Homework 4 - Partial Differential Equations

Quynh Nguyen

November 19, 2015

Note: The zip folder include main c++ code in hw4/main and ROOT plot code in /hw4/ROOT. Code is compiled in g++ (clang) in Mac OSX.

1 Problem 1

1.a Solving PDE by characteristic method

The given equation

$$\frac{\partial f}{\partial t} + bx\frac{\partial f}{\partial x} = ax^n \tag{1}$$

with initial condition f(x,0)=f_0(x). Let's introduce new set of variable (ξ,s) such that ξ =x, s=0 on t=0. Here, s is linearly independent from ξ. The goal is to make LHS an ODE of s:

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial t}\frac{\partial t}{\partial s} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial s} = ax^n \tag{2}$$

matching coefficients between eq. (1) and eq. (2) gives

$$\frac{\partial x}{\partial s} = bx \Rightarrow x = \xi e^{sb} \text{ (because x=}\xi\text{ when s=0)} \tag{3}$$

$$\frac{\partial t}{\partial s} = 1 \Rightarrow s = t \text{ (because s=0 when t=0)} \tag{4}$$

Equation (3) also defines the characteristic curve. Now integrate $\frac{\partial f}{\partial s} = a(\xi e^{sb})^n$, here ξ is treated as a constant

$$f(\xi, s) = \frac{a\xi^n}{bn}e^{sbn} + C(\xi) \tag{5}$$

where C(ξ) is a constant of integration. The initial condition, when t=s=0 and $x = \xi$

$$f(\xi, s = 0) = \frac{a\xi^n}{bn} + C(\xi) = f_0(\xi) \tag{6}$$

$$\Rightarrow C(\xi) = f_0(\xi) - \frac{a\xi^n}{bn} \tag{7}$$

The solution is then:

$$f(\xi, s) = \frac{a\xi^n}{bn} e^{sbn} + f_0(\xi) - \frac{a\xi^n}{bn} \tag{8}$$

In terms of x and t

$$f(x, t) = \frac{a(xe^{-tb})^n}{bn} e^{tbn} + f_0(xe^{-tb}) - \frac{a(xe^{-tb})^n}{bn} \tag{9}$$

simplify

$$f(x, t) = \frac{ax^n(1 - e^{-tbn})}{bn} + f_0(xe^{-tb}) \tag{10}$$

1.b Solutions graph

1.b.1 Plot

See Figure 1 and 2. Plotting was done in a simple Mathematica notebook problem1.hw4.nb in main subfolder

1.b.2 Explanation

The solution for the given initial condition and n=b=1 is

$$f(x, t) = ax(1 - e^{-t}) + e^{-(x.e^{-t} - 1)} \tag{11}$$

with a=0, $f(x, t) = f_0(x) = e^{-(x.e^{-tb} - 1)}$ which approaches e (fatten out) for large value of t. with a=0.05, over a long time, the function become e+ ax. That is, a straight line start out at e with slope of 0.05

