AMath 586 / ATM 581
Take-home Final
Due Thursday, June 9, 2016

Name: Your name here

Due to Canvas by 11:00pm PDT on the due date.

To submit, see https://canvas.uw.edu/courses/1038268/assignments/3292505

This final is worth 65 points. An extra credit problem worth up to 10 points more is at the end.

Please work on this exam on your own! You may consult the literature and use the discussion board and office hours if questions arise.

Before June 3: Please fill out the course evaluation form if you have not already done so, found at:
https://uw.iasystem.org/survey/161314
Feedback is encouraged to improve this course in the future.

The notebook `notebooks/animate_demo.ipynb` shows how to make animations using the `interact` command in a notebook, as discussed in lecture on May 27.

Problems 1 and 2 follow up on the discussion from lecture on dispersive waves and the fact that Leapfrog allows waves to propagate in the wrong direction for the advection equation. If you want to read more about this, I suggest the paper *Group Velocity in Finite Difference Schemes* by L. N. Trefethen, SIAM Review 24(1982), http://dx.doi.org/10.1137/1024038.

Problem 1. Using the notebook `notebooks/Lax-Wendroff_wave-packet` as a guide, create a notebook that implements the Leapfrog method with wave packet initial data and periodic boundary conditions.

Note that you need to specify two sets of initial data, $U_j^0 = \eta(x_j)$ and U_j^1 at time k, which is not provided as part of the initial data for the PDE. To begin with, set $U_j^1 = \eta(x_j - ak)$, the true solution at time $t_1 = k$.

(a) Check that your method is second order accurate using the same convergence test code as in the Lax-Wendroff notebook (i.e. produce a table of errors and also a log-log plot). Use $a = 2$, $\xi_0 = 100$, and go to the final time $T = 1$ using N time steps, for values of $m = 50r - 1$ and $N = 120r$ for $r = 1,\ 2,\ 4,\ 8,\ \ldots,\ 128$.

Hint: If you are only seeing first order accuracy, make sure that you are taking the right number of time steps since the first time step is now computing U^2, not U^1 as in Lax-Wendroff. If you take one too many time steps of size k, the solution will be off by $O(k)$ and hence appear first order accurate. Check for other possible errors too of course.

(b) Try using the initial data $U_j^1 = U_j^0 = \eta(x_j)$. This introduces an $O(k)$ error at time t_1. Use the convergence test in the notebook to show that in spite of this the method still produces $O(k^2)$ errors at the final time (with ak/h fixed) provided you use an even number of time steps, e.g. for the set of values suggested above. But if you use an odd number of time steps, e.g. $N = 120r + 1$ in the convergence test, you should see only first order accuracy asymptotically. Try to explain these results. **Hint:** $U_j^n \approx u(x_j, t_n) + O(k^2)$ when n is even, but what about n odd?

(c) Try setting $U_j = -\eta(x_j + ak)$. This does not model the PDE data well at all, but is chosen to maximally excite the parasitic mode that propagates in the wrong direction and oscillates in time, as demonstrated in lecture on May 27.

For Gaussian initial data ($\xi_0 = 0$) plot the solution $U^0,\ U^1,\ U^2,\ U^3$ to verify that this oscillates in time, and also plot the solution after some larger number of steps to verify that it propagates in the wrong direction.

Problem 2.

Create a new notebook `Leapfrog_outflow.ipynb` that implements Leapfrog for the advection equation $u_t + au_x = 0$ on the interval $0 \leq x \leq 1$ with boundary condition $u(0,t) = 0$. Use the same wave packet initial conditions as in the previous problem,

$$u(x,0) = \eta(x) = \exp(-\beta(x-0.5)^2)\cos(\xi_0 x)$$

with $\beta = 100$ and ξ_0 being an input variable for your function. Note that this isn't exactly 0 at $x = 0$ but $\exp(-25) \approx 10^{-11}$ so the solution shouldn't be affected by this — the true solution simply propagates to the right and the boundary $x = 1.5$ should be an outflow boundary where no boundary condition can be prescribed for the PDE.

