Solutions to Assignment #9

Total of 70 points.

1 (# 2.10 in M&M 2nd ed). (*15 points*)

> This problem exercises your ability to be organized and not lose your head. It is a truly impressive calculus exercise. On the other hand, you are doing something that was probably a research paper in 1950.

The goal is to solve the simplest heat equation $\tilde{u}_t = \tilde{u}_{xx}$ for $\tilde{u}(x,t)$ on an unequally-spaced grid by transforming the spatial variable and using an equally-spaced grid in the transformed variable. I am interested in this process anyway, and I found it convenient to generalize the process slightly. Therefore I define

$$x = \varphi(s), \qquad \text{with inverse function } s = \psi(x),$$

to be a differentiable and increasing function which maps the interval $[0,1]$ to itself. Of course $x = \varphi(\psi(x))$ and $s = \psi(\varphi(s))$. Exercise 2.10 is about the case $\varphi(s) = s^2$, $\psi(x) = \sqrt{x}$. Let $u(s,t) = \tilde{u}(\varphi(s),t) = \tilde{u}(x,t)$; we are describing a new function of two variables s and t, but of course it is "the same function as" the original function of x and t. The grid is

$$s_j = j\Delta s \quad \text{with} \quad \Delta s = \frac{1}{J},$$

which gives an unequally-spaced grid $x_j = \varphi(s_j)$ in the x variable. In exercise 2.10, $x_j = j^2/J^2$.

The chain rule allows us to replace x derivatives of any function $f(x,t)$ by s derivatives times functions of s:

$$f_x = \frac{\partial f}{\partial s}\frac{\partial s}{\partial x} = \frac{\partial f}{\partial s}\psi'(x) = \psi'(\varphi(s))\, f_s.$$

Let

$$\nu(s) = \psi'(\varphi(s))$$

as this function will appear frequently; check that $\nu(s) = 1/(2s)$ in exercise 2.10. It follows that $u_x = \nu(s)\, u_s$. The heat equation $u_t = u_{xx}$ becomes

$$u_t = \nu(s)\Big(\nu(s)\, u_s\Big)_s \tag{1}$$

Note that this can also be written $u_t = \nu(s)^2 u_{ss} + \nu(s)\nu'(s) u_s$, a form for later use.

We are going to apply an equally-spaced finite difference method to equation (1). In particular, the staggered-grid method of exercise 2.7 adapts easily to this case:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \nu(s_j)\, \frac{\left\{\nu(s_{j+1/2})\left(U_{j+1}^n - U_j^n\right) - \nu(s_{j-1/2})\left(U_j^n - U_{j-1}^n\right)\right\}}{\Delta s^2}, \tag{2}$$

where $U_j^n \approx u(s_j, t_n) = \tilde{u}(x_j, t_n)$.

We are tasked to compute the truncation error of method (2). Let $u(s,t)$ be the exact solution to (1), as usual. I will shorten notation by writing $u, u_t, u_s, u_{ss}, u_{sss}$ for these derivatives evaluated at the point (s_j, t). Also I will write Δ for Δs. Here are some Taylor expansions (with remainder):

$$\begin{aligned}
u(s_j, t_n + \Delta t) - u &= \Delta t\, u_t + \frac{1}{2}\Delta t^2 u_{tt}(s_j, \tau) \\
u(s_j + \Delta, t_n) - u &= \Delta\, u_s + \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} + \frac{1}{24}\Delta^4 u_{ssss}(\sigma_1, t_n) \\
u - u(s_j - \Delta, t_n) &= \Delta\, u_s - \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} - \frac{1}{24}\Delta^4 u_{ssss}(\sigma_2, t_n)
\end{aligned}$$

Also, recalling the solution of exercise 2.7, and letting $\nu = \nu(s_j)$,

$$\begin{aligned}
\nu(s_j + \Delta/2) &= \nu + (\Delta/2)\,\nu'(\sigma_3) \\
\nu(s_j - \Delta/2) &= \nu - (\Delta/2)\,\nu'(\sigma_4)
\end{aligned} \tag{3}$$