2 Problem 2: Possion equation

2.a Code implementation for Gauss-Seidel and SOR

code in 2.GaussSeidel.cc and 2.SOR.cc under main subfolder

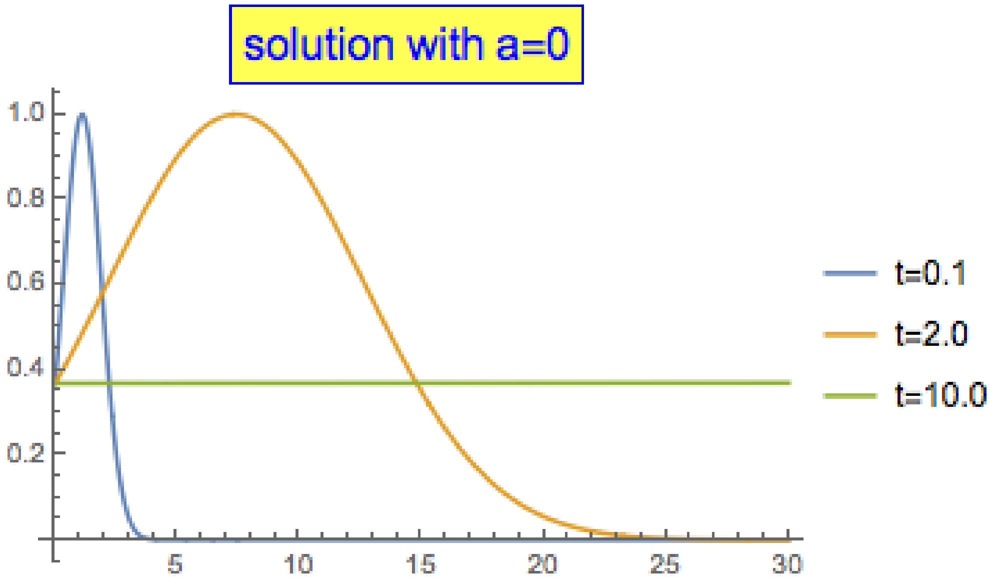


Figure 1: Solution with a=0 at different times

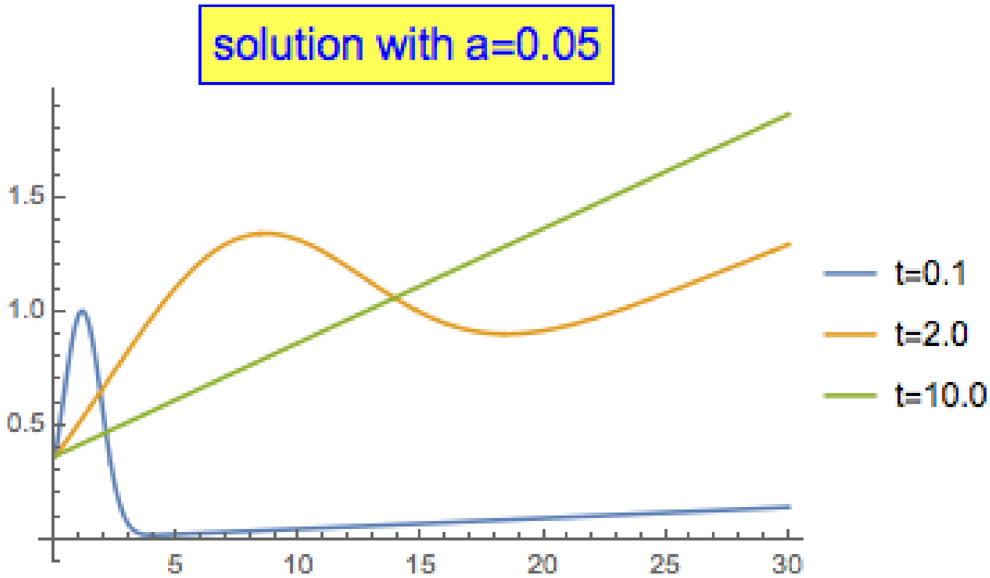


Figure 2: Solution with a=0.05 at different times

A 100 x 100 grid was created and boundary condition $\Phi = 0$ is imposed. The integration is only done on the interior grid points while the boundary grid points are fixed at 0. The index for x is 0,1,..,J and for y is 0,1,2,..,L where J=L=99. Spatial resolution Δ x = 1.5/100 and Δy =2/100.

Algorithm for the Gauss-Seidel and SOR method was derived from definition of derivative, respectively shown below:

$$u_j^{n+1} = \frac{1}{2(\Delta x^2 + \Delta y^2)}[\Big(u_{j+1,l}^n + u_{j-1,l}^{n+1}).\Delta y^2 + (u_{j,l+1}^n + u_{j,l-1}^{n+1}).\Delta x^2 - \Delta x^2.\Delta y^2\Big] \tag{12}$$

$$u_j^{n+1} = (1-\omega)u_{j,l}^n + \omega.\frac{1}{2(\Delta x^2 + \Delta y^2)}[\Big(u_{j+1,l}^n + u_{j-1,l}^{n+1}).\Delta y^2 + (u_{j,l+1}^n + u_{j,l-1}^{n+1}).\Delta x^2 - \Delta x^2.\Delta y^2\Big] \tag{13}$$

where $u_{j,l}^n$ is value of function Φ at grid point (x,y)=(j.Δx,l.Δy) at time step n. ω was chosen to be optimal value

$$\omega_{optimal} = 2.\frac{1 - sin(\frac{\Pi}{J+1})}{cos^2(\frac{\Pi}{J+1})} \approx 1.939 \tag{14}$$

In the code, two array were used to hold value of u at different time for any given iteration.