Use $U^1 = \eta(x - ak)$ as the starting value, set $m = 99$ and use 300 time steps so that the Courant number is $2/3$.

Since Leapfrog is a 3-point method, we need to use something other than Leapfrog to compute U^{n+1}_{m+1} at the right boundary. This problem explores what happens with different numerical boundary conditions.

Set $\xi_0 = 0$ so that the initial data is a pure Gaussian.

You might want to introduce a parameter `bcmethod` into the calling sequence of your function so that you can easily switch between trying different methods. Produce a few sample plots (or animations in a notebook) for each case.

(a) Try setting $U^{n+1}_{m+1} = 0$ in each step. Because Leapfrog is a multistep method in time it allows left-going waves as well as right-going waves. Confirm that with this boundary condition the outgoing wave creates a reflection at the boundary that propagates back in. Note that the reflected wave has maximum possible wave number $\xi = \pi/h$, i.e. a sawtooth oscillation (modulated by a Gaussian). What happens when this parasitic wave hits the left boundary, where U^{n+1}_0 is prescribed?

(b) Try using the following modification of the Leapfrog method as your boundary condition: $U^{n+1}_{m+1} = U^{n-1}_{m+1} - \frac{ak}{2h}(U^n_{m+1} - U^n_m)$. You should find that this leads to a fairly strong reflected parasitic wave, although not nearly as bad as the method in part (a), and it should converge as the grid is refined.

(c) Try using the upwind method as your boundary condition, i.e. $U^{n+1}_{m+1} = U^n_{m+1} - \frac{ak}{h}(U^n_{m+1} - U^n_m)$. You should find that this works better than the method in (b).

(d) Finally try using the Beam-Warming method at the right boundary. Note that this should work better than upwind. Compute the max-norm of the error at the final time $T = 1$ to confirm this. (You could plot the error at each time and animate this if you want to see better how the errors behave, but not required.)

The Allen-Cahn Equation. The remaining problems lead you through the numerical solution of a PDE called the Allen-Cahn equation.

First consider the ODE

$$v'(t) = \frac{1}{\epsilon} g(v) \tag{1}$$

where

$$g(v) = v(\alpha - v)(v - 1) \tag{2}$$

with $0 < \alpha < 1$. This equation has three possible steady state solutions: $v(t) \equiv 0$, $v(t) \equiv \alpha$, $v(t) \equiv 1$. The middle one is an *unstable* steady state. If $v(0) = \alpha + \delta$ with δ small but nonzero, then $v(t)$ moves away from α, towards 0 if $\delta < 0$ or towards 1 if $\delta > 0$. These are the two stable steady states. The parameter $\epsilon > 0$ controls the rate of decay towards these steady states. For small ϵ the solution moves rapidly towards 0 or 1.

Problem 3. Set $\alpha = 0.3$ and $\epsilon = 1$. Use `odeint` in Python to plot solutions curves $v(t)$ for several different initial values $v(0)$ lying between 0 and 1, in particular for `v0 = linspace(0,1,11)`. Plot all these curves $v(t)$ for $0 \le t \le 10$ on a single plot.

Produce similar plots for $\epsilon = 0.1$ and $\epsilon = 0.05$.

We can turn (1) into a PDE in one space dimension and time by letting $v(x, t)$ vary in space and adding spatial diffusion, obtaining

$$v_t(x, t) = \kappa v_{xx}(x, t) + \frac{1}{\epsilon} g(v). \tag{3}$$

This is a scalar reaction-diffusion equation, a variant of the *Allen-Cahn equation* that is used as a simple model of phase transition.

The lower stable steady state corresponds to a material in one phase (e.g. solid) while the upper stable steady state corresponds to a different phase (e.g. liquid). If the cubic $g(v)$ is replaced by the quadratic $g(v) = v(1-v)$ then (3) is called *Fisher's equation* and models the propagation of a gene in a population, for example.