It follows that:

$$\begin{aligned}
&\nu(s_j)\,\frac{\{\nu(s_{j+1/2})\,(u(s_{j+1}, t_n) - u) - \nu(s_{j-1/2})\,(u - u(s_{j-1}, t_n))\}}{\Delta s^2} \\
&\quad = \frac{\nu}{\Delta^2}\Bigg\{(\nu + (\Delta/2)\,\nu'(\sigma_3))\left(\Delta\, u_s + \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} + \frac{1}{24}\Delta^4 u_{ssss}(\sigma_1, t)\right) \\
&\qquad\qquad - (\nu - (\Delta/2)\,\nu'(\sigma_4))\left(\Delta\, u_s - \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} - \frac{1}{24}\Delta^4 u_{ssss}(\sigma_2, t)\right)\Bigg\} \\
&\quad = \frac{\nu}{\Delta^2}\left\{\nu\Delta^2 u_{ss} + \nu\frac{1}{24}\Delta^4 u_{ssss}(\sigma_1, t) + \nu\frac{1}{24}\Delta^4 u_{ssss}(\sigma_2, t)\right\} \\
&\qquad + \frac{\nu}{\Delta^2}\Bigg\{(\Delta/2)\,\nu'(\sigma_3)\left(\Delta\, u_s + \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} + \frac{1}{24}\Delta^4 u_{ssss}(\sigma_1, t)\right) \\
&\qquad\qquad + (\Delta/2)\,\nu'(\sigma_4)\left(\Delta\, u_s - \frac{1}{2}\Delta^2 u_{ss} + \frac{1}{6}\Delta^3 u_{sss} - \frac{1}{24}\Delta^4 u_{ssss}(\sigma_2, t)\right)\Bigg\} \\
&\quad = \nu^2 u_{ss} + \frac{\nu^2\Delta^2}{12}\left(\frac{u_{ssss}(\sigma_1, t_n) + u_{ssss}(\sigma_2, t_n)}{2}\right) \\
&\qquad + \nu\left(\frac{\nu'(\sigma_3) + \nu'(\sigma_4)}{2}\right)\left\{u_s + \frac{1}{6}\Delta^2 u_{sss}\right\} + \frac{\nu\Delta}{4}\Big(\nu'(\sigma_3) - \nu'(\sigma_4)\Big)\,u_{ss} \\
&\qquad + \frac{\nu\Delta^3}{48}\Big(\nu'(\sigma_3)u_{ssss}(\sigma_1, t_n) - \nu'(\sigma_4)u_{ssss}(\sigma_2, t_n)\Big)
\end{aligned}$$

At this point we see that we should have gone further in our Taylor expansions in (3). Let $\nu' = \nu'(s_j)$. Note that going one term further is possible without repeating work, as it is equivalent to the replacements

$$\nu'(\sigma_3) = \nu' + \frac{\Delta}{4}\nu''(\sigma_5), \qquad \nu'(\sigma_4) = \nu' - \frac{\Delta}{4}\nu''(\sigma_6)$$

so that

$$\frac{\nu'(\sigma_3)+\nu'(\sigma_4)}{2} = \nu' + \frac{\Delta}{8}\Big(\nu''(\sigma_5)-\nu''(\sigma_6)\Big)$$
$$\nu'(\sigma_3)-\nu'(\sigma_4) = \frac{\Delta}{2}\left(\frac{\nu''(\sigma_5)+\nu''(\sigma_6)}{2}\right) = \frac{\Delta}{2}\,\nu''(\sigma_7)$$

Also

$$\frac{u_{ssss}(\sigma_1,t_n)+u_{ssss}(\sigma_2,t_n)}{2} = u_{ssss}(\sigma_8,t_n).$$

(Twice above, and several more times below without comment, I used (will use) the Intermediate Value Theorem.) Then

$$\begin{aligned}
&\nu(s_j)\,\frac{\big\{\nu(s_{j+1/2})\,(u(s_j+\Delta,t_n)-u)-\nu(s_{j-1/2})\,(u-u(s_j-\Delta,t_n))\big\}}{\Delta^2}\\
&\quad= \nu^2 u_{ss} + \frac{\nu^2\Delta^2}{12}u_{ssss}(\sigma_8,t_n) + \nu\left(\nu' + \frac{\Delta}{8}\Big(\nu''(\sigma_5)-\nu''(\sigma_6)\Big)\right)\left\{u_s + \frac{1}{6}\Delta^2 u_{sss}\right\}\\
&\qquad+\frac{\nu\Delta}{4}\Big(\frac{\Delta}{2}\,\nu''(\sigma_7)\Big)\,u_{ss} + o(\Delta^2)\\
&\quad= \nu^2 u_{ss} + \nu\nu' u_s + \frac{\nu^2\Delta^2}{12}u_{ssss}(\sigma_8,t_n) + \nu\nu'\frac{1}{6}\Delta^2 u_{sss} + \nu\frac{\Delta}{8}\Big(\nu''(\sigma_5)-\nu''(\sigma_6)\Big)u_s\\
&\qquad+\frac{\nu\Delta^2}{8}\,\nu''(\sigma_7)\,u_{ss} + o(\Delta^2)
\end{aligned}$$