2.b Error as a function of time

See more under Fig 3

2.c Contour

See Fig. 4 and Fig. 5

2.d Convergence rate

In Gauss-Seidel method, the error reduce by a factor of 10^{-p} after N $\approx \frac{1}{4}pJ^2$ steps in $O(J^2)$, which is 10000 in this problem. SOR method, for the same precision, the number of steps required is $O(J)$ which is 100 faster.

3 Problem 3

code in 3.FTCS.cc

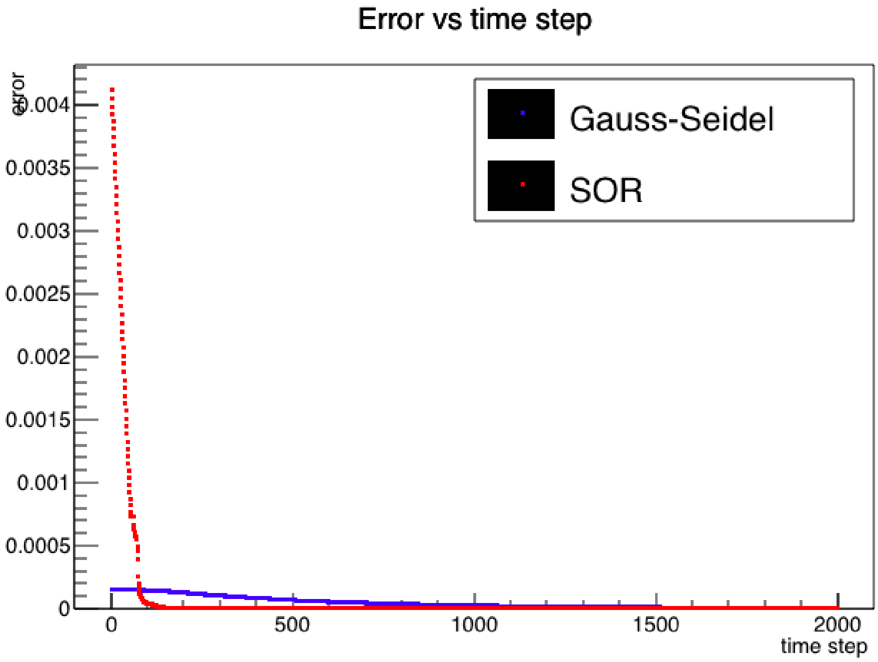
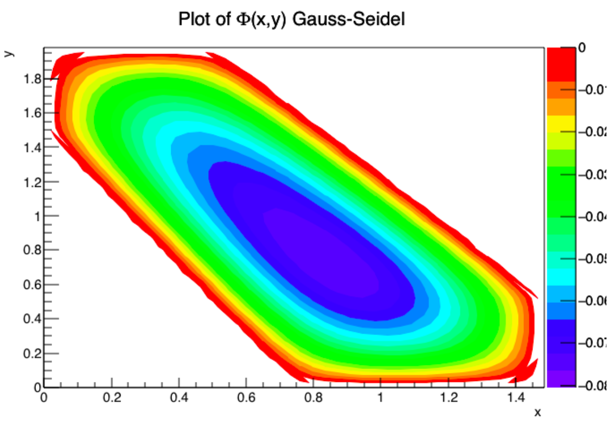


Figure 3: Error of the two method. SOR method starts out with larger error but converge much about 100 times faster



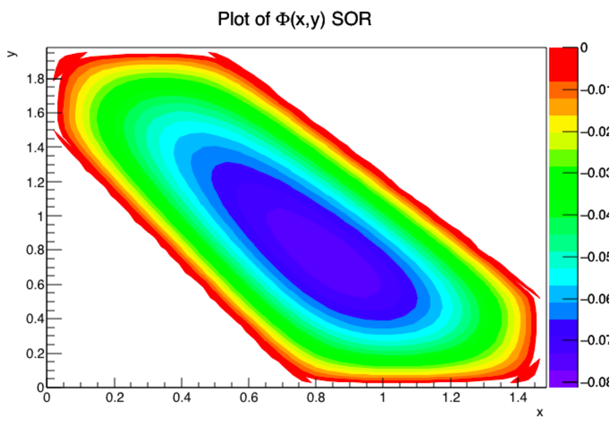


Figure 4: Contour plot of solution by Gauss-Seidel method

Figure 5: Contour plot of soloution by SOR method

3.a dependence of v on ρ the wave speed c(ρ)

