Computational Physics Homework 4

Stanley "Alex" Breitweiser"

November 17, 2015

1 Question 1

1.1 Part a - Method of Characteristics

Starting with

$$f_t + bxf_x = ax^n$$

we see that it is of the form for evolution along a coordinate curve,

$$\frac{df}{ds} = \frac{dt}{ds}f_t + \frac{dx}{ds}f_x.$$

This implies that, along this curve,

$$\frac{df}{ds} = ax^2 \qquad \frac{dt}{ds} = 1 \qquad \frac{dx}{ds} = bx.$$

Using t for our s coordinate (note that this is valid since they are co-varying and we are free to constantly reparameterize $t_0 = s_0 = 0$), this means that our function f evolves as

$$\frac{df}{dt} = ax^n$$

along a curve defined by

$$\frac{dx}{dt} = bx.$$

Solving the coordinate equation explicity, we see that

$$x = x_0 e^{bt};$$

using this to solve for the value of f along the curve, we see that it evolves as

$$f = f_0 + \int_0^t ax^n dt = f_0 + ax_0 \int_0^t e^{nbt} dt$$

$$f = f_0 + \frac{ax_0}{nb}(e^{nbt} - 1).$$

Note that f_0 and f above are functions of x_0, so we need to correctly invert the x equation when plugging it in

$$f = f_0(x_0(x)) + \frac{ax_0(x)}{nb}(e^{nbt} - 1)$$

$$f = f_0(xe^{-bt}) + \frac{ax}{nb}(e^{(n-1)bt} - e^{-bt}).$$

1.2 Part b - Solution and Plotting

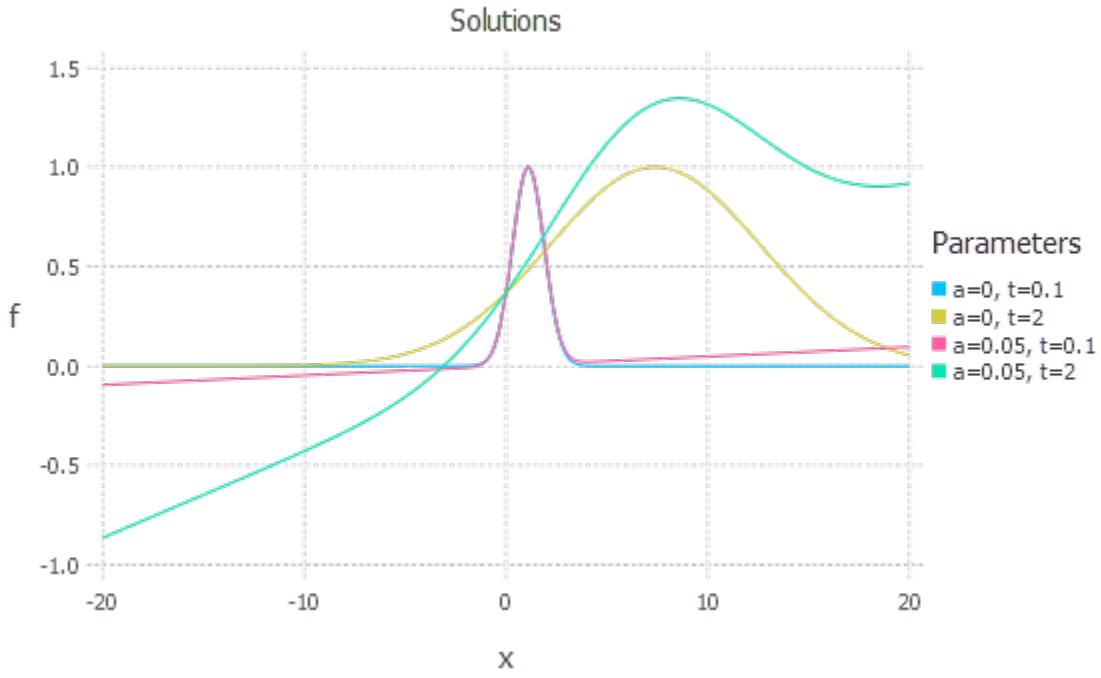
Using our initial condition

$$f_0(x_0) = e^{-(x_0-1)^2} = e^{-(xe^{-t}-1)}$$

and the values $n = b = 1$, we obtain the expression for $f(x,t)$

$$f = e^{-(xe^{-t}-1)} + ax(1-e^{-t})$$

and the following plots for $a = 0, 0.05$ at times $t = 0.1, 2$:



We see that, for $a = 0$, we get a wave propogating to the right and diffusing, as expected from the form of the solution with no a term; but for a non-zero, a linear offset evolves as the e^{-t} suppression disappears, leaving us with an ax term.

2 Question 2

2.1 Part a - Gauss-Seidel Relaxation

We want to solve the Poisson equation,

$$\nabla^2\Phi = 1,$$

on the quadrilateral

$$0 \le x \le 1.5 \qquad 0 \le y \le 2 \qquad y \ge 1.5 - 2x \qquad y \le 2.75 - 1.5x$$

with the boundary condition $\Phi = 0$.

We solve this using the Gauss Seidel relaxation method; namely, we consider this as the stationary solution of the equation

$$\frac{\partial\Phi}{\partial t} = \nabla^2\Phi - 1,$$

and assume that $\frac{\partial \Phi}{\partial t}$ converges to 0 in the long-time limit from our initial conditions.

We start by discretizing our space into a 100 by 100 grid in the space

$$0 \leq x \leq 1.5 \qquad 0 \leq y \leq 2$$

and starting with the initial conditions

$$\Phi^0_{j,l} = 0,$$

where $\Phi^n_{j,l}$ is the value of Φ on the nth iteration at the jth x point and the lth y point. Since our boundary conditions are homogenous, we simply iterate inside the region of interest, leaving Φ to be 0 outside and on the boundary. (If the BCs were inhomogenous, we would decompose the solution linearly and separately iterate on the source free boundary conditions and the homogenous-boundary source solutions, recombining them at the end). This will leave slight numerical errors near the boundary, since our finite differentiation might sample outside the boundary and the solution is not exactly zero there, but this should still give a good approximation on the interior of the region of interest.

To iterate, we start by by substituting our finite difference formulas into the differential equation (note that h_x and h_y are the known x and y step sizes, respectively, and we consider the time step to be 1):

$$\Phi^{n+1}_{j,l} - \Phi^n_{j,l} = \frac{\Phi^n_{j+1,l} + \Phi^{n+1}_{j-1,l} - 2\Phi^n_{j,l}}{h_x^2} + \frac{\Phi^n_{j,l+1} + \Phi^{n+1}_{j,l-1} - 2\Phi^n_{j,l}}{h_y^2} - 1$$

and re-arrange to obtain

$$\Phi^{n+1}_{j,l} = \frac{1}{2(h_x^2 + h_y^2)}(h_y^2(\Phi^n_{j+1,l} + \Phi^{n+1}_{j-1,l}) + h_x^2(\Phi^n_{j,l+1} + \Phi^{n+1}_{j,l-1})) - \frac{h_x^2 h_y^2}{2(h_x^2 + h_y^2)}.$$