Again we need to one term further in the ν expansions. It is equivalent to the replacements

$$\nu''(\sigma_5) = \nu'' + \frac{\Delta}{6}\nu'''(\sigma_9), \qquad\qquad \nu''(\sigma_6) = \nu'' - \frac{\Delta}{6}\nu'''(\sigma_{10})$$

so $\nu''(\sigma_5)-\nu''(\sigma_6) = (\Delta/3)(1/2)\,(\nu'''(\sigma_9)+\nu'''(\sigma_{10})) = (\Delta/3)\nu'''(\sigma_{11})$. Finally

$$\begin{aligned}
&\nu(s_j)\,\frac{\big\{\nu(s_{j+1/2})\,(u(s_j+\Delta,t_n)-u)-\nu(s_{j-1/2})\,(u-u(s_j-\Delta,t_n))\big\}}{\Delta^2}\\
&\quad= \nu^2 u_{ss} + \nu\nu' u_s + \frac{\nu^2\Delta^2}{12}u_{ssss}(\sigma_8,t_n) + \nu\nu'\frac{1}{6}\Delta^2 u_{sss} + \nu\frac{\Delta^2}{24}\,\nu'''(\sigma_{11})u_s\\
&\qquad+\frac{\nu\Delta^2}{8}\,\nu''(\sigma_7)\,u_{ss} + o(\Delta^2)\\
&\quad= \nu^2 u_{ss} + \nu\nu' u_s + \frac{\nu^2\Delta^2}{12}u_{ssss} + \nu\nu'\frac{1}{6}\Delta^2 u_{sss} + \nu\frac{\Delta^2}{24}\,\nu''' u_s\\
&\qquad+\frac{\nu\Delta^2}{8}\,\nu''\,u_{ss} + o(\Delta^2)\\
&\quad= \nu^2 u_{ss} + \nu\nu' u_s + \frac{\nu\Delta^2}{24}\,(2\nu u_{ssss} + 4\nu' u_{sss} + 3\,\nu''\,u_{ss} + \nu''' u_s) + o(\Delta^2)
\end{aligned}$$

At the second-to-last stage I have replaced each of σ_7, σ_8, and σ_{11} by s_j at cost of an overall $o(\Delta^2)$ term.

Now to actually get the truncation error:

$$\frac{u(s_j,t+\Delta t)-u}{\Delta t} - \nu(s_j)\,\frac{\left\{\nu(s_{j+1/2})\,(u(s_j+\Delta,t)-u)-\nu(s_{j-1/2})\,(u-u(s_j-\Delta,t))\right\}}{\Delta^2}$$
$$\begin{aligned}
&= u_t - \nu^2 u_{ss} - \nu\nu' u_s + \frac{1}{2}\Delta t u_{tt}(s,\tau) \\
&\qquad - \frac{\nu\Delta^2}{24}\left(2\nu u_{ssss} + 4\nu' u_{sss} + 3\,\nu''\,u_{ss} + \nu''' u_s\right) + o(\Delta^2) \\
&= \frac{1}{2}\Delta t^2 u_{tt}(s,\tau) - \frac{\nu\Delta^2}{24}\left(2\nu u_{ssss} + 4\nu' u_{sss} + 3\,\nu''\,u_{ss} + \nu''' u_s\right) + o(\Delta^2)
\end{aligned}$$

I have used the fact that $u(s,t)$ solves the PDE, namely $u_t = \nu(s)^2 u_{ss} + \nu(s)\nu'(s) u_s$.

It is an easy exercise to check that for any two functions p and q of s:

$$(p\,q''')' + (p\,q')''' = 2p\,q'''' + 4p'\,q''' + 3p''\,q'' + p'''\,q'$$

Therefore

$$T(s_j,t_n) = \frac{1}{2}\Delta t u_{tt} - \frac{\nu\Delta^2}{24}\Big(\,(\nu\,u_{sss})_s + (\nu\,u_s)_{sss}\Big) + o(\Delta^2) \tag{4}$$

as claimed in the statement of exercise 2.10; recall $\nu(s) = 1/(2s)$ there.