The traffic flow equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v) = 0 \tag{15}$$

By chain rule

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial \rho}.\frac{\partial \rho}{\partial x} = 0 \tag{16}$$

The equation is now in the form of wave equation with wave speed c(ρ)

$$c(\rho) = \frac{\partial (\rho v)}{\partial \rho} = v + \rho\frac{\partial v}{\partial \rho} \tag{17}$$

given

$$v(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right) \tag{18}$$

the wave speed is then:

$$c(\rho) = v_{max}\left(1 - 2\frac{\rho}{\rho_{max}}\right) \tag{19}$$

Explanation: v(ρ) is the velocity of vehicles. When $\rho = \rho_{max}$, v(ρ)=0 i.e the traffic does not flow when the density is at maximum.

On the other hand, c(ρ) is the density wave velocity. When there is no traffic, i.e ρ=0, c(ρ) and v(ρ) are the same and equal v_{max}. When the density is half its maximum, the density wave doesn't travel c(ρ)=0 corresponding to "stable" traffic. When density is maximum, c(ρ) $= -v_{max}$ the density wave travel backward, corresponding to the traffic backs up at maximum vehicles velocity.

3.b Boundary condition

Let j be index of x, j run from 0,1,... to J. To set up periodic boundary condition in evolving ρ_j, if j=0, j-1 is set to J and if j=J, j+1 is set to 0.

3.c FTCS and LAX algorithm

Using c(ρ) derived above, FTCS algorithm is:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}.v_{max}(1 - 2\frac{u_j^n}{\rho_{max}})\big[u_{j+1}^n + u_{j-1}^n\big] \tag{20}$$

To derive LAX, perious time step u_j^n is replaced by its spatial average of u at two adjacent grid point.

$$u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - \frac{\Delta t}{2\Delta x}.v_{max}\Big[1 - 2\frac{u_{j+1}^n + u_{j-1}^n}{2.\rho_{max}}\Big]\left[u_{j+1}^n + u_{j-1}^n\right] \tag{21}$$

where u_j^n is value of ρ at time step n and grid point x=Δx.j and $\frac{\Delta t}{\Delta x}.v_{max}(1 - 2\frac{u_j^n}{\rho_{max}}) = \mu$ is the Courant number. With the choices of ρ_{max}=1.0, v_{max}=L/10, while Δx=L/100 and Δt=0.01, μ_{max}=0.1 and this ensures stability of the solution (which requires μ no larger than 1).

After 1000 steps, density element that travels at v_{max} makes exactly one round: 1000 $v_{max} = 1000 \times 0.01 \times L/10 = L$.

3.d 3D Plots

For analysis of 3D plots, see under Fig (7), (6) and (9).

4 Problem 4 KdV equation

4.a code implementation

For code, see 4.cc under main subfolder.

Taking from the lecture note, the coupled ODEs that needed to sovled for solution of KdV using Galerkin method is:

$$\dot{c_n} = -\sum c_{n-n'}c_{n'}ik' + \alpha ik^3 c_n \tag{22}$$

The LHS is a function of an array of 41 element corresponding to n from -20 to 20. The RHS is put into an explicit Runga-Kutta fourth-order solver and integrate forward intime to get solution at t=0,20,20,80,120,200. However, because $c_{-n} = c_n^*$, only c_n with non-negative n were sufficient to recover the solution. The initial value of c array is derived from initial condition. Because

$$Cos(\pi x/L) = \frac{e^{i\pi x/L} + e^{-i\pi x/L}}{2} \tag{23}$$

it can be deduced that $c_{-1} = c_1 = \frac{1}{16}$ while $c_n = 0$ for n $\neq \pm 1$.

