18.336 Pset 3 Solutions

Problem 1: Staggered-grid Leap-frog

(a) For $\sigma = 0$, leap-frog works as follows. We first express u^{n+1} in terms of u^n and $v^{n+1/2}$ via, after taking the spatial Fourier transform to replace the m dependence with $e^{im\theta}$:

$$\hat{u}^{n+1} = \hat{u}^n + b\lambda(2i\sin\frac{\theta}{2})\hat{v}^{n+1/2},$$

where $\lambda = \Delta t/\Delta x$ as usual. Then we express $v^{n+3/2}$ in terms of $v^{n+1/2}$ and u^{n+1} via:

$$\hat{v}^{n+3/2} = \hat{v}^{n+1/2} + c\lambda(2i\sin\frac{\theta}{2})\hat{u}^{n+1}.$$

This can be expressed as a produce of 2×2matrices:

$$\begin{pmatrix} \hat{u}^{n+1} \\ \hat{v}^{n+3/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2ic\lambda\sin\frac{\theta}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 & 2ib\lambda\sin\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{u}^{n} \\ \hat{v}^{n+1/2} \end{pmatrix}$$

which, when we multiply this out, yields a G amplification matrix of:

$$\begin{pmatrix} 1 & 2ib\lambda\sin\frac{\theta}{2} \\ 2ic\lambda\sin\frac{\theta}{2} & 1 - 4bc\lambda^2\sin^2\frac{\theta}{2} \end{pmatrix}$$

with eigenvalues

$$g_\pm = 1 - 2bc\lambda^2\sin^2\frac{\theta}{2} \pm \sqrt{\left(1 - 2bc\lambda^2\sin^2\frac{\theta}{2}\right)^2 - 1}.$$

If the $\sqrt{\cdots}$ is imaginary, then $|g_\pm|^2 = 1$ and it is stable. If the $\sqrt{\cdots}$ is real, then $g_- < -1$ and it is unstable (for constant λ). Thus, stability requires a non-negative discriminant, which means that we must have $1 - 2bc\lambda^2\sin^2\frac{\theta}{2} \geq -1$ for all θ. This immediately gives the CFL condition $\boxed{bc\lambda^2 \leq 1}$ (where equality gives the funny degenerate-eigenvalue case that we'll usually just avoid).

(b) For $\sigma > 0$, the update equations become

$$\hat{u}^{n+1} = \frac{\left(1 - \frac{\sigma\Delta t}{2}\right)\hat{u}^n + b\lambda(2i\sin\frac{\theta}{2})\hat{v}^{n+1/2}}{1 + \frac{\sigma\Delta t}{2}},$$

and similarly for $\hat{v}^{n+3/2}$. Our G amplification matrix is then, after some algebra:

$$\frac{1}{\sigma_+^2}\begin{pmatrix} \sigma_+ & 0 \\ 2ic\lambda\sin\frac{\theta}{2} & \sigma_- \end{pmatrix}\begin{pmatrix} \sigma_- & 2ib\lambda\sin\frac{\theta}{2} \\ 0 & \sigma_+ \end{pmatrix} = \frac{1}{\sigma_+^2}\begin{pmatrix} \sigma_+\sigma_- & \sigma_+ \cdot 2ib\lambda\sin\frac{\theta}{2} \\ \sigma_- \cdot 2ic\lambda\sin\frac{\theta}{2} & \sigma_+\sigma_- - 4bc\lambda^2\sin^2\frac{\theta}{2} \end{pmatrix},$$

where $\sigma_\pm \equiv 1 \pm \frac{\sigma\Delta t}{2}$, leading to eigenvalues g of

$$g_\pm = \frac{\left[\sigma_+\sigma_- - 2bc\lambda^2\sin^2\frac{\theta}{2}\right] \pm \sqrt{[\cdots]^2 - (\sigma_+\sigma_-)^2}}{\sigma_+^2}.$$

We want to show that $bc\lambda^2 \leq 1$ is a sufficient condition for stability when $\sigma > 0$ (this means that our PML regions won't cause our system to go unstable, at least from a Von-Neumann analysis). It is easy to show explicitly in this case that the $\sqrt{\cdots}$ is imaginary for all θ and