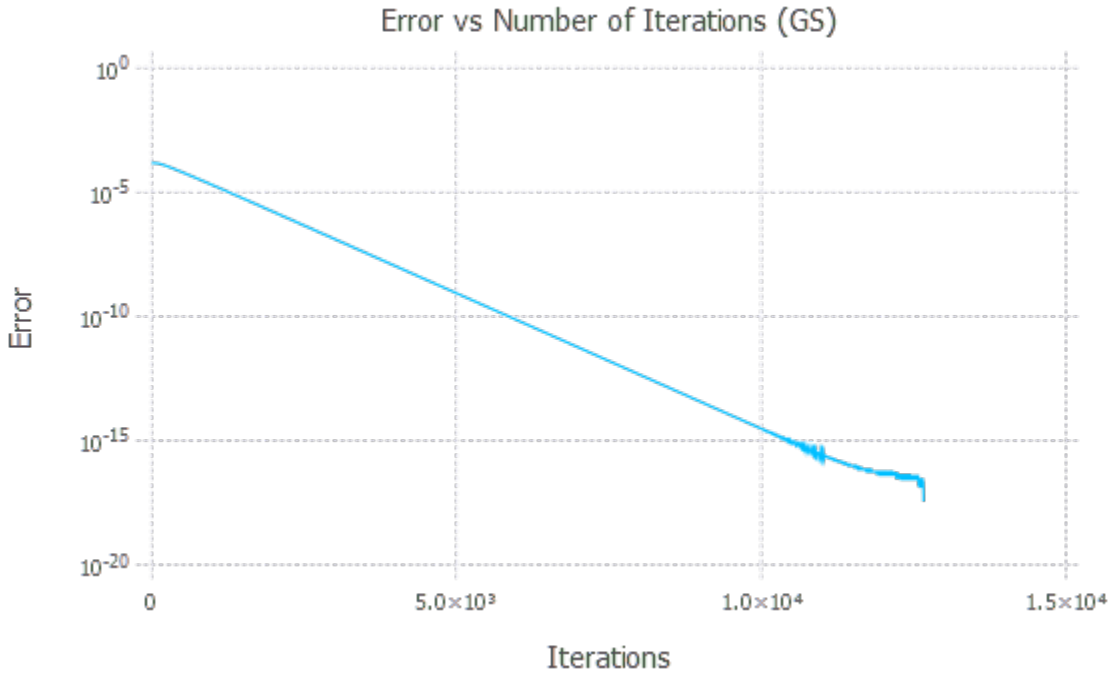
Note that the unequal-time spatial derivatives (mixing n and $n+1$ points) are the Gauss-Seidel convention; as discussed in class, they give better convergence properties, and they also conveniently eliminate the need to store extra time steps, as they naturally arise during iteration from the bottom left to the top right.

2.2 Part b - Error Estimation

To guess the number of iterations needed for convergence, we use the approximation we derived in class,

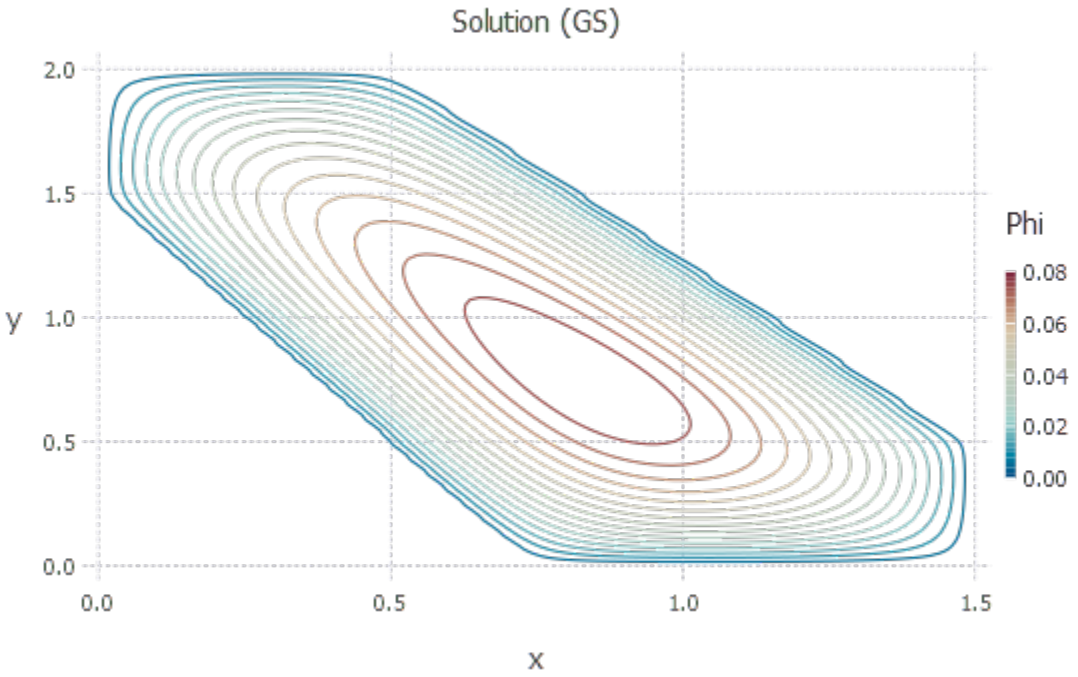
$$N \approx \frac{1}{4} p J^2,$$

where J is the grid size and p is the desired accuracy in base 10. Picking p to be 10, we actually find that the error (estimated by the maximal change over a time step) converges to zero before the estimated 25,000 iterations.



2.3 Part c - Contour Plot of Solution

Below we include a contour plot of the solution obtained using Gauss-Seidel relaxation.



2.4 Part d - Successive Over-Relaxation

Successive over-relaxation uses a different updating rule, which tries to "over-shoot" the solution and converge back in an attempt to speed up the convergence. In particular, we use the new updating rule,

$$\Phi_{j,l}^{n+1} = (1-w)\Phi_{j,l}^{n} + w\frac{1}{2(h_x^2+h_y^2)}(h_y^2(\Phi_{j+1,l}^{n} + \Phi_{j-1,l}^{n+1}) + h_x^2(\Phi_{j,l+1}^{n} + \Phi_{j,l-1}^{n+1})) - \frac{h_x^2 h_y^2}{2(h_x^2+h_y^2)}$$

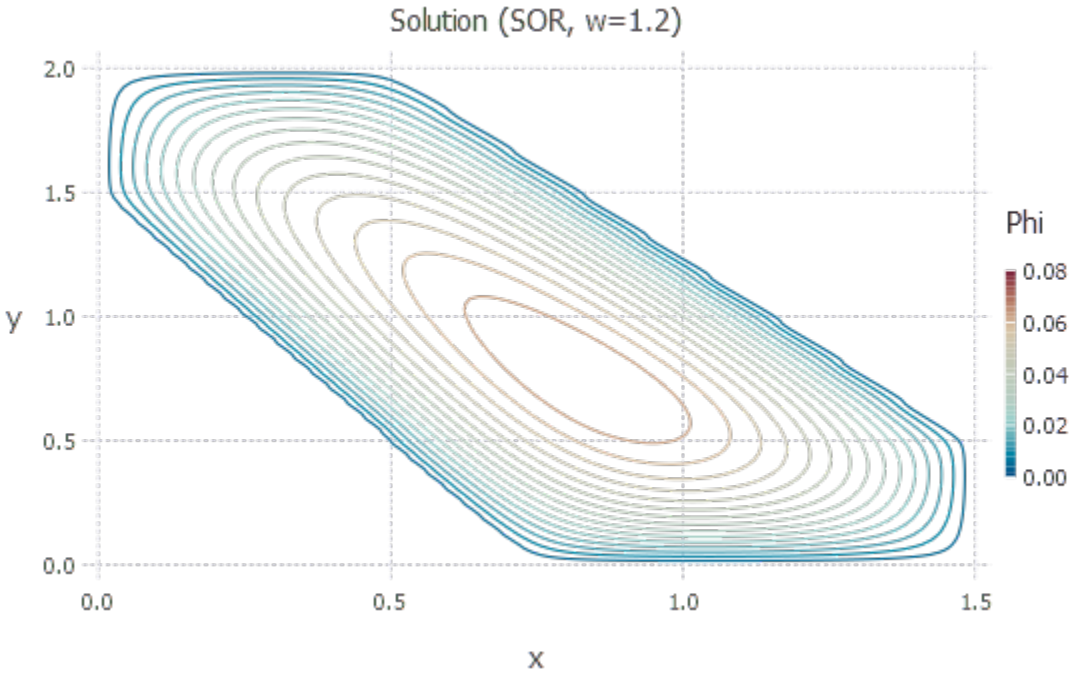
$$1 \leq w \leq 2,$$

which weights the Gauss-Seidel derivative terms by w and the old term by $1-w \leq 1$ (note that the source terms remains unchanged). We pick $w = 1.2$, for reasons discussed later. Indeed, this does seems to converge faster, as predicted by our estimate in class,

$$N \approx \frac{1}{3}pJ.$$

(Note that, in the below figure, we ran for the same number of iterations as it took for Gauss-Seidel to converge, for consistency).

However, it underestimated the solution a little (note the slightly lower coloring in the contour plot):



This tradeoff seems to increase with w; larger values converged faster but gave solutions further below Gauss-Seidel. $w = 1.2$ seemed to give the best trade-off, nearly halving the iterations and getting almost the same solution.

3 Question 3

3.1 Part a - Intuition

The dependence of v on ρ makes sense because an individual driver wants to have sufficient space to stop between him and the next driver, in case the next driver stops suddenly. Therefore, the speed should be 0 when then cars are completely packed, and maximum (the speed limit) when there are no other cars. The only linear function which satisfies this is

$$v = v_{max}(1 - \frac{\rho}{\rho_{max}}),$$

which is therefore in some sense the best linear approximation satisfying these constraints.