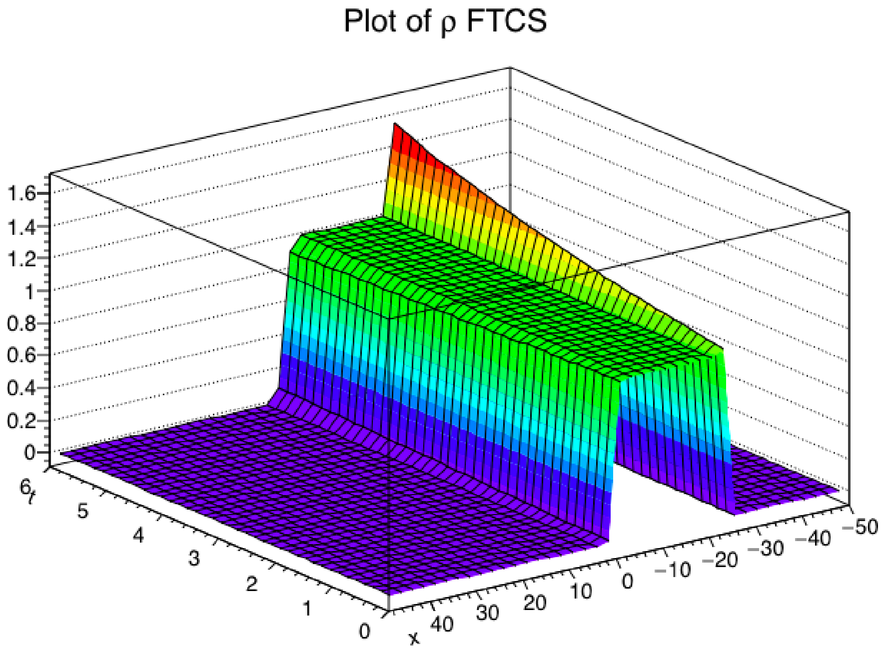


Figure 6: 3D Plot of the solution using FTCS. The solution start to break down after a few time step. The density becomes larger than maximum density, which is unphysical. FTCS scheme is not stable for solving this equation

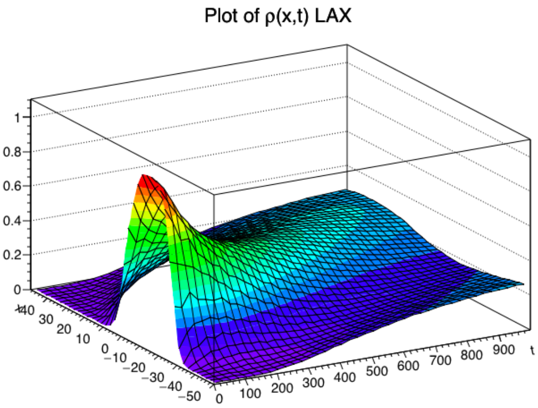


Figure 7: 3D Plot of the solution using LAX for one period of traffic. The solution is well behaved and is a lot smoother than FTCS using the same parameter

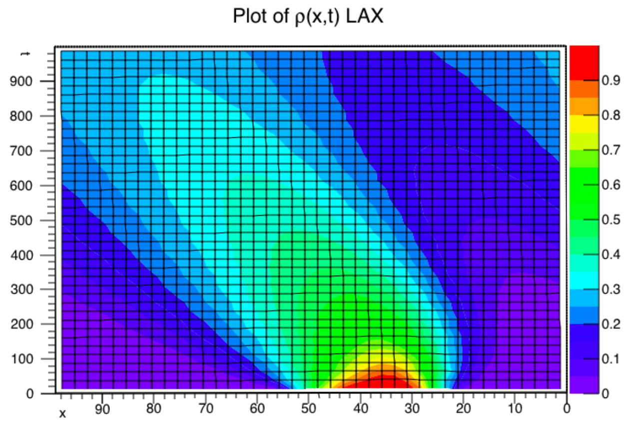


Figure 8: Contour plot. The density starts out as a square function but over time, due to dispersive velocity c(ρ), is distributed over spatial grid. However, the total density is conversed.

