comparison of their "closeness".

2. (a) Review (if necessary) the method of characteristics (Section 1.4.1) for solving the PDE (with constant c)

$$u_t + cu_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \qquad u(x,0) = f(x), \quad x \in \mathbb{R}.$$

   Then, use Duhamel's principle to solve

$$u_t + cu_x = f(x,t), \quad x \in \mathbb{R}, \quad t > 0, \qquad u(x,0) = 0, \quad x \in \mathbb{R}.$$

(b) Solve

$$u_t + cu_x = f(x,t), \quad x \in \mathbb{R}, \quad t > 0, \qquad u(x,0) = \phi(x), \quad x \in \mathbb{R},$$

by first splitting the problem in two and then combining the two solutions appropriately. Compare your answer to the one given in Section 1.4.2.

3. Write a formula for the solution to the Cauchy wave problem

$$\begin{aligned} u_{tt} - c^2 u_{xx} &= \sin x, \qquad x \in \mathbb{R}, \quad t > 0, \\ u(x,0) = u_t(x,0) &= 0, \quad x \in \mathbb{R}. \end{aligned}$$

For reference, see problem 3 in Problem Set 2. Challenge: Can you plot the solution surface (it is a surface, not a curve) for $c = 1$ in OCTAVE? Some helpful commands for doing so are `meshgrid` and `mesh`.

4. Do Exercise 4.9. Part (a) entails writing out how you initialize the v_j^0, $j = 0, 1, \ldots, n+1$, and the matrix-vector computation that produces the v_j^{m+1} from the v_j^m, $j = 0, 1, \ldots, n+1$, $m = 0, 1, 2, \ldots$. As we already did most of this in class on a more general problem, your main task will be to make things specific to the IBVP of this exercise. When we derived an implicit scheme in class, our matrix formula did not carry with it enough generality to cover the IBVP here, so you will need to do (b) as if from scratch.

For part (d), I would like your answers to include both graphs like those of Figures 4.2–4.5 (i.e., ones that contain exact and finite-difference solutions) and comments about issues related to getting reasonably accurate finite-difference solutions. (For instance, in Example 4.2 it is necessary to keep the ratio $r = \Delta t/(\Delta x)^2 \leq 1/2$.) For the explicit scheme, you are welcome to use the set of OCTAVE scripts I used for demonstration in class downloadable from
http://www.calvin.edu/ scofield/courses/m333/materials/octave/finiteDiffScheme1/
as a point of departure. (As with the OCTAVE scripts from the previous problem set, place these in your OCTAVE working directory.) The main script is `explicitSolver.m` in which I have included many comments to explain various steps. You can get a general overview of how the script works by typing `help explicitSolver` at the OCTAVE prompt. Since your IBVP has constant coefficients, I would like to see you replace calls (in the code you submit with homework) to functions like `u_xxCoeff.m` with simpler code, and perhaps take advantage of these constant coefficients when building the matrix `A_t`. Code modifications necessary to implement your implicit scheme will be somewhat more substantial. One reminder is that it is more efficient to solve a matrix system $A\boldsymbol{x} = \boldsymbol{b}$ for $\boldsymbol{x}$ (A and $\boldsymbol{b}$ known) via Gaussian elimination than by inverting A. In OCTAVE one can use the command

```
octave:1> x = A \ b;
```

to implement Gaussian elimination and store it in the variable `x`.