Now, it remains to transform the s derivatives, in expression (4) for the truncation error, into x derivatives. Note

$$s = \psi(x) \quad\implies\quad 1 = \psi'(x)\varphi'(s) = \nu(s)\varphi'(s)$$

so we can write, after some simplifications,

$$\begin{aligned}
u_s &= \frac{u_x}{\nu} = \varphi' u_x, \\
u_{ss} &= \varphi'' u_x + (\varphi')^2 u_{xx}, \\
u_{sss} &= \varphi''' u_x + 3\varphi''\varphi' u_{xx} + (\varphi')^3 u_{xxx}, \\
u_{ssss} &= \varphi'''' u_x + \left(4\varphi'''\varphi' + 3(\varphi'')^2\right) u_{xx} + 6\varphi''(\varphi')^2 u_{xxx} + (\varphi')^4 u_{xxxx}
\end{aligned}$$

Therefore, after some simplification,

$$\begin{aligned}
(\nu\,u_{sss})_s + (\nu\,u_s)_{sss} &= 2\nu u_{ssss} + 4\nu' u_{sss} + 3\,\nu''\,u_{ss} + \nu''' u_s \\
&= 2\nu(\varphi')^4 u_{xxxx} + \left[12\nu\varphi''(\varphi')^2 + 4\nu'(\varphi')^3\right] u_{xxx} \\
&\qquad + \left[8\nu\varphi''\varphi' + 6\nu(\varphi')^2 + 12\nu'\varphi''\varphi' + 3\nu''(\varphi')^2\right] u_{xx} \\
&\qquad + \left[2\nu\varphi'''' + 4\nu'\varphi''' + 3\nu''\varphi'' + \nu'''\varphi'\right] u_x
\end{aligned}$$

In the case $x = s^2$ of exercise 2.10 we have $\varphi = s^2$, $\varphi' = 2s$, $\varphi'' = 2$, $\varphi''' = 0$, and $\varphi'''' = 0$. Also $\nu = 1/(2s)$, $\nu' = -1/(2s^2)$, $\nu'' = 1/(s^3)$, and $\nu''' = -3/(s^4)$. It follows that, after simplification,

$$(\nu\,u_{sss})_s + (\nu\,u_s)_{sss} = 16s^3\,u_{xxxx} + 32s\,u_{xxx} + 0\,u_{xx} + 0\,u_x.$$

(I think this is telling us that the transformation $x = s^2$ is rather special among such transformations.) Thus, from (4),

$$\begin{aligned} T_j^n &= \frac{1}{2}\Delta t u_{tt} - \frac{\nu(\Delta s)^2}{24}\Big(\left(\nu\, u_{sss}\right)_s + \left(\nu\, u_s\right)_{sss} \Big) + o((\Delta s)^2) \\ &= \frac{1}{2}\Delta t u_{tt} - \frac{(\Delta s)^2}{48s}\Big(16s^3\, u_{xxxx} + 32s\, u_{xxx}\Big) + o((\Delta s)^2) \\ &= \frac{1}{2}\Delta t u_{tt} - (\Delta s)^2\Big(\frac{1}{3}\, x\, u_{xxxx} + \frac{2}{3}\, u_{xxx}\Big) + o((\Delta s)^2) \end{aligned} \tag{5}$$

exactly as claimed in exercise 2.10.

One more matter remains. We are asked to compare (5) with the truncation error found by working with an arbitrary, unequally-spaced mesh $\{x_j\}$ in exercise 2.3. There the formula for truncation error was

$$\begin{aligned} T_j^n = \frac{1}{2}\Delta t u_{tt} &- \frac{1}{3}(\Delta x_j - \Delta x_{j-1})u_{xxx} \\ &- \frac{1}{12}\left[(\Delta x_j)^2 + (\Delta x_{j-1})^2 - \Delta x_j \Delta x_{j-1}\right] u_{xxxx} + (\text{higher order}) \end{aligned}$$

where $\Delta x_j = x_{j+1} - x_j$.

For the grid $x_j = j^2/J^2$ in exercise 2.10,

$$\Delta x_j = \frac{2j+1}{J^2}, \qquad \Delta x_{j-1} = \frac{2j-1}{J^2}.$$

Thus the formula from exercise 2.3 becomes, after simplification, and recalling $\Delta s = 1/J$,

$$\begin{aligned} T_j^n &= \frac{1}{2}\Delta t u_{tt} - \frac{1}{3}\left(\frac{2}{J^2}\right) u_{xxx} - \frac{1}{12}\left[\frac{4j^2+3}{J^4}\right] u_{xxxx} + (\text{higher order}) \\ &= \frac{1}{2}\Delta t u_{tt} - \frac{2}{3}(\Delta s)^2 u_{xxx} - \frac{1}{12}\left[4x_j(\Delta s)^2 + 3(\Delta s)^4\right] u_{xxxx} + (\text{higher order}). \end{aligned}$$

Indeed this agrees with (5) to order $(\Delta s)^2$.