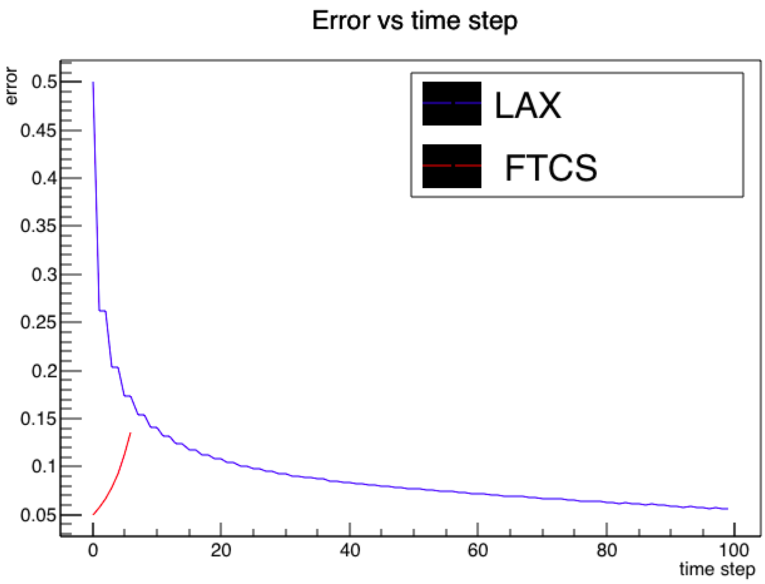


Figure 9: LAX has converging error while FTCS has error (which only plotted for several initial time steps) increase exponentially

4.b Plots

Because $c_{-n} = c_n^*$, the solution can be written as a real Fourier series of Cosine and Sine.

$$f(x, t = t_i) = \sum_{n=-20}^{20} c_n e^{\frac{in\pi}{L}} = Re(c_0) + \sum_{n=1}^{20} 2Re(c_n)cos(\frac{in\pi}{L}) - 2Im(c_n)sin(\frac{in\pi}{L}) \quad (24)$$

4.c Power Spectrum

$c_{-n} = c_n^*$ also lead two negative and positive k have the same power. $|c_n|^2$ was plotted against k_n for n from 0 to 20.

4.d

The power spectrum is started out as a 'delta function' because the initial wave have only one frequency k=π/L. This however no longer true over time when the wave disperses out due to α term and the power spectrum gets other frequency as well.