Consider initial data

$$v(x, 0) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x \ge 0 \end{cases} \tag{4}$$

and the Cauchy problem on $-\infty < x < \infty$ so we don't have to worry about boundary conditions for the moment.

If $\kappa = 0$ (no diffusion) then $v(x, t) = v(x, 0)$ for all time since both $v = 1$ and $v = 0$ are steady states and so $v_t \equiv 0$. With diffusion, however, this step discontinuity immediately smooths out and $v(x, t)$ for $t > 0$ will be a continuous function taking all values between 0 and 1. For these values of v the reaction term drives v back towards 0 (where $v < \alpha$) or towards 1 (where $v > \alpha$), tending to sharpen the smeared profile back towards a step discontinuity. There is a competition between the smearing effect of diffusion and the sharpening effect of the reaction, leading to a steady profile that is smeared to a finite degree that depends on the relation between the parameters ϵ and κ.

The smearing effect of diffusion is symmetric about $v = 1/2$: for $\epsilon \to \infty$ the solution to the pure diffusion equation with data (4) has $v > 1/2$ for $x < 0$ and $v < 1/2$ for $x > 0$ and appears symmetric about this point. The sharpening from the reaction term is also symmetric about $v = 1/2$ if $\alpha = 1/2$. In this case $v(x, t)$ approaches a steady state profile $v(x, t) \to V(x/\delta)$ as $t \to \infty$. A new parameter δ has been introduced that will be related to κ and ϵ below. The idea is that the profile $V(\xi)$ should be independent of the parameters κ and ϵ but is rescaled based on these parameters since the width of the transition from $v = 1$ to $v = 0$ will depend on these parameters.

We can determine δ and V by inserting $v(x,t) = V(x/\delta)$ into the PDE (3), obtaining a boundary value problem

$$0 = \frac{\kappa}{\delta^2} V''(x/\delta) + \frac{1}{\epsilon} g(V(x/\delta)). \tag{5}$$

Multiplying by δ^2/κ and rearranging gives

$$V''(x/\delta) = -\frac{\delta^2}{\kappa\epsilon} g(V(x/\delta)). \tag{6}$$

This suggests that we should choose δ^2 to be proportional to $\kappa\epsilon$ in order to obtain an ODE for V that is independent of the parameters. In order to easily solve the resulting BVP it is convenient to choose

$$\delta = \sqrt{2\kappa\epsilon}. \tag{7}$$

Setting $\xi = x/\delta$ then gives the ODE for $V(\xi)$,

$$V''(\xi) = -2V(\xi)(1/2 - V(\xi))(V(\xi) - 1). \tag{8}$$

with asymptotic boundary conditions $V(\xi) \to 1$ as $x \to -\infty$ and $V(\xi) \to 0$ as $x \to \infty$. We also want $V(\xi)$ to be centered about $\xi = 0$, so we would like $v(0) = 1/2$.

Note that even without solving the ODE (8) we can deduce that the width of the transition zone in the traveling wave is proportional to δ and hence to $\sqrt{\kappa\epsilon}$. This information might be useful if we wanted to use a nonuniform grid to solve the problem numerically, or to choose an appropriate value of h for a uniform grid.

In fact the ODE can be solved and the solution satisfying the conditions stated above is

$$V(\xi) = \frac{1}{1 + \exp(\xi)} \tag{9}$$

Hence this is a steady state solution for the case $\alpha = 1/2$.

If $\alpha \neq 1/2$ then the effect of the reaction term is not symmetric. If $0 < \alpha < 1/2$ then some values of v less than $1/2$ are driven towards $v = 1$ by the reaction term. When coupled with the symmetric diffusion this leads to a traveling wave propagating with some velocity c that is positive if $\alpha < 1/2$ (or negative if $\alpha > 1/2$).