Since v is a function only of ρ, we can expand

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0$$

into

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho\frac{\partial v}{\partial x} = 0$$

and reduce

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho\frac{\partial v}{\partial \rho}\frac{\partial \rho}{\partial x} = 0$$

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial x} + \rho(-\frac{v_{max}}{\rho_{max}})\frac{\partial \rho}{\partial x} = 0$$

$$\frac{\partial \rho}{\partial t} + v_{max}(1 - 2\frac{\rho}{\rho_{max}})\frac{\partial \rho}{\partial x} = 0$$

This gives us a wave speed

$$c(\rho) = v_{max}(1 - 2\frac{\rho}{\rho_{max}}) \leq v$$

which makes sense; waves can only travel as fast as their densest (slowest) part, which should be twice their average density (assuming maximal amplitude waves).

3.2 Part b - Initial Conditions

We start by setting the parameters for the solution; we assume cars are about 20 ft, the length of a full size truck. We want to sample around the same length as a car, so we set our sampling length to be about 100 ft, implying a maximum density of 5. We want to model something like a rush hour commute, so we sample along a 10 mile (52,800 ft) track over a one hour (3600 seconds) time period, which is more than long enough to circle the track. The speed limit is usually 65mph, which is about 95.3 feet per second. We then set our time sampling, h_t, based on our Courant number restriction,

$$\mu := \frac{h_t * c}{h_x} \leq \frac{1}{2}$$

which gives us roughly (since c is of the order $v_{max} \approx 100$)

$$h_t = \frac{h_x}{2c} \approx \frac{100}{200} \approx 0.5s$$

. This is required for numerical stability, as discussed in class. For safety, since performance really isn't a concern, we set our time sampling to $h_t = 0.1s$

After setting these parameters, we set up an initial density

$$\rho_0 = \rho_{max} \qquad -\frac{L}{4} \leq x \leq 0$$

and zero elsewhere.

3.3 Part c - Lax and FTCS Solutions

To solve with FTCS, we simply use our discrete equations for derivatives, to obtain the updating rule

$$\rho_i^{n+1} = \rho_i^n - \frac{h_t}{2h_x}[f(\rho_{i+1}^n) - f(\rho_{i-1}^n)] \qquad f(\rho) = \rho v(\rho).$$

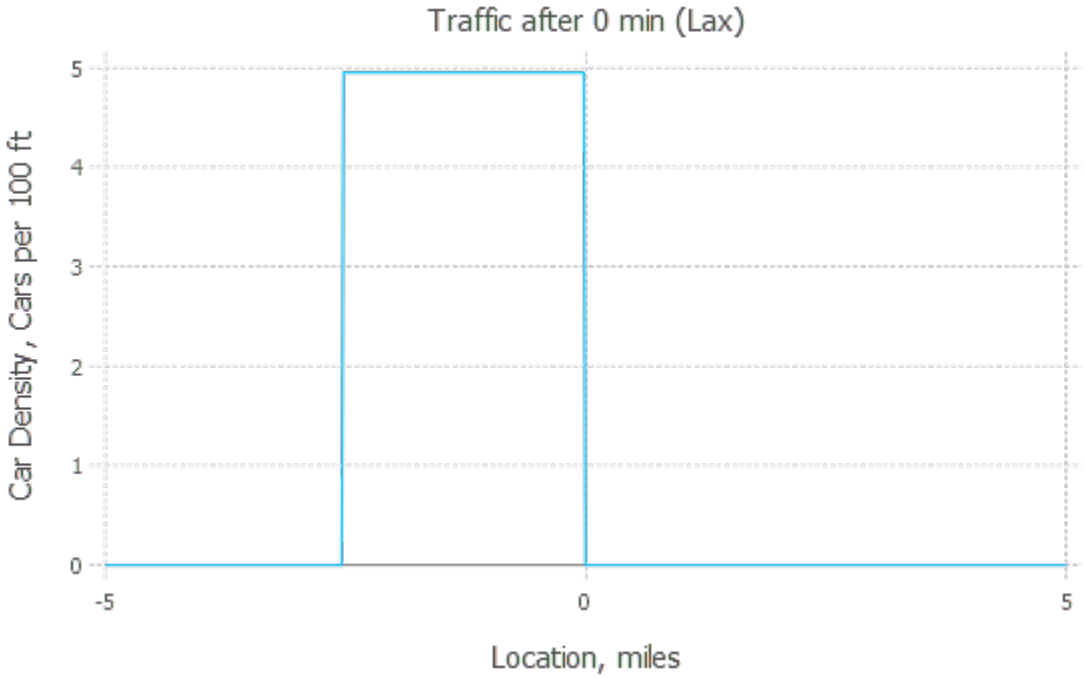
While these equations could have been rewritten in terms of just $c(\rho)$ and ρ (note that $c(\rho) = f'$) as

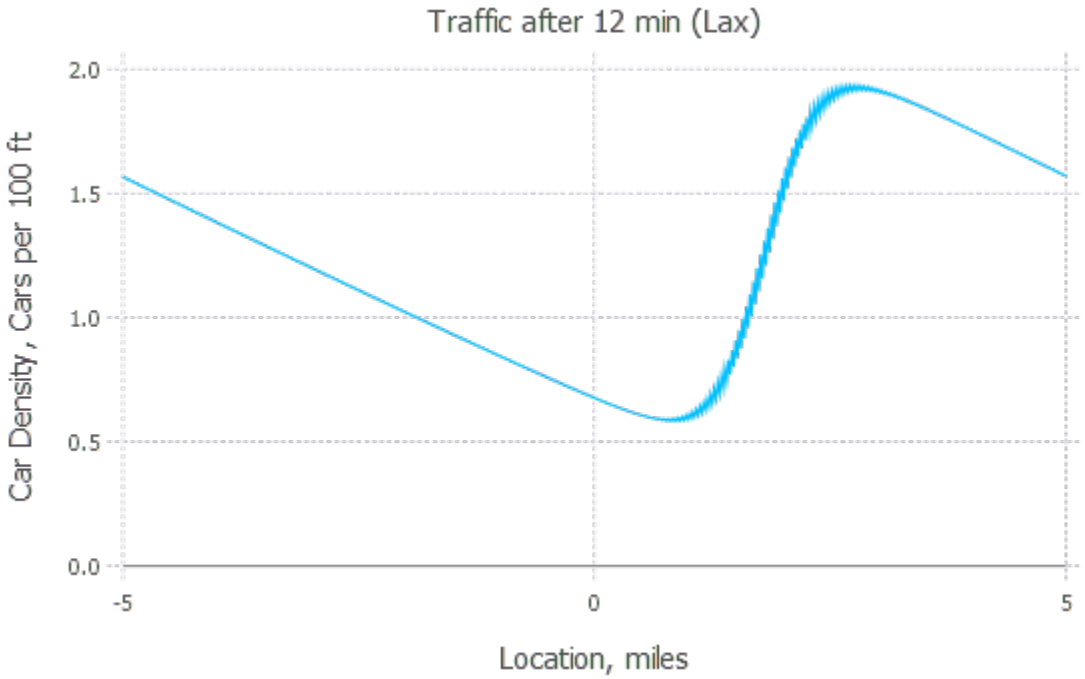
$$\rho_i^{n+1} = \rho_i^n - \frac{h_t}{2h_x}c(\rho_i^n)[\rho_{i+1}^n - \rho_{i-1}^n],$$