We are done with exercise 2.10!

2 (# 4.4 in M&M 2nd ed). *(15 points)*

Again this problem exercises your ability to be organized and concise. The lop-sided Lax-Wendroff formula you are addressing here answers a question you should have had: *How to implement Lax-Wendroff at a boundary where the characteristics are heading outward?* This problem also can be regarded as describing *second order upwinding*, although, because the coefficients d_{-2}, d_{-1}, d_0 below are not all positive, we lose the maximum principle property which holds for first order upwinding.

The problem statement is perhaps not clear enough. I hope that the version I gave in class helped. Note that I interpret the phrase "*... for the solution of the equation $u_t + au_x = 0$ agrees with the Taylor series expansion of $u(x_j, t_{n+1})$ to as high an order as possible when a is a positive constant*" to mean that we want to expand in Taylor series around the point (x_j, t_n) and use the PDE to turn the time partial derivatives in the Taylor expansion of $u(x_j, t_{n+1})$ into space derivatives.

First note these consequences of the PDE $u_t + au_x = 0$: $u_t = -au_x$, $u_{tt} = a^2 u_{xx}$. Let $\nu = a\Delta t/\Delta x$ as before. To simplify Taylor expansions, let $u = u(x_j, t_n)$, $u_t = u_t(x_j, t_n)$, $u_x = u_x(x_j, t_n)$, etc. Then

$$\begin{aligned}
u(x_j, t_{n+1}) &= u + \Delta t\, u_t + \frac{\Delta t^2}{2}\, u_{tt} + O(\Delta t^3) = u - a\Delta t\, u_x + \frac{a^2\Delta t^2}{2}\, u_{xx} + O(\Delta t^3) \\
u(x_{j-1}, t_n) &= u - \Delta x\, u_x + \frac{\Delta x^2}{2}\, u_{xx} + O(\Delta x^3) \\
u(x_{j+1}, t_n) &= u + \Delta x\, u_x + \frac{\Delta x^2}{2}\, u_{xx} + O(\Delta x^3).
\end{aligned}$$

We want

$$-u(x_j, t_{n+1}) + c_{-1}u(x_{j-1}, t_n) + c_0 u + c_{+1}u(x_{j+1}, t_n) = 0 + O(\Delta t^3) + O(\Delta x^3).$$

That is, we have these equations:

$$\begin{aligned}
u: &\qquad -1 + c_{-1} + c_0 + c_1 = 0 \\
u_x: &\qquad a\Delta t - \Delta x\, c_{-1} + \Delta x\, c_{+1} = 0 \\
u_{xx}: &\qquad -\frac{a^2\Delta t^2}{2} + \frac{\Delta x^2}{2}\, c_{-1} + \frac{\Delta x^2}{2}\, c_{+1} = 0
\end{aligned}$$

Rewritten using ν we have these three equations:

$$c_{-1} + c_0 + c_1 = 1, \qquad -c_{-1} + c_1 = -\nu, \qquad c_{-1} + c_1 = \nu^2.$$

On the one had, the matrix for these three equations is invertible so there is only one solution. On the other hand the coefficients in (4.36) in the text are a solution to these three equations. Thus we have again derived Lax-Wendroff.

Next we repeat the same story with different details. First add this Taylor expansion to the above list: $u(x_{j-2}, t_n) = u - 2\Delta x\, u_x + 2\Delta x^2\, u_{xx} + O(\Delta x^3)$. We want

$$-u(x_j, t_{n+1}) + d_{-2}u(x_{j-2}, t_n) + d_{-1}u(x_{j-1}, t_n) + d_0 u = 0 + O(\Delta t^3) + O(\Delta x^3)$$

so we have these equations:

$$\begin{aligned}
-1 + d_{-2} + d_{-1} + d_0 &= 0 \\
a\Delta t - 2\Delta x\, d_{-2} - \Delta x\, d_{-1} &= 0 \\
-\frac{a^2\Delta t^2}{2} + 2\Delta x^2\, d_{-2} + \frac{\Delta x^2}{2}\, d_{-1} &= 0
\end{aligned}$$

or

$$d_{-2} + d_{-1} + d_0 = 1, \qquad 2d_{-2} + d_{-1} = \nu, \quad 4d_{-2} + d_{-1} = \nu^2.$$

The solution gives the formula

$$U_j^{n+1} = \frac{1}{2}\nu(\nu - 1)U_{j-2}^n + \nu(2 - \nu)U_{j-1}^n + \left(1 - \frac{3}{2}\nu + \frac{1}{2}\nu^2\right) U_j^n. \tag{6}$$

Equation (6) is indeed what one gets by replacing $j \mapsto j-1$ *and* $\nu \mapsto \nu-1$ on the right of (4.36). The question Morton & Mayers want you to answer now is *why* these replacements are the right ones to give equation (6). And they want you to find the stability condition.