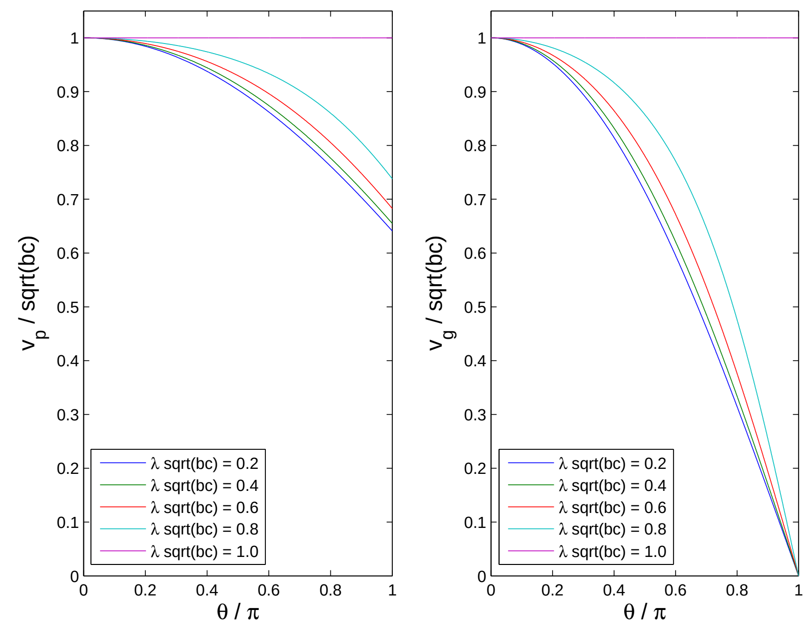

Figure 1: Phase velocity v_p (left) and group velocity v_g (right) as a function of θ for various values of $\sqrt{bc}\lambda$. Velocity is plotted in units of $\sqrt{bc}$, the exact PDE velocity.

sufficiently small Δt, which makes $|g|^2 = \sigma_-^2$ which is < 1 for sufficiently small Δt. There is actually an even simpler proof, however. Recall that we only need to show that $|g|^2 < 1 + K\Delta t$ for some $K > 0$ and for sufficiently small Δt. In particular, it is clear by Taylor-expanding g in Δt that $|g|^2 = |g(\sigma = 0)|^2 + O(\Delta t)$, and we already showed above that $|g(\sigma = 0)|^2 = 1$ for $bc\lambda^2 \leq 1$. Q.E.D. (In fact, we could have seen this even without computing G or g explicitly.)

(c) We take $\sigma = 0$, and consider individual Fourier components $u = e^{i(\theta m - \phi n)}$, $v = Ae^{i(\theta m - \phi n)}$, for frequencies $\omega\Delta t = \phi$, $\beta\Delta x = \theta$. Plugging these into the update equations, we get

$$e^{i\phi}\begin{pmatrix} 1 \\ Ae^{i\phi/2} \end{pmatrix} = G\begin{pmatrix} 1 \\ Ae^{i\phi/2} \end{pmatrix}$$

and thus we can immediately see that $g = e^{i\phi}$ (recall that $|g| = 1$ so ϕ is real, which is good because it means that group-velocity is a useful concept). We could find A via the eigenvectors of G, but this is unnecessary in order to find the velocities.1 In particular, from the expression for g above we immediately find, with some trigonometric simplification:

$$\begin{aligned}\phi(\theta) &= \pm\tan^{-1}\left[\frac{\sqrt{1-\left(1-2bc\lambda^2\sin^2\frac{\theta}{2}\right)^2}}{\left(1-2bc\lambda^2\sin^2\frac{\theta}{2}\right)}\right] \\ &= \pm\cos^{-1}\left(1-2bc\lambda^2\sin^2\frac{\theta}{2}\right) \\ &= \pm 2\sin^{-1}\left(\sqrt{bc}\lambda\sin\frac{\theta}{2}\right)\end{aligned}$$

The $\pm$ corresponds to left- and right-propagating waves, which have equal and opposite velocity. The phase velocity is $v_p = (\phi/\theta)/\lambda$ and the group velocity is $v_g = (d\phi/d\theta)/\lambda$, where