The traveling wave profile is given by the same function $V(x)$ that satisfies the boundary value problem (8), but now it translates at some speed c, and has the form

$$v(x,t) = V((x - ct)/\delta), \tag{10}$$

Problem 4

(a) For any $0 < \alpha < 1$, show that $v(x,t) = V((x-ct)/\delta)$ is a traveling wave solution to (3) provided that c satisfies

$$c = \sqrt{\frac{2\kappa}{\epsilon}} \left(\frac{1}{2} - \alpha\right). \tag{11}$$

Hint: It might be useful to first show that $V'(\xi) = V(\xi)(V(\xi) - 1)$.

(b) Suppose we define the width of the transition zone (wave front) in a traveling wave to be the distance in x over which v falls from 0.99 to 0.01. Show that the width of wave front is roughly 9δ. This can be used to choose a suitable value of h. For example, choosing $h \approx \delta$ would give roughly 9 grid points in the wave front, which is probably about the minimum needed to resolve it well numerically.

Problem 5.

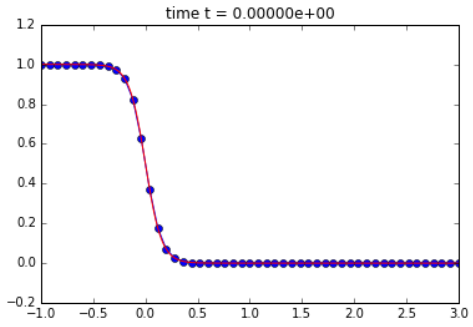
Implement a fractional step method to approximate solutions to the Allen-Cahn equation on the interval $-1 \le x \le 3$ with initial data $v(x,0) = V(x/\delta)$, where δ is determined from specified values of $\kappa = 0.3$ and ϵ.

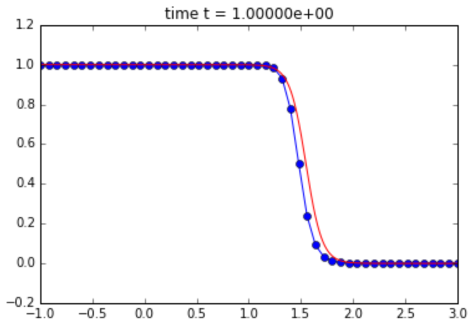
Your code should use:

- The Crank-Nicolson method for the diffusion equation. You might want to use code from the notebook `notebooks/HeatEquation_CN.ipynb` for the diffusion part. Adapt this to work on the interval $-1 \le x \le 3$ and use boundary conditions $u(-1,t) = 1, \;\; u(3,t) = 0$, which are reasonable as long as the traveling wave does not reach the boundary.
- The Forward Euler method for the ODE in each time step, or the 2-step explicit Runge-Kutta method (5.3). Introduce a parameter `odemethod` in the calling sequence of your function to select one of these.

Test your method using $\alpha = 0.3$ and $\epsilon = 0.01$ with initial data given by the exact traveling wave solution at $t = 0$, going up to time $t = 1$. Produce some plots of the solution to show that the method works.

You should see results like this for the case when Forward Euler is used and $m = 49$ with 100 time steps up to time 1:





What order of accuracy do you observe for each choice of ODE solver? Is it worth using the Runge-Kutta method in view of the first-order errors introduced by the fractional step method?

Extra Credit. Worth up to 10 additional points.

Explore what happens with smaller values of ϵ and comment on the various things that can go wrong if the mesh width h and/or the time step k are not chosen small enough. Analyze what you see as much as possible.

Some ideas to consider are below, but this is open-ended.

- What general guidelines can you deduce from experiments about how h and k should be chosen relative to ϵ to get good results?
- Do you think the ODE solver part of this problem is "stiff", and would you be able to take larger time steps if you replaced the explicit solver by an implicit method such as TR-BDF2?
- If you try using an implicit method and take much larger time steps (e.g. with $k/\epsilon \gg 1$), what sort of behavior do you see, and why?
- Would it be better to use TR-BDF2 in place of Crank-Nicolson for the diffusion equation?