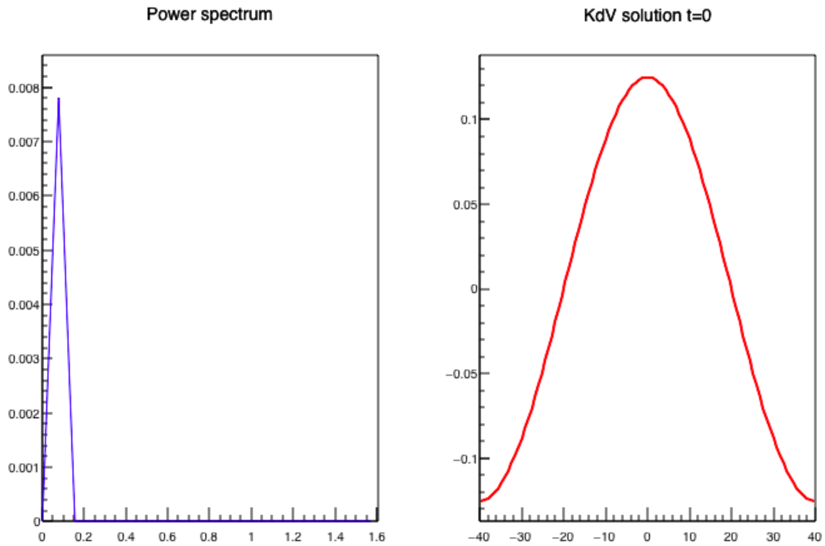


Figure 10: Solution at t=0, also the given initial function

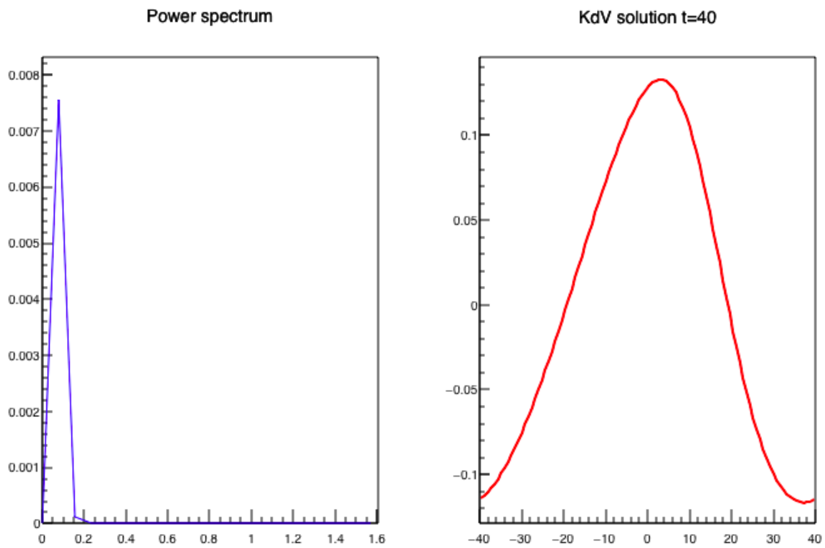


Figure 11: Solution at t=40

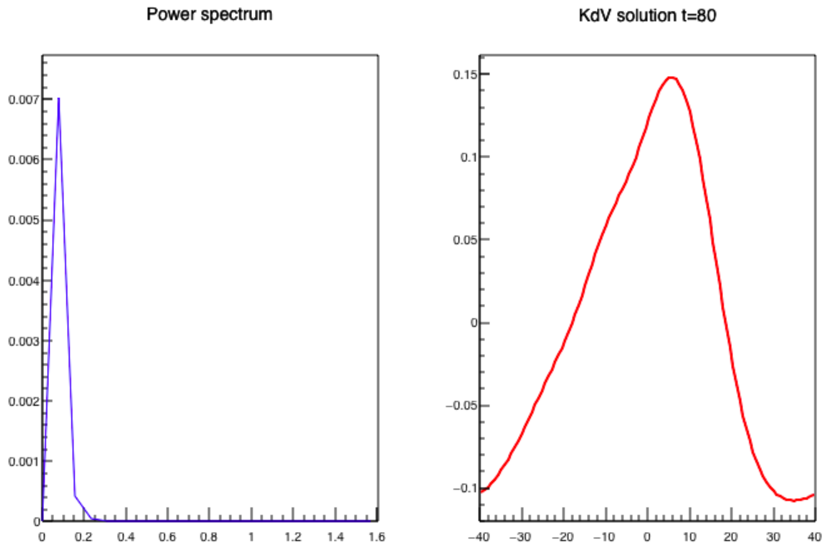


Figure 12: Solution at t=80

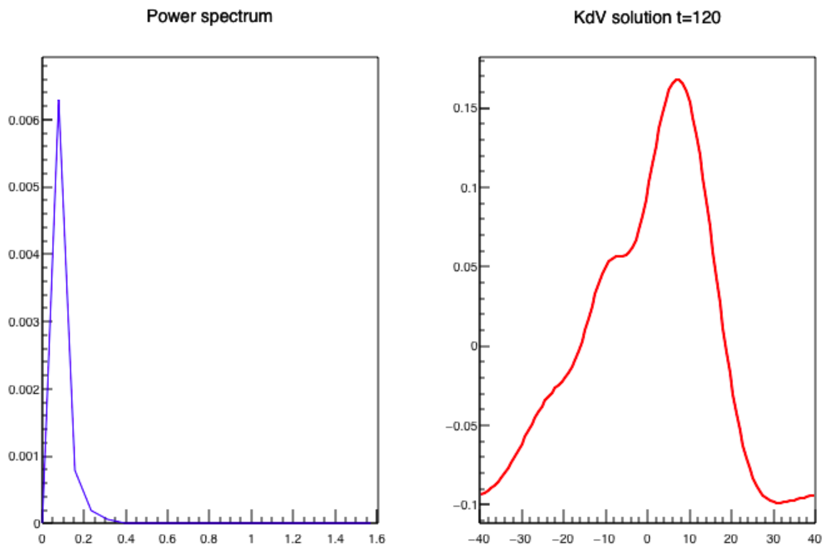


Figure 13: Solution at t=120

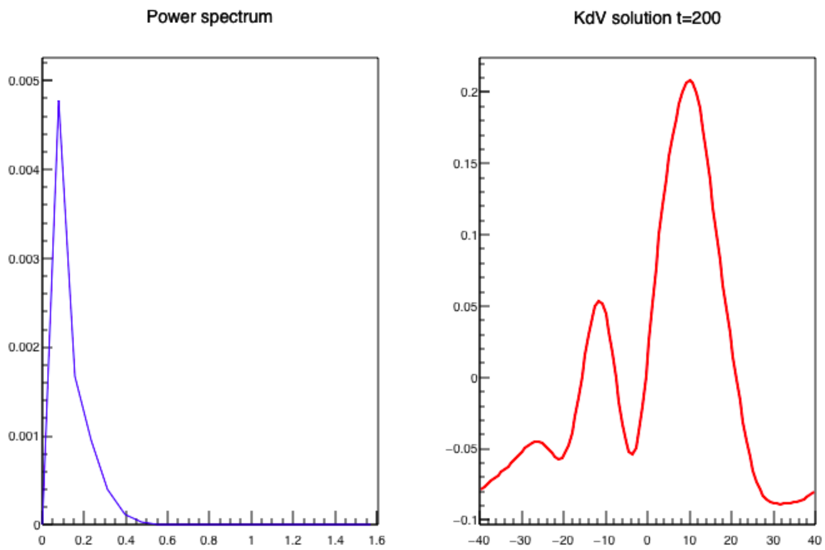


Figure 14: Solution at t=200

The solution plots reflects two effects: dispersive broadening and non-linear steepening. i.e over time, the wave is both broaden in width and steepened in amplitude. The peak moves to the right because the main mode has positive velocity.