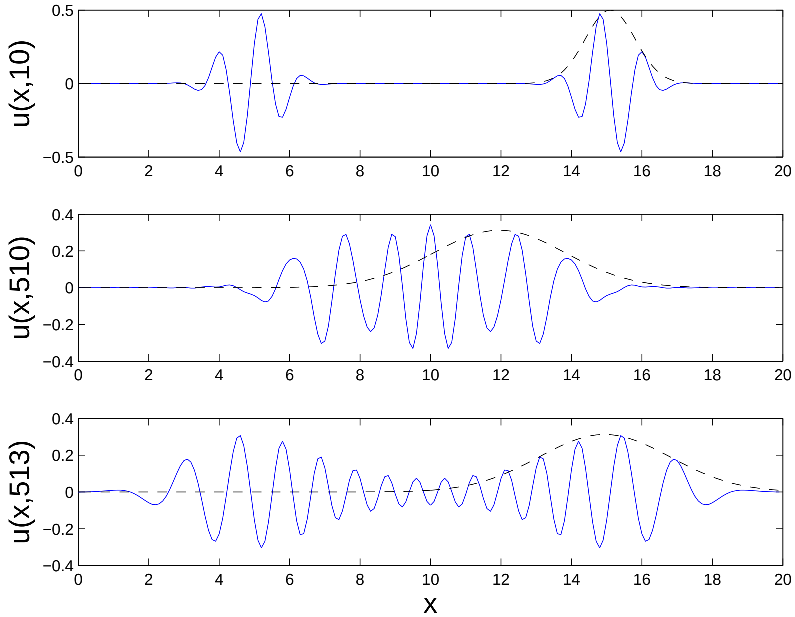

Figure 2: Pulse $u(x,t)$ location at $t = 10$ (top), $t = 510$ (middle), and $t = 513$ (bottom). Leap-frog result is shown as blue-solid line, whereas a theoretical Gaussian envelope is shown as black-dashed lines. For the top ($t = 10$), the envelope is simply fitted to the data. For the bottom two, the envelope location is predicted from the group velocity. (Envelope width in bottom two, and amplitude in all three, are simply fitted for comparison.)

the derivative is

$$\frac{d\phi}{d\theta} = \pm \frac{\sqrt{bc}\lambda \cos\frac{\theta}{2}}{\sqrt{1 - bc\lambda^2 \sin^2\frac{\theta}{2}}}.$$

Clearly, for $\theta \to 0$ we get both v_p and v_g approaching the "exact" velocity $\sqrt{bc}$ from the PDE, whereas for $\theta \to \pi$ we get $v_g \to 0$. These are shown in Fig. 1 for several values of $\sqrt{bc}\lambda$. As we might expect, as $\sqrt{bc}\lambda$ approaches 1 (the stability limit), the velocities approach $\sqrt{bc}$ (the exact velocity).

Problem 2: PML, Matlab, and You

For this problem, I used the pset3prob2.m file to implement the leap-frog scheme; this file is posted on the web site. Note that there is one crucial thing that many people overlook: for the u_t equation we are evaluating at $m\Delta x$ and must use σ_m, whereas for the v_t equation we are evaluating at $(m + \frac{1}{2})\Delta x$ and must therefore use $\sigma_{m+1/2}$; this matters in part (c), where it leads us to use different σ values for the u and v equations.

(a) At $t = 10$, the center of the right-traveling $u(x,t)$ pulse is at about $x = 15$ (determined by fitting a Gaussian envelope to the data, although you could also predict this analytically given some understanding of how the source works), as shown in Fig. 2(top). In order to calculate the appropriate group velocity, we need to know what θ is. This can be easily determined by the *temporal* frequency $\phi = \omega\Delta t$, by inverting the dispersion relation from problem 1. In