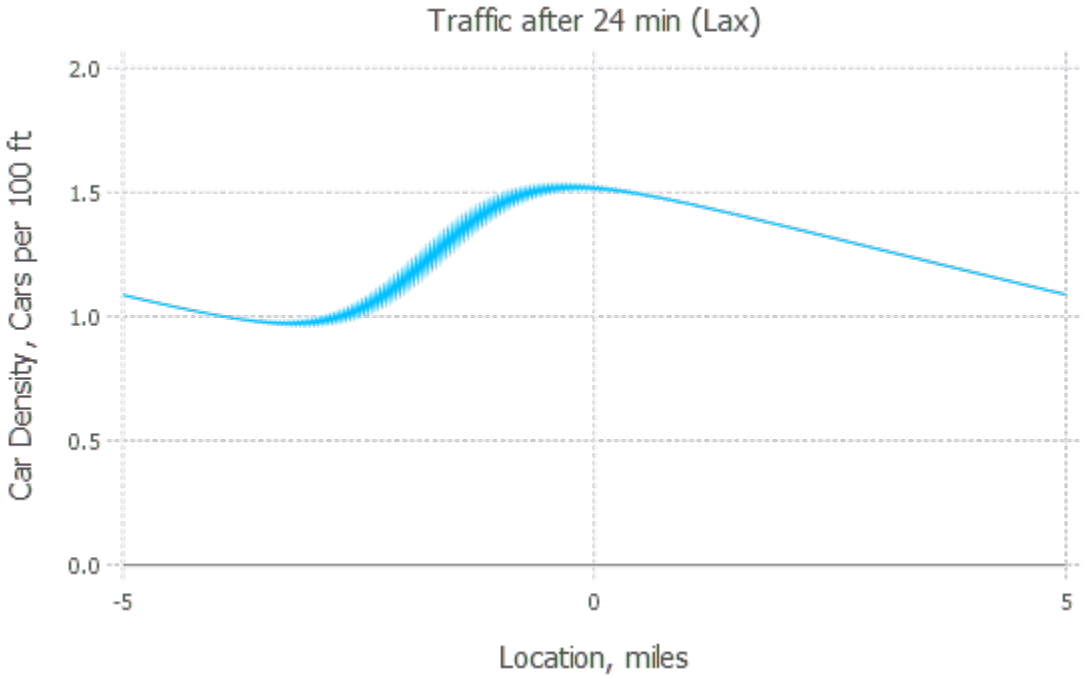
the former version seemed to show better numerical stability (discussed more in Part d). The Lax updating rule is similar, with the exception that it smooths out the previous value a bit by averaging the neighboring value; this gives us the updating rule

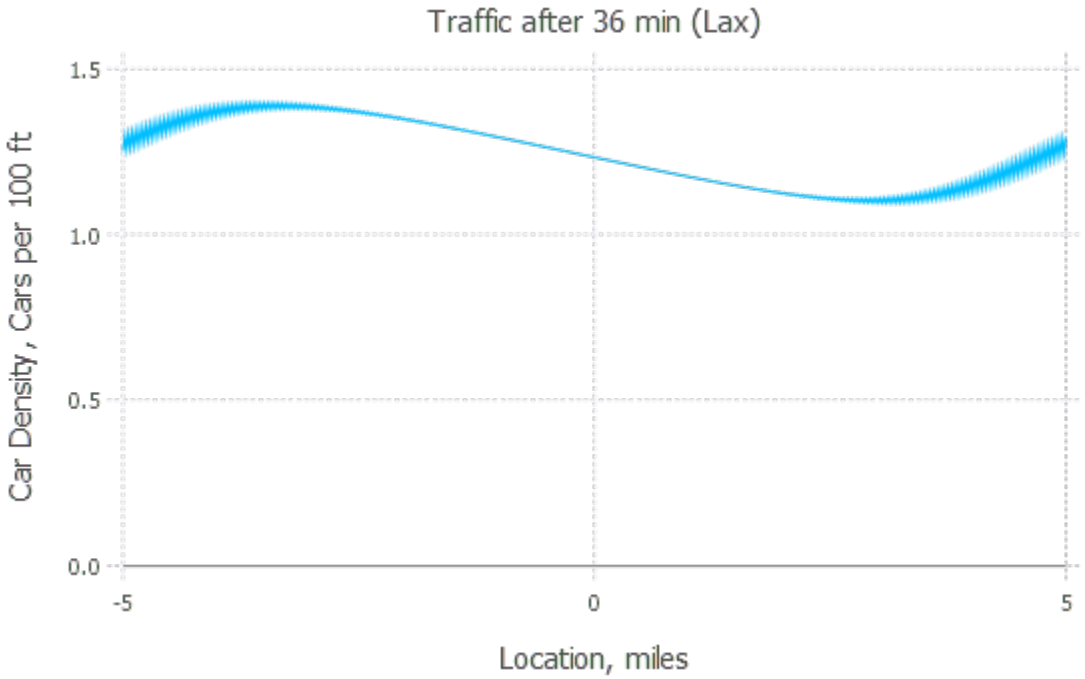
$$\rho_i^{n+1} = \frac{1}{2}(\rho_{i+1}^n + \rho_{i-1}^n) - \frac{h_t}{2h_x}[f(\rho_{i+1}^n) - f(\rho_{i-1}^n)].$$

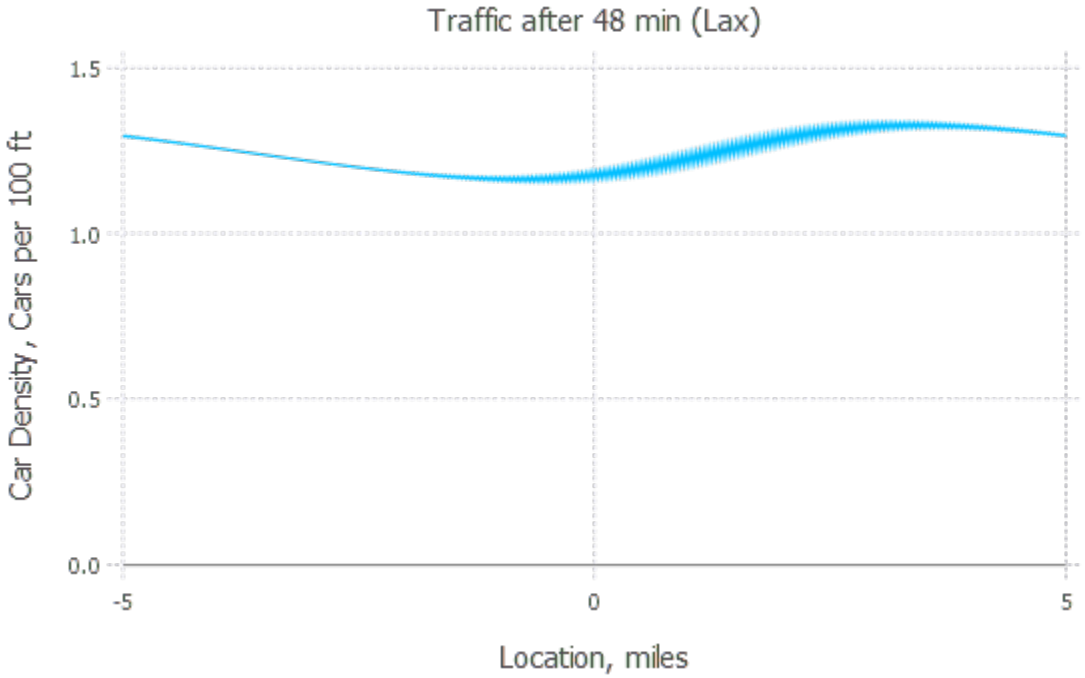
The time evolution of both is shown below, as snapshots of the traffic density.

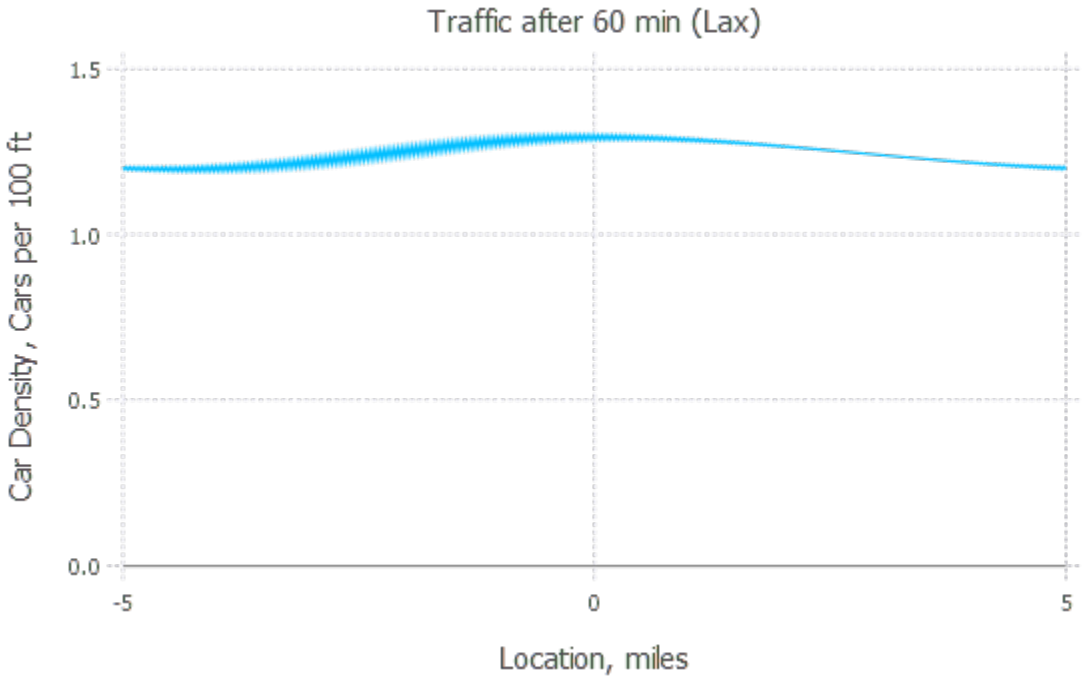

Traffic after 0 min (Lax)
Car Density, Cars per 100 ft
5
4
3
2
1
0
-5
0
5
Location, miles