In the KdV equation:

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

α controls strength of the dispersive term. When α decreases, the dispersive broadening effect is weaken and the wave is narrower. Meanwhile, the non-linear term f$\frac{\partial f}{\partial x}$ become more important, it steepens the wave and broadens the power spectrum. See Fig. (15)

References

[1] Roman Scoccimarro *Lecture notes*

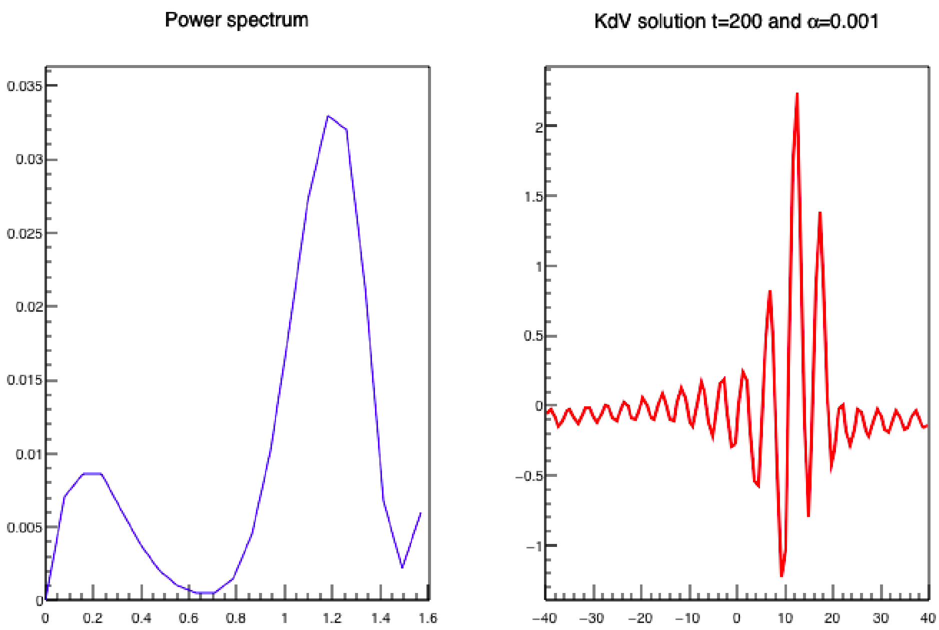


Figure 15: Solution with alpha = 0.001