The transformation "$\xi = x - (\Delta x/\Delta t)t$" is suggested. This requires interpretation, in part because the authors are guilty of using a common but lazy convention of only telling you the interesting part of the transformation.

Also, it helps to think in terms of the stencils of the two methods. So, *draw the stencils for* (4.36) *and* (6) *to follow the next paragraph.*

The intended transformation

$$(x,t) \mapsto (\xi, t')$$

is precisely stated by

$$\xi = x - \frac{\Delta x}{\Delta t}t \qquad \text{and} \qquad t' = t.$$

The inverse transformation is: $x = \xi + (\Delta x/\Delta t)t$, $t = t'$. (It is often useful to write down both directions right away.) We want to know how derivatives change, so we apply the chain rule:

$$u_x = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial t'}\frac{\partial t'}{\partial x} = u_\xi$$
$$u_t = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial t} + \frac{\partial u}{\partial t'}\frac{\partial t'}{\partial t} = -\frac{\Delta x}{\Delta t}u_\xi + u_{t'}.$$

Thus the original PDE $u_t + au_x = 0$ becomes

$$u_t + \left(a - \frac{\Delta x}{\Delta t}\right)u_\xi = 0. \tag{7}$$

I have dropped the prime from the new (but identical) time variable. Note that the characteristics of $u_t + au_x = 0$ have slope a while the characteristics of equation (7) have slope $a - (\Delta x/\Delta t)$.

Now for the *punchline*: Consider the stencils of (4.36) and (6) when $a > 0$. In their respective stencils, the characteristic of $u_t + au_x = 0$ passing through (x_j, t_{n+1}) crosses the line $t = t_n$ at the "same point" in the interval $[x_{j-2}, x_j]$ as the characteristic of (7) through (x_j, t_{n+1}) crosses the line $t = t_n$ in the interval $[x_{j-1}, x_{j+1}]$. In particular,

$$\left(a - \frac{\Delta x}{\Delta t}\right)\Delta t + \Delta x = a\Delta t.$$

These are the distances from the rightmost point in the stencil at time $t = t_n$ to the place where the characteristic crosses $t = t_n$, for (6) applied to $u_t + au_x = 0$ and for (4.36) applied to (7), respectively.

We therefore note that the CFL condition for (6) applied to $u_t + au_x = 0$ is

$$a\Delta t \le 2\Delta x \qquad \text{or} \qquad \nu \le 2$$

if $a > 0$ and $\nu = a\Delta t/\Delta x$. As we tend to believe that CFL is a good guide for stability of Lax-Wendroff, we can guess that this condition is sufficient.

To check that "$\nu \le 2$" is the stability condition, we go back to (6) and actually do a von Neumann analysis. This would be very awkward if done directly on (6), but the point is that we know (6) is the $j \mapsto j-1$ and $\nu \mapsto \nu - 1$ modification of (4.36). It follows that if $U_j^n = \lambda^n e^{ik(j\Delta x)}$ is substituted into (6) we get

$$\lambda(k) = e^{-ik\Delta x}\left(1 - i(\nu-1)\sin k\Delta x - 2(\nu-1)^2 \sin^2(\frac{1}{2}k\Delta x)\right) \tag{8}$$

instead of (4.38) exactly because the transformations $j \mapsto j-1$ and $\nu \mapsto \nu-1$ take (4.36) to equation (6). In particular, the overall phase factor "$e^{-ik\Delta x}$" comes from the $j \mapsto j-1$ part of the tranformation. Taking absolute values and following the same argument as in (4.39), we see that the method is stable if and only if

$$|\nu - 1| \le 1$$

which is $\nu \le 2$ because we are only considering the $a > 0$ case.

