Homework 2

(Due Friday, February 22, 2010)

1. Use the "movie" function in matlab to visualize the dispersion and dissipation of the wave solutions to $L_i w = 0$ for the three partial differential operators discussed in class:
$$L_1 u = \partial_t u + a\partial_x u$$
$$L_2 u = \partial_t u + a\partial_x u - D\partial_x^2 u, \text{ for } D > 0$$
$$L_3 u = \partial_t u + a\partial_x u - \mu\partial_x^3 u.$$
Note that this does not require you to numerically solve the PDEs! You can simply graph a wave-like solution, $w(x,t) = A_0 e^{i(kx-\omega(k)t)}$, or a superposition of such solutions at various values of t. An important part of this problem is choosing a suitable domain (in both time and space) and values of k, D, and μ so that your visualization clearly shows what you intend.
For this problem, you will "hand in" your matlab code by emailing me a *single* M-file along with the command to set it running. If you choose to not use matlab, please discuss your plans with me, so that we can be sure I will be able to grade your work.
2. Investigate the numerical dispersion and dissipation for the First-Order Upwind (discussed in class) and Crank-Nicholson discretizations of $u_t + au_x = 0$ for $a > 0$. For Crank-Nicholson, analyze the expected dissipation and dispersion by hand, and explain what you expect to see in a numerical study. In particular, graph the expected amount of dissipation and the phase velocity, v_{ph} as a function of k (or of kh_x) for these two schemes and discuss the results.
Then, use your codes for these two schemes from HW1 to get numerical results that confirm your expectations. Choose the initial and boundary conditions, $u_0(x)$ and $u_1(x)$, to match a single wave-like solution $e_{j,l} = e_0 e^{i(kjh_x - \omega l h_t)}$ or a superposition of such solutions, and discuss the results. Again, an important part of this problem is the appropriate choice of the parameters, a, k, h_x, and h_t, as well as an appropriate visualization of the results. You may want to use the movie function again, or you can use matlab's "surf" function to display your solution as a surface in space and time. Be sure to label all axes appropriately.