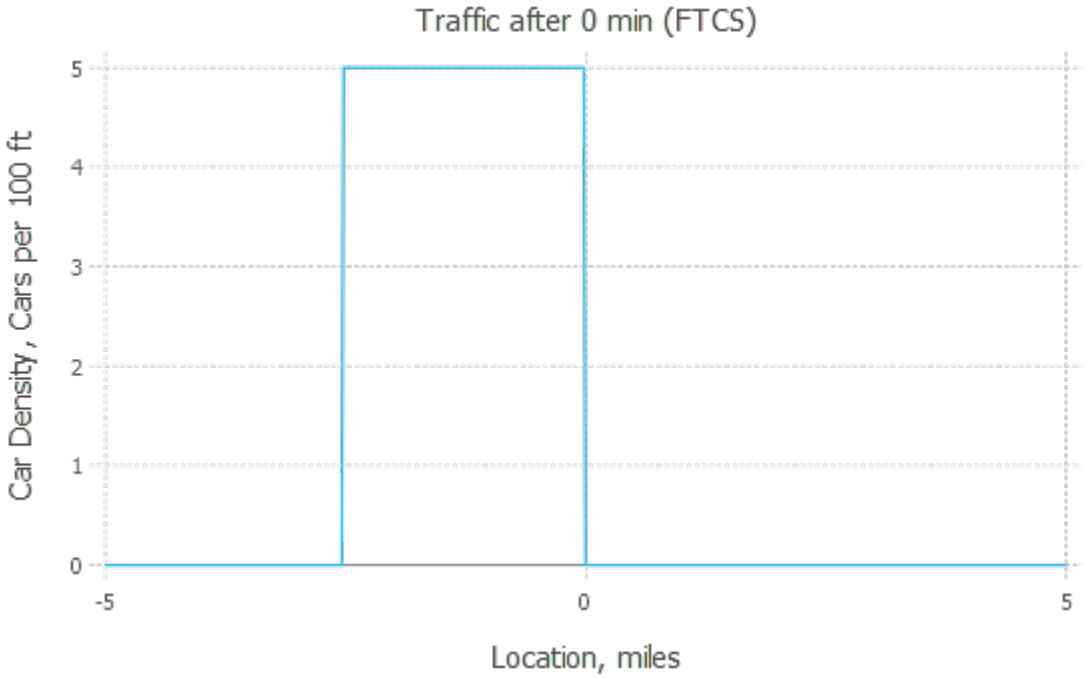

Traffic after 12 min (Lax)
Car Density, Cars per 100 ft
2.0
1.5
1.0
0.5
0.0
-5
0
5
Location, miles

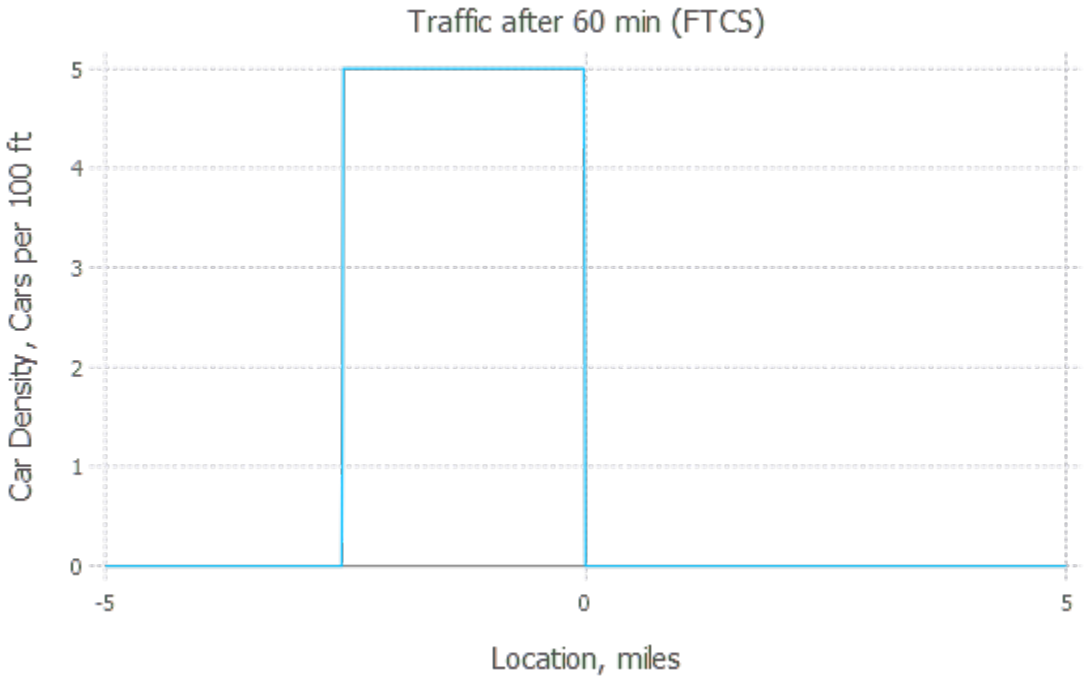

Traffic after 24 min (Lax)
Car Density, Cars per 100 ft
2.0
1.5
1.0
0.5
0.0
-5
0
5
Location, miles


Traffic after 36 min (Lax)
Car Density, Cars per 100 ft
1.5
1.0
0.5
0.0
-5
0
5
Location, miles


Traffic after 48 min (Lax)
Car Density, Cars per 100 ft
1.5
1.0
0.5
0.0
-5
0
5
Location, miles


Traffic after 60 min (Lax)
Car Density, Cars per 100 ft
1.5
1.0
0.5
0.0
-5
0
5
Location, miles


Traffic after 0 min (FTCS)
Car Density, Cars per 100 ft
5
4
3
2
1
0
-5
0
5
Location, miles


Traffic after 60 min (FTCS)
Car Density, Cars per 100 ft
5
4
3
2
1
0
-5
0
5
Location, miles

3.4 Part d - Explanation

The Lax method correctly smooths the wave, slowly moving it forward as it disperses more and more, moving faster and smoothing until it is almost completely flat. This makes sense; drivers in the front spread out a bit, start to move faster, and clear more space for drivers behind them to move, eventually forming a smooth, constant flow.

FTCS, however, did not move the solution at all, apparently stuck in perfect stasis. In a sense, this is the only solution that makes sense; note that, for our initial condition, we have identically that $f(\rho) = \rho v(\rho) = 0$, since either ρ is zero or $\rho = \rho_{max}$, forcing v to 0. That means that our equation simply reduces to

$$\frac{\partial \rho}{\partial t} = 0,$$

the equation of a perfectly static state. Replacing with the $c(\rho)$ version seems to only do worse, since it stays static for a moment before quickly becoming unstable, producing large noise and NaN results.

We could view this as FTCS being somewhat of an unphysical rule; more precisely, the exact and sharp nature of the initial condition is unphysical, since true systems always have a small amount of noise. Lax accounts for this by applying a bit of smoothing, which seems to work well in this large system modeling. Another technique could be to apply small randomness (something like a few percent) into the intial condition, a common technique in data modeling used to break these kind of static anomalies arising from symmetric initial parameters.