3. (*20 points*) I wrote the following code:

```
function uu = diffusion(J);
% DIFFUSION  Solves a particular nonhomogeneous diffusion problem, with
% nonconstant diffusivity, on a square:
%    div ( a(x,y) grad u ) + f(x,y) = 0
% with u = 0 on the sides of the square [-1,1]x[-1,1].  Uses the finite
% difference method (6.29) in Morton & Mayers 2nd ed.
% ELB 5/8/07

dx = 2/J;
x = -1:dx:1;  y = x;              % regular grid
xs = -1+(dx/2):dx:1-(dx/2);  ys = xs;   % staggered grid
uu = zeros(J+1,J+1);              % will hold solution

A = sparse((J-1)^2,(J-1)^2);
b = zeros((J-1)^2,1);             % not sparse; no need for sparse storage
for r=1:J-1                       % r is interior x-direction local index
    for s=1:J-1                   % s is interior y-direction local index
        % inefficiency here: a(x,y) recomputed each time when needed
        aw = a(xs(r),y(s+1));
        ae = a(xs(r+1),y(s+1));
        as = a(x(r+1),ys(s));
        an = a(x(r+1),ys(s+1));
        k = (s-1) * (J-1) + r;    % 1 <= k <= (J-1)^2;  global index; row #
        A(k,k) = - (aw + ae + as + an);  % diagonal entry
        % for next four cases: only insert off-diag if away from bdry
        if r > 1,  A(k,(s-1) * (J-1) + (r-1)) = aw;  end
        if r < J-1,  A(k,(s-1) * (J-1) + (r+1)) = ae;  end
        if s > 1,  A(k,(s-2) * (J-1) + r) = as;  end
        if s < J-1,  A(k,(s) * (J-1) + r) = an;  end
        b(k) = - (dx^2) * f(x(r+1),y(s+1));
    end
end
```

```
% % diagnostics; uncomment selectively
% full(A), b
% figure(1), spy(A)
% return

%tic
v = A \ b;                          % solve sparse system!
%time=toc

uu(2:J,2:J) = reshape(v,J-1,J-1)'; % turn global vector into x,y array

% fancy plot
subplot(3,2,1:4)
surf(x,x,uu,'LineStyle','none','FaceColor','interp')
xlabel x, ylabel y, zlabel u
title('solution u(x,y) of  \nabla \cdot ( a(x,y) \nabla u ) + f(x,y) = 0')
[xx,yy] = meshgrid(x,y);
subplot(3,2,5)
surf(x,y,a(xx,yy),'LineStyle','none')
xlabel x, ylabel y, title('diffusivity a(x,y)')
subplot(3,2,6)
mesh(x,y,f(xx,yy))
xlabel x, ylabel y, title('source f(x,y)')

function z = a(x,y)
z = 0.7 + (x - y)./(1 + (x - y).^2);

function z = f(x,y)
z = zeros(size(x));
z(((x+0.5).^2 + (y+0.5).^2) < 0.1) = 1;
```

It produced figure 1 when $J = 100$. The result is a nonnegative smooth function $u(x, y)$ which reaches its maximum of about 0.108 near $(x, y) = (-.55, -.35)$. The function goes to $u = 0$ along the edge of the square as it should. This function could be interpreted as the equilibrium (i.e. steady state) temperature of a plate with zero temperature along the boundary and a heat source only in a disc of radius $\sqrt{0.1}$ centered at $(-.5, -.5)$. The heat source is symmetric around the line $x = y$, and the square region has a reflection symmetry across that line, but the solution does not.

But the diffusivity $a(x, y)$ does not have that symmetry! And the peak of the temperature is on the side of $x = y$ where the diffusivity is *lowest*, as shown in the subplot of $a(x, y)$. The reason the peak happens where diffusivity is lowest is that the heat produced by $f(x, y)$ tends to build up on the low diffusivity side of $x = y$, and it doesn't "get to" the edge of the plate where heat is being carried away by a refrigerator.

The $J = 20$ ($\Delta x = \Delta y = 0.1$) case ran in much less than one second while the $J = 200$ ($\Delta x = \Delta y = 0.01$) case ran in much more than one minute. This is not too surprising, however, because the dimension of the linear system being solved is $N = (J - 1)^2$, which increased by a factor of 100 from $N = 19^2 = 361$ to $N = 199^2 = 39601$.