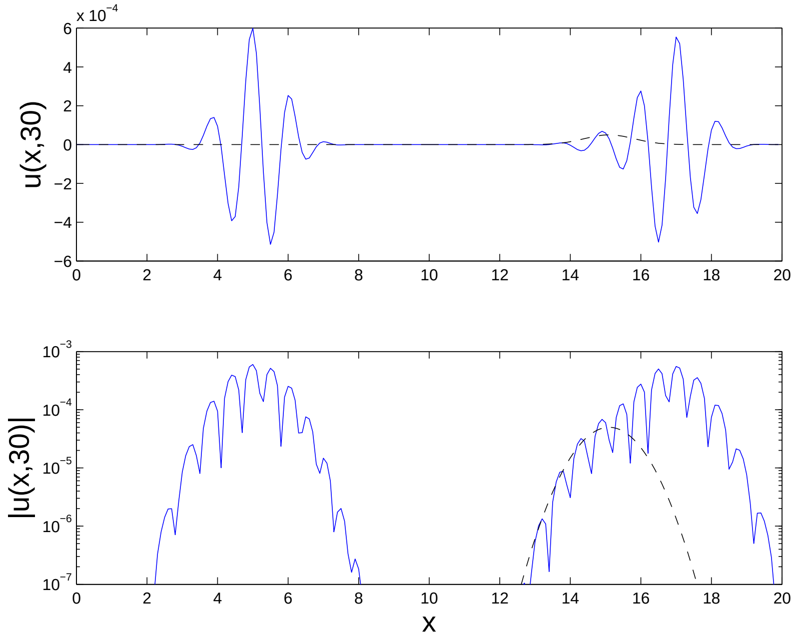

Figure 3: Computed $u(x, 30)$ for quadratic-σ PML layer (blue-solid lines), on both a linear scale (top) and absolute values on a semilog scale (bottom). Black-dashed lines show the theoretical 10^{-4} attenuated Gaussian pulse envelope for the *exact* PDE. The offset in x is due to the fact that the computed $u(x, 30)$ is dominated by numerical reflections, not by the attenuated transmitted field.

particular, we get:

$$\theta(\phi) = 2\sin^{-1}\left(\frac{\sin(\phi/2)}{\sqrt{bc}\lambda}\right)$$

and from this we find $\theta \approx 0.50102$ for $\omega = 5$, and thus $v_g = 0.99384$. This means that the pulse center at $t = 510$ should be at $11.92 \approx 15 + 0.99384 \cdot 500 \mod 20$, where the $\mod 20$ is to account for the periodic boundary conditions (x wraps around every time it changes by 20). We show the resulting data in Fig. 2(middle), along with a Gaussian located at the predicted location from the group velocity. It looks reasonable, but is a bit hard to tell because the pulses are overlapping. So, we do a third run at $t = 513$ to separate the pulses, which is shown in Fig. 2(bottom). The predicted pulse center seems to coincide very well with the actual pulse center. Note that the pulse has broadened and distorted somewhat due to dispersion!

(b) In the *exact* PDE, if we set $\sigma = \sigma_0$ in the PML region of length L, pulses travelling through the PML would be attenuated by $e^{-\sigma_0 L/\sqrt{bc}}$. Thus, if we want 10^{-4} attenuation for $L/\sqrt{bc} = 1$, we should set $\sigma_0 = 4\ln 10 \approx 9.2103$. If we do this numerically, however, we find that at $t = 30$ there is a ***substantial*** reflected wave, with $\max|u(x,t)|$ only about 0.2565 of $\max|u(x, 10)|$ (the incident amplitude). Much less than 10^{-4}!

(c) For a general $\sigma(x)$, the attenuation in the exact PDE would be $\exp(-\int_0^L \sigma(x)dx/\sqrt{bc})$. In the case of the quadratic $\sigma(x)$, $\int_0^L \sigma(x) = 2\int_0^{L/2} \sigma_2 x^2 dx = \sigma_2 L^3/12$, and thus we should set $\sigma_2 = 48\ln 10 \approx 110.52$ to get 10^{-4} reflection theoretically. Numerically, we do *much* better than before: at $t = 30$ the amplitude $\max|u(x,t)|$ is attenuated by about 0.0013 from the incident value. Nevertheless, this is still a factor of 10 worse than the theoretical 10^{-4}

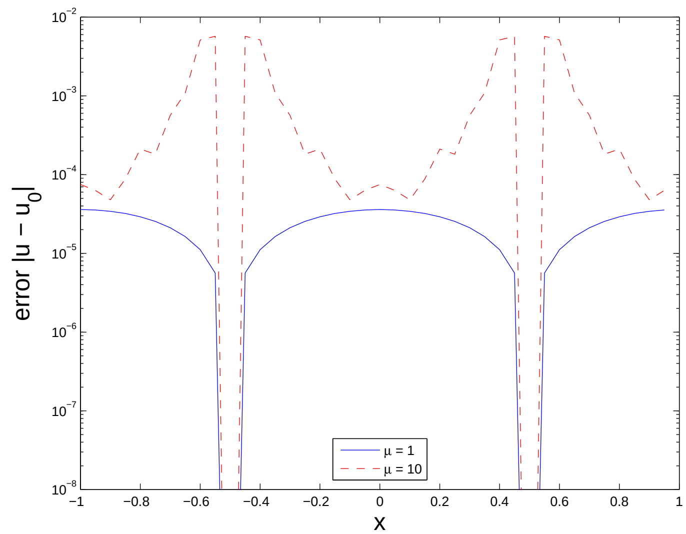

Figure 4: Absolute error in $u(x, 0.5)$ as computed by Crank-Nicolson with fixed μ compared to the analytical Fourier-series solution, for $\mu = 1$ and $\mu = 10$ with $\Delta x = 0.05$.

attenuation, and again it is due to numerical reflection. To see this more clearly, we have plotted in Fig. 3 the $u(x, 30)$ solution (solid blue) along with the predicted Gaussian-pulse envelope of the 10^{-4} attenuated right-going transmitted pulse (dashed black). On the log scale, we can (somewhat) clearly see that the $u(x, 30)$ solution is not Gaussian pulses: each "pulse" is a superposition of *two* pulses, the attenuated transmitted pulse and, slightly *ahead* of it, a much larger *reflected* pulse.