4 Question 4

4.1 Part a - Galerkin Method

Here we want to solve the KdV equation,

$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} + \alpha\frac{\partial^3 f}{\partial x^3} = 0,$$

with periodic boundary conditions $-L \leq x \leq L$ and initial condition $f(x,0) = \frac{1}{8}cos(\frac{\pi x}{L})$. We do this using a finite element method, where we try to expand the solution into an approximation using a finite number of basis functions. In particular, as suggested in class, we assume the basis

$$f = \sum_{n=-M}^{M} c_n(t)e^{ik_n x} \qquad k_n = \frac{\pi n}{L}$$

which manifestly satisfies the boundary conditions, with the requirement that $c_n = c^*_{-n}$ so that f is real. Plugging this into the KdV equation, we obtain

$$\sum_n \dot{c}_n e^{ik_n x} + \sum_{n',n''} c_{n'}c_{n''}ik_{n'}e^{i(k_{n'}+k_{n''})x} - \alpha\sum_n ik_n^3 c_n e^{ik_n x} = 0.$$

Noting that exponents can never be negative, we take the inner product with each of our basis function, matching up terms with the same oscillatory behavior to obtain the elementwise equation

$$\dot{c}_n = \alpha i k_n^3 c_n - \sum_{n'=max(n-M,-M)}^{min(M,n+M)} c_{n-n'}c_{n'}ik_{n'}$$

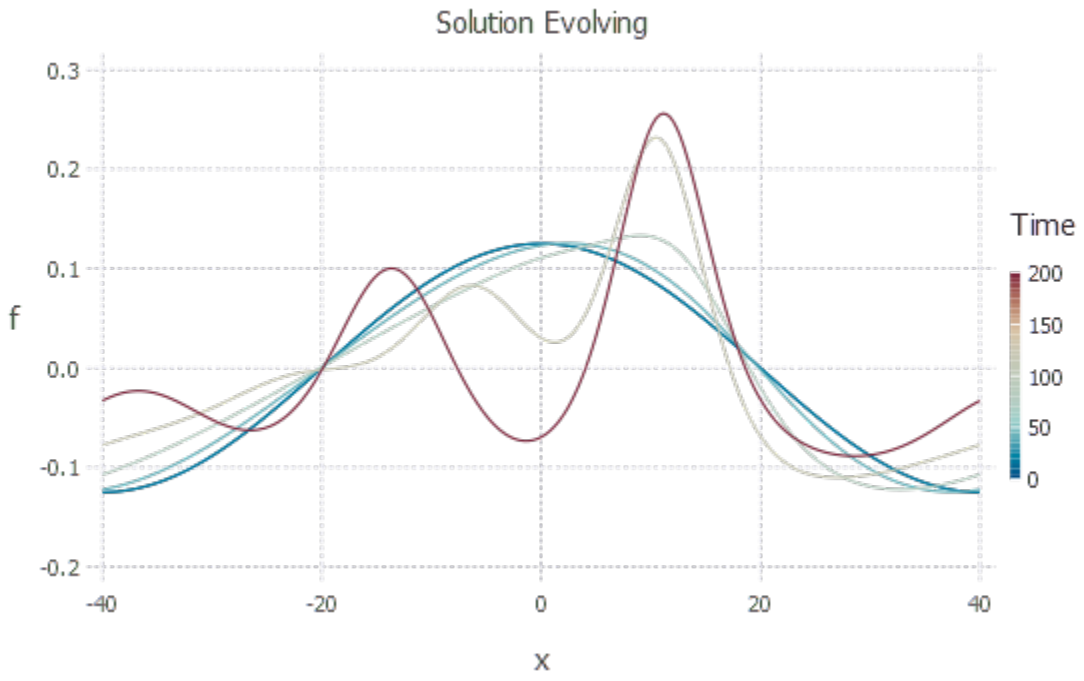
where the sums are cut off to stay within our mode range. We then simply advance this differential equation with the Runge-Kutta solver we made for Homework 2.

Note that our initial condition is

$$f(x,0) = \frac{1}{8}cos(\frac{\pi x}{L}) = \frac{1}{16}(e^{ik_{-1}x} + e^{ik_1 x})$$

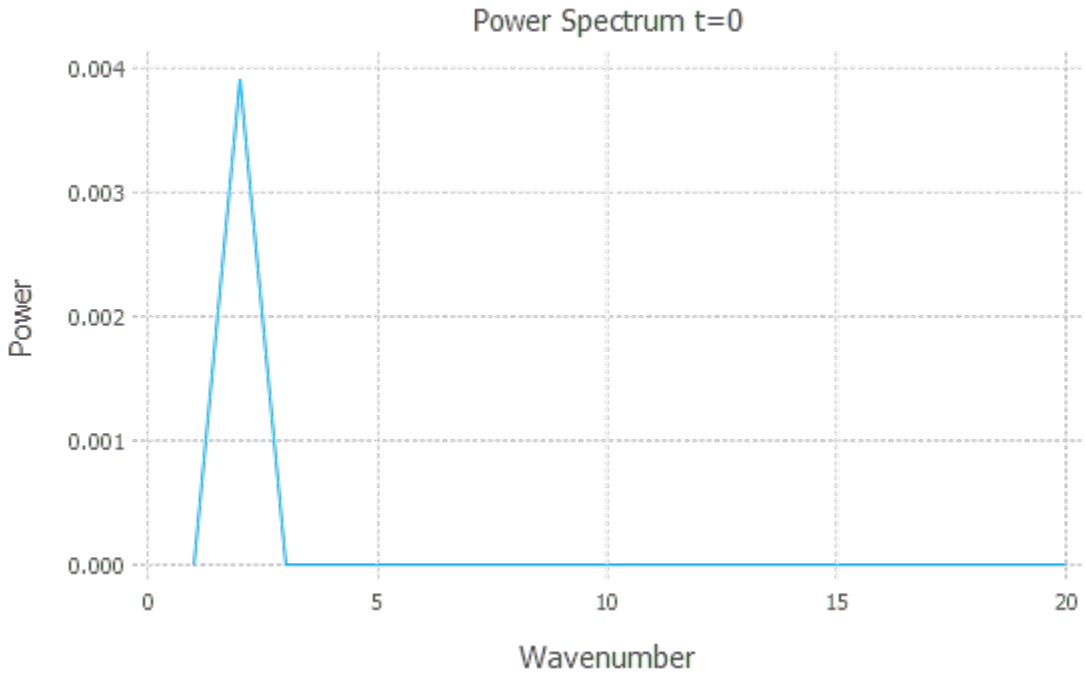
4.2 Part b - Plots

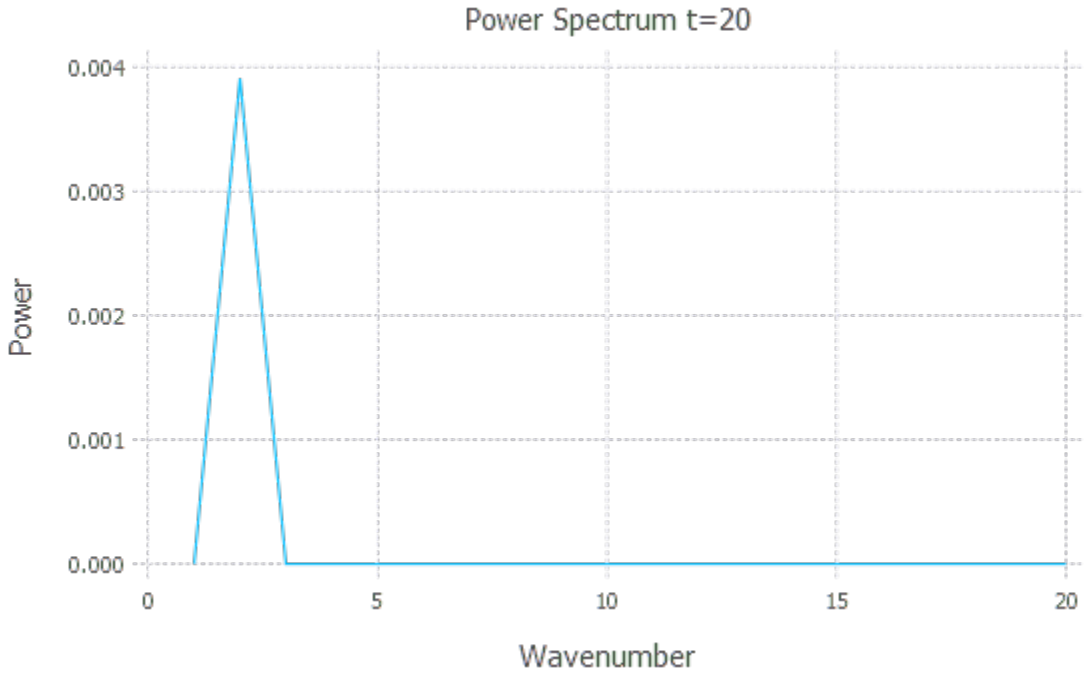
Our solution plot for various timesteps is plotted below:

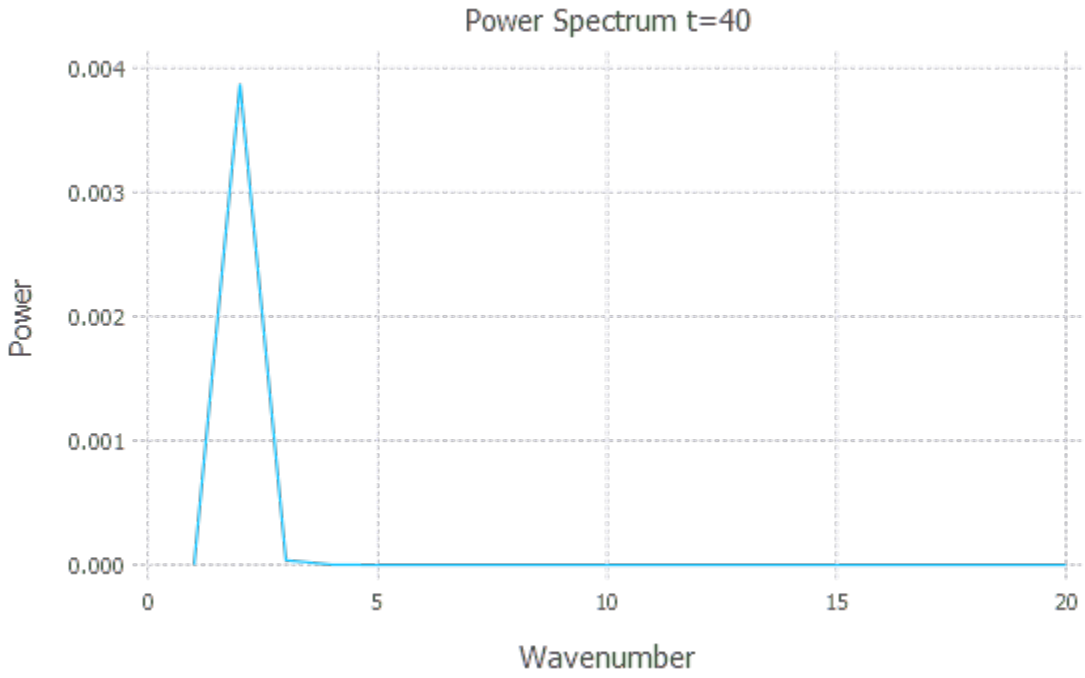


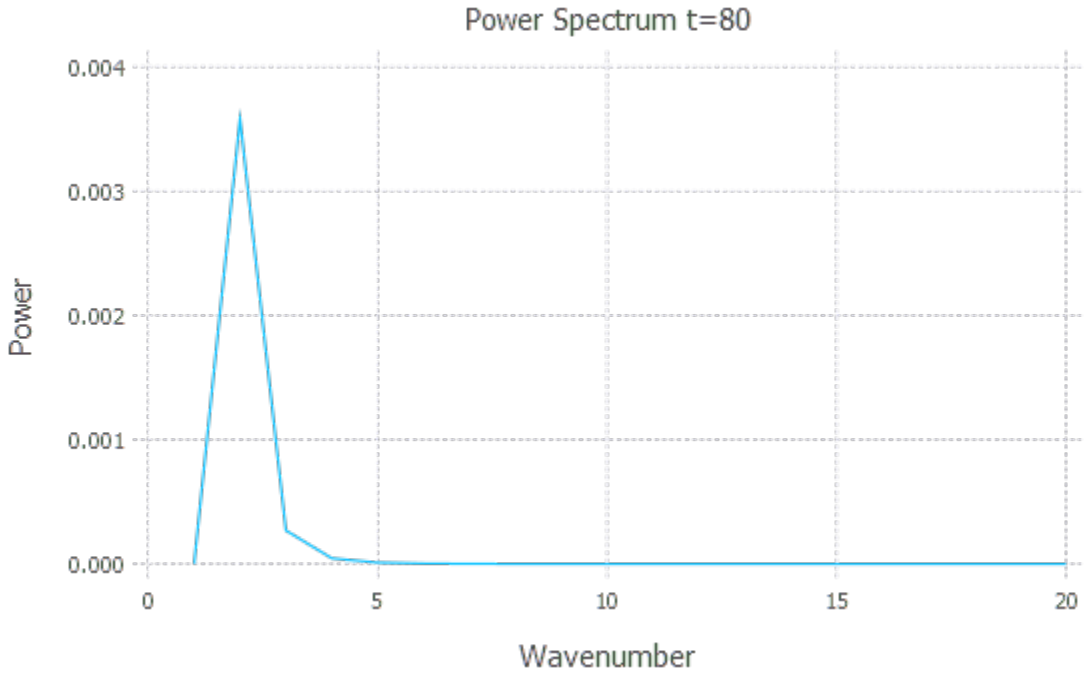
4.3 Part c - Power Spectra

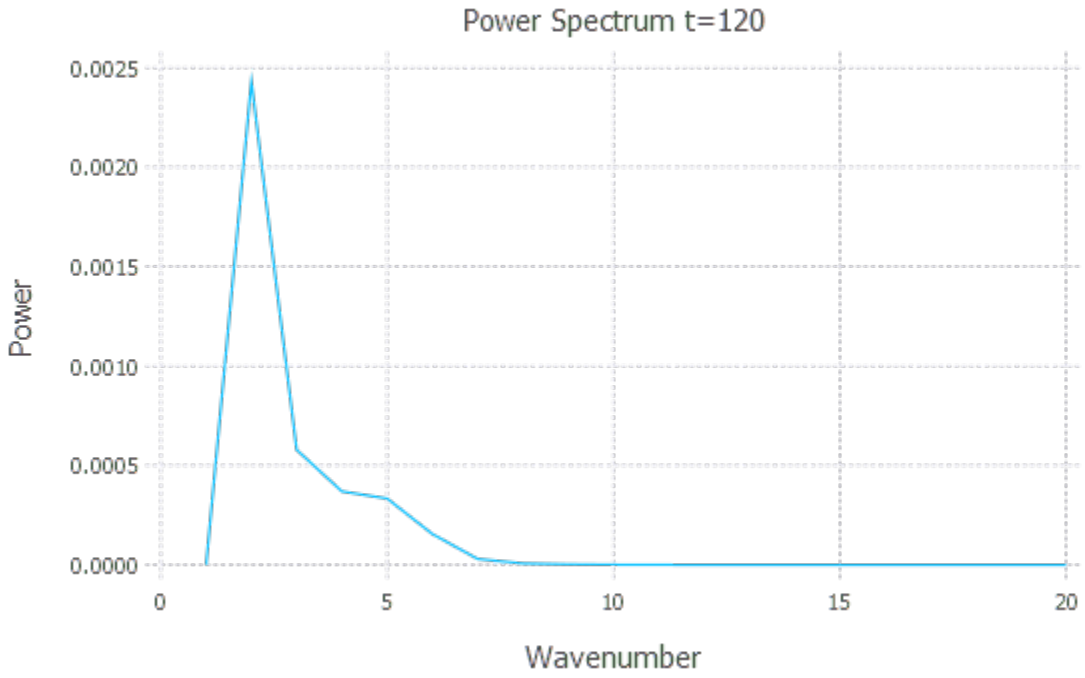
Below are the power spectra for each of the various times:

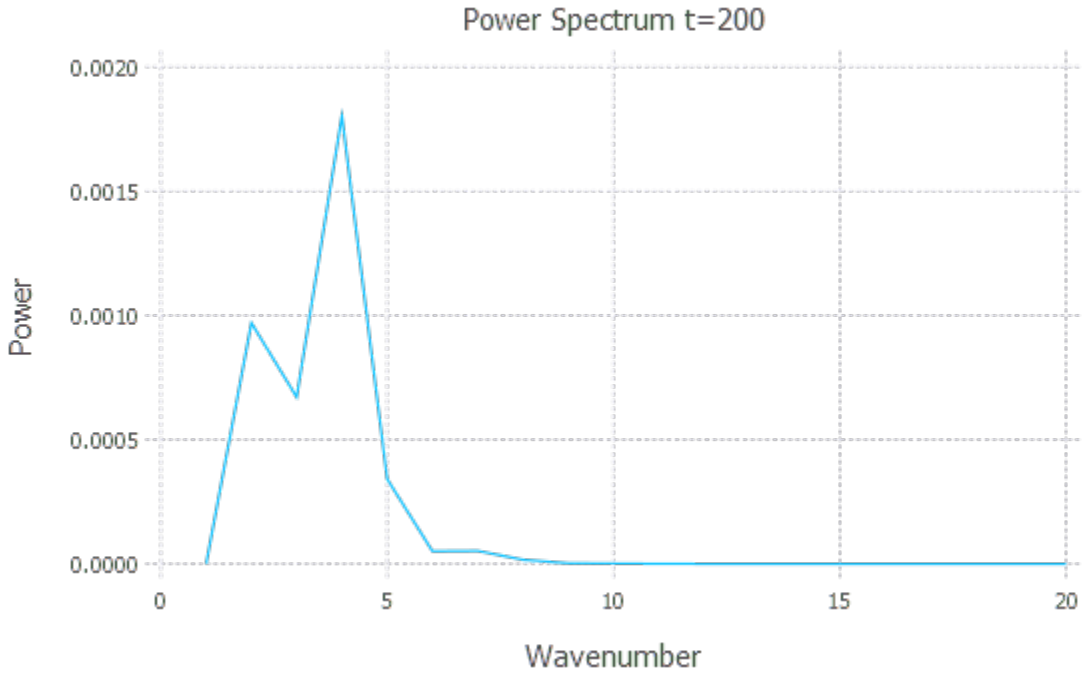

Power Spectrum t=0
0.004
0.003
0.002
0.001
0.000
Power
0
5
10
15
20
Wavenumber


Power Spectrum t=20
0.004
0.003
0.002
0.001
0.000
Power
0
5
10
15
20
Wavenumber


Power Spectrum t=40
0.004
0.003
0.002
0.001
0.000
Power
0
5
10
15
20
Wavenumber


Power Spectrum t=80
0.004
0.003
0.002
0.001
0.000
Power
0
5
10
15
20
Wavenumber


Power Spectrum t=120
Power
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
5
10
15
20
Wavenumber


Power Spectrum t=200
Power
0.0020
0.0015
0.0010
0.0005
0.0000
0
5
10
15
20
Wavenumber

4.4 Part d - Explanation

We can see what is happening by looking at our ODEs,

$$\dot{c}_n = \alpha i k_n^3 c_n - \sum_{n'=max(n-M,-M)}^{min(M,n+M)} c_{n-n'} c_{n'} i k_{n'}.$$

We see that the α term kind of scales the coefficients smaller ($k_n \leq 1$), while the other term drifts them along. More concretely, we can look at various terms in the KdV equation; if we let α be small, we get

$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} \approx 0,$$

which looks to be some linear partial differential equation that admits, for example, the solution

$$f = \frac{x}{t}.$$

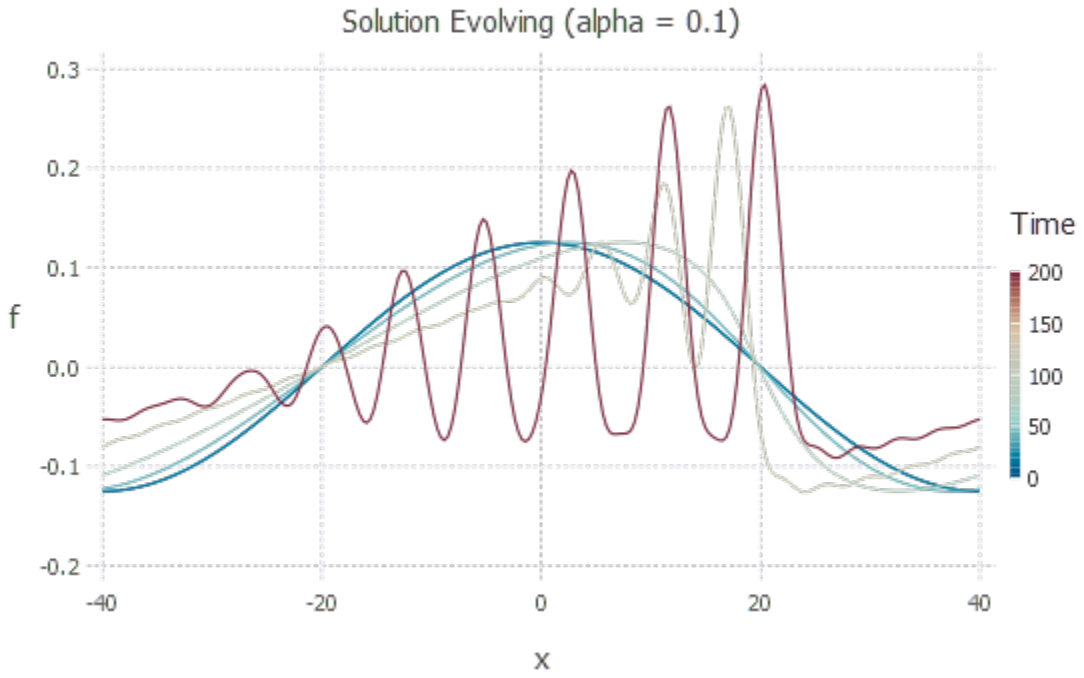
From this, we see that it can create sharp linear solutions. If, on the other hand, we let α be large, we expect to get something of the form

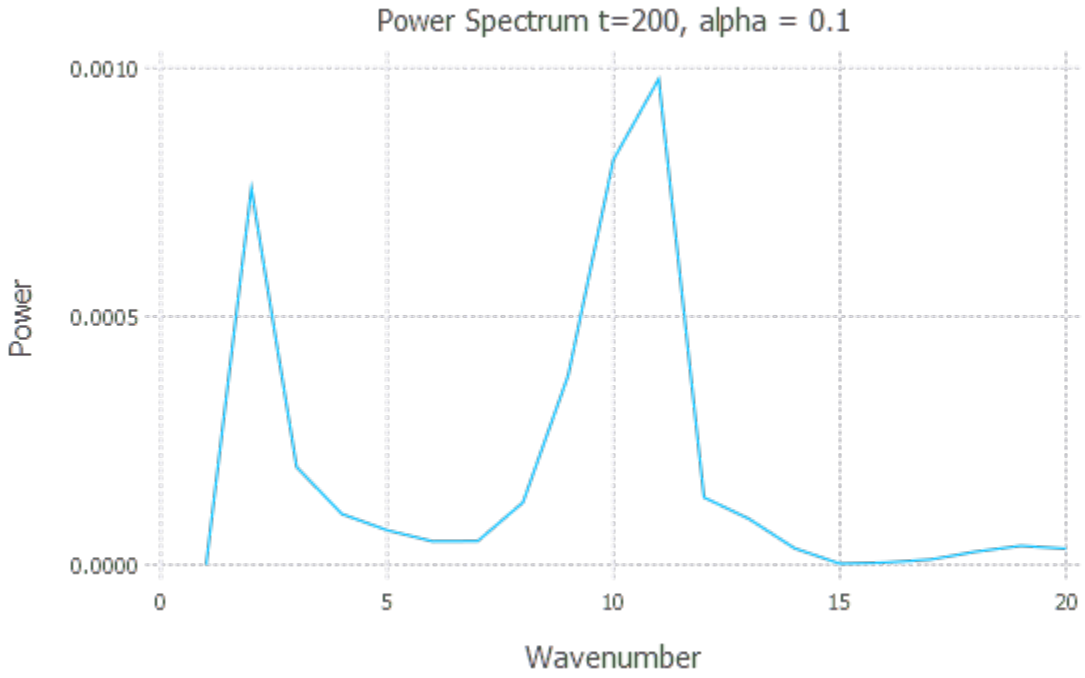
$$\frac{\partial f}{\partial t} + \alpha\frac{\partial^3 f}{\partial x^3} \approx 0,$$

which looks almost like a diffusion equation that disperses the solution over time.

This interpretation makes sense looking at our solution and power spectra; the solution starts to develop sharp peaks, but is smoothed out somewhat by some kind of diffusive process. The power spectra, similarly, starts to drift towards higher frequency, but is being damped by the α term from going too far.

We can test this by decreasing α; the solution is shown below for $\alpha = 0.1$, as is the power spectrum at $t = 200$. As predicted, we see sharper peaks and a power spectrum showing powerful high-frequency modes.




Power Spectrum t=200, alpha = 0.1
0.0010
0.0005
0.0000
Power
0
5
10
15
20
Wavenumber