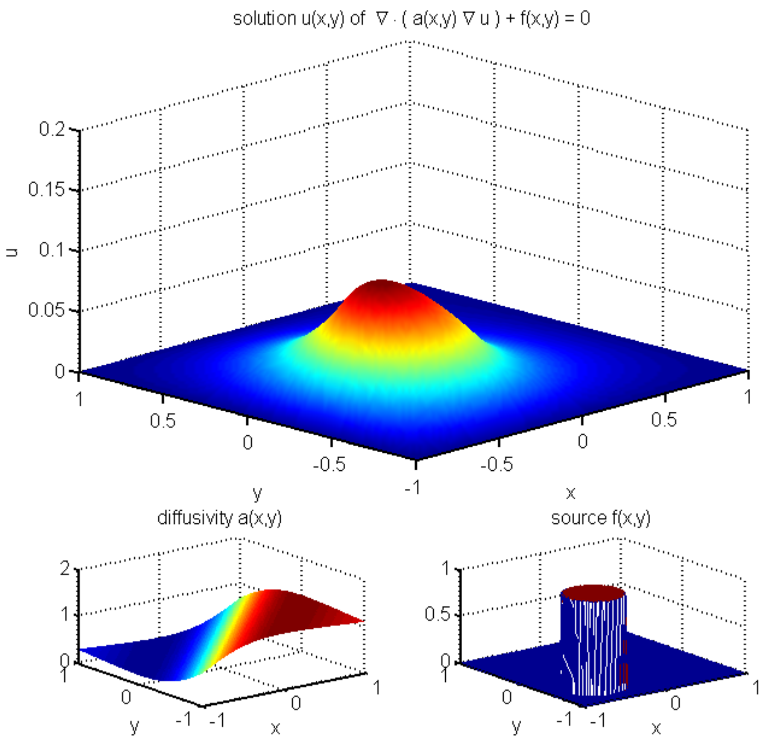

FIGURE 1. Figure produced by `diffusion(100)`.

Fortunately the sparse linear solver invoked by "`A\b`" works very well in this case. In fact, though the size N of the problem goes up by a factor of 110 in this case, the time I measured for solving just the linear system only goes up by a factor of 164. (I used "`tic, v = A\b; toc`" to measure solve time.) In other words, the work scales as $O(N^{1.09})$ for these minimal data; note $110^{1.09} = 164$. Compare this to scaling by $O(N^3)$, which would apply if Gaussian elimination were applied to the system.

Finally, note that the time for the `diffusion(200)` to run was 69 seconds or so, but almost all the time was spent *forming* the system $Av = b$ not solving it; the solve part was only 0.61 seconds, or roughly a hundredth of the run time.

4. (*20 points*) [*There was a small typo. The initial condition I will use for the solution is* $u(x, 0) = \exp\left(-(x - 50)^2/4\right)$.]

I would describe the telegraph equation as a generalization of the wave equation this way: The wave equation is the limit one gets when one supposes no loss in resistance in the wire (so $R = 0$) and no loss of the signal by conduction into the surrounding insulator/seawater (so $G = 0$). The self-inductance L of the wire remains, as does the capacitance effect C of the insulation/seawater. Thus $\alpha = G/C = 0$ and $\beta = R/L = 0$ which gives the wave equation $u_{tt} = c^2 u_{xx}$. Note that the speed of propagation of the signal is high but by no means equal to the speed of light. Instead $c = \sqrt{1/(LC)}$, so it is properties of the wire (its self-inductance) and of the surrounding dielectric material (insulation and seawater) that control the speed of propagation in the wire. (It might seem that L and C might be made arbitrarily small by engineering choices, but presumably if CL is very small the speed of propagation is determined by additional physical effects; special relativistic effects perhaps. In any case we don't expect the speed of propagation of signals in the telegraph wire to exceed the speed of light!)

```
function [UU, x, t] = teleleap(alpha,beta,J)
% TELELEAP Solve telegraph equation by staggered leap frog method.
% Example:
%  >> [UU, x, t] = teleleap(0,0,100);         % wave eqn
%  >> imagesc(x,t,UU), xlabel x, ylabel t
%  >> [UU, x, t] = teleleap(0.03,0.001,100);  % big leakage losses
%  >> imagesc(x,t,UU), xlabel x, ylabel t
% ELB 4/25/07

c = 1;  L = 100;  tf = 40;

dx = L / J;  x = 0:dx:L;
dt = 0.5 * dx;  N = ceil(tf / dt);  dt = tf / N;  t = 0:dt:tf;
nu = c * dt / dx;
UU = zeros(N+1,J+1);

UU(1,:) = exp(- (x-50).^2 /4);
W = - c * (UU(1,2:J+1) - UU(1,1:J)) / dx;
V = zeros(1,J+1);

c0 = (alpha + beta) * dt/2;
c1 = dt * alpha * beta;
for n = 1:N
    UU(n+1,:) = UU(n,:) + dt * V;
    W = W - nu * (V(2:J+1) - V(1:J));
    V(2:J) = ( (1 - c0) * V(2:J) - nu * (W(2:J) - W(1:J-1)) - c1 * UU(n+1,2:J) ) ...
        / (1 + c0);
end
```