Problem 3: Diffusion

The Crank-Nicolson Scheme is implemented in the file pset3prob3.m on the web site. In particular, it can be written in the form:

$$(1 - A)u^{n+1} = (1 + A)u^n$$

where A is an $N \times N$ matrix implementing the linear operation:

$$Au_m \equiv \frac{b\mu}{2}(u_{m+1} - 2u_m + u_{m-1}).$$

So, at each time step we need to solve an equation of the form $(1 - A)u = w$. This is sparse, so if we were solving a very large problem we would definitely either use an iterative method or exploit the fact that A is tridiagonal. However, this problem is small so we just solve it stupidly by Matlab's backslash operator.

You might also think you have to write a program to implement the analytical solution, since it requires an infinite summation. However, for $t \neq 0$ the summation converges *extremely* quickly. The terms are proportional to $e^{-\pi^2(2\ell+1)^2 t}$, which for $t = 0.5$ is $< 10^{-19}$ for $\ell > 0$. Thus, to machine precision we only need the $\ell = 0$ term!

(a) For $\Delta x = 0.05$, we ran using the initial condition for $\mu = 1$ and $\mu = 10$, and the absolute error $|u - u_0|$ is plotted in Fig. 4, where u_0 is the analytical solution. The L_∞ norm $(\max |u - u_0|)$ is 0.000036 for $\mu = 1$ and 0.0057 for $\mu = 10$, a difference of a factor of about 160. It is

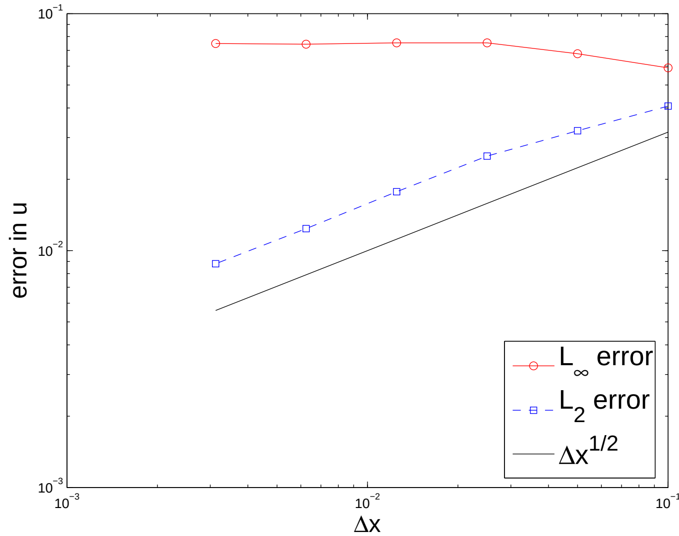

Figure 5: L_∞ and L_2 error in $u(x, 0.5)$, versus Δx resolution, as computed by Crank-Nicolson with fixed $\lambda = 1$. The L_∞ error does not converge, while the L_2 error converges as $\sqrt{\Delta x}$ (exact $\sqrt{\Delta x}$ scaling shown for reference).

tempting to say that, since Crank-Nicolson has $O(\Delta t^2)$ error, and decreasing μ decreases Δt by the same amount, then we might expect the error to decrease by 100 when we decrease μ by 10. However, matters are not so simple: if we were decrease μ further, we would find that the error eventually approaches a constant, due to the fact that Δx is fixed and we become dominated by the spatial discretization error.

(b) For this part, we keep $\lambda = \Delta t/\Delta x$ fixed at $\lambda = 1$, so that $\mu = \lambda/\Delta x$ varies with resolution. This still converges, more or less, since both Δt and Δx go to zero and Crank-Nicolson is unconditionally stable. However, the convergence rate is much worse than the quadratic rate one might expect for *smooth* initial conditions, as seen in Fig. 5. In particular, the L_∞ error $\max |u - u_0|$ doesn't converge at all—it is asymptotically constant! The L_2 error ($\sqrt{\sum [u(m\Delta x) - u_0(m\Delta x)]^2 \Delta x}$) does converge, but only as $\sqrt{\Delta x}$.