Your Report Title

Roberto Lucchesi

July 23, 2024

Introduction

In this report we're going to analize the behavior of the upwind scheme for the solution of hyperbolic equations. We're going to test the scheme on three different initial conditions, a top hat function, a triangular function and a sine function, to observe the behavior of the scheme with different regularity of the initial condition. We're going to test the scheme with different Courant numbers to observe the effect of the speed of the signal on the solution.

The report is structured as follows:

- In the first chapter we're going to introduce the mathematical background of the problem, defining the hyperbolic equations and the upwind scheme, and its convergence to the solution through the Courant number.
- In the second chapter we're going to present the implementation of the upwind scheme and the results of the tests on the three initial conditions.
- In the third chapter we're going to test the effect of the Courant number on the solution.

Mathematical Background

Hyperbolic equations

A generic system of the form:

$$U_t + A(U)U_x + BU = F(x,t)$$

where:

- $U(x,t) \in R^k$
- A and $B \in R^{k \times k}$
- $F(x,t) : R^k \to R^k$

is said to be hyperbolic.

We are interested in the analysis of systems where $\mathbf{B = 0}$, $\mathbf{F(x,t) = 0}$ and $\mathbf{A}$ is diagonizable with real eigenvalues. In such cases, the system is decoupled and formed by *k* scalar equations. Infact, if $\mathbf{A}$ is diagonizable it exist a matrix $\mathbf{P}$ such that:

$$PAP^{-1} = \lambda_i I = \Lambda$$

and by sobstituting $W = PU$ we can rewrite:

$$W_t + \Lambda W_x = 0$$

representing a decoupled set of $\mathbf{k}$ scalar equations. For such systems the λ_i are called **characteristics velocities** of the system.

The interpretation of those velocities are more evident when k = 1. The single scalar hyperbolic equation in the form

$$u_t + au_x = 0 \quad \text{with} \quad u(x,0) = u_0(x)$$

The solution of the initial value problem for such equation is of the form

$$u(x,t) = u_0(x - at)$$

a traslation of the initial condition on the right if $a > 0$ (**upwind**) on the left if $a < 0$ (**downwind**) with a velocity **a**. We can identify the straight line of equation $\xi = x - at$ where:

$$u(x,t) = u_0(\xi) = \text{ const along } \xi = x - at$$

Meaning that, if we have a given initial state $u_0(x)$, to retrieve the solution at a time t we just need to follow the characteristic lines.
Hyperbolic equations are useful for us engineers because they describe a wave-like phenomenon, a perturbation in the initial condition propagates along the space-time with a finite velocity $\mathbf{a}$, traveling along the characteristic lines. This special form also allows use to drop any request on the continuity of $u_0(x)$ and its derivatives, allowing us to solve the problem in a weak sense.

Transport equation

The Burgers equation is a simple example of a hyperbolic equation. It is a scalar equation that modelizes a convective-diffusive phenomenon of the form:

$$u_t + a(u)u_x - \nu u_{xx} = 0$$

where a(u) is the convective speed (or transport speed) and ν is the cinematic viscosity (or diffusion coefficient). We're interested in the non-viscous linear model, so we consider a = cost and $\nu = 0$. We can then rewrite the equation as:

$$u_t + au_x = 0 \tag{1}$$

also called, the transport equation.
For this kind of problem, we only need one boundary condition, with the caveat that the support of **u** must be compact. Such boundary condition can be expressed as:

$$u(0,t) = 1 \quad \text{or} \quad u_x(0,t) = 0$$

Numerical methods

We're going to need some numerical method to solve the problem in a discretized domain. In order to do that, we need to first define how the domain is discretized. For the spatial domain, we consider a uniform grid with a step Δx and we define the nodal points of the grid as:

$$x_i = i\Delta x \quad \text{with} \quad i = 0, 1, 2, ..., N-1$$

For the temporal domain, the timestep Δt is assumed constant throughout the simulation.

The solution of the propagation of one node of the initial condition $u(x_i, 0) = u_0(x_i)$ for a time $\tau = \Delta t$, is given by:

$$u(x_i, \tau) = u(x_i - a\tau, 0)$$

If we assume that $a\tau < \Delta x$, meaning that $u(x_i - a\tau, 0)$ is within the node x_{i-1} and x_i on the *x* axis, we can linearly interpolate between these two known solutions to retrieve the value of $u(x_i, \tau)$. The interpolation is done on the segment containing the abscissa, and for the two adiacent intervals we can build the slope, function of a parameter $\sigma \in$ [-1,1] as:

$$\Delta u = \frac{1+\sigma}{2}\Delta_{i-\frac{1}{2}}u - \frac{1-\sigma}{2}\Delta_{i+\frac{1}{2}}u$$

with

$$\Delta_{i-\frac{1}{2}}u = \frac{u_i - u_{i-1}}{\Delta x} \quad \text{and} \quad \Delta_{i+\frac{1}{2}}u = \frac{u_{i+1} - u_i}{\Delta x}$$

The slope is then used to interpolate the value of $u(x_i, \tau)$ as:

$$u(x) = u_i + \Delta u(x - x_i) \quad \text{with} \quad x \in [x_{i-1}, x_{i+1}]$$

By solving the equation for $x = x_i - a\tau$ we find:

$$u(x_i, \tau) = u_i^0 - a\tau\Delta u^0$$

And in a more general form, between two time steps t^n and t^{n+1}, such that $t^{n+1} = t^n + \Delta\tau$ we can write:

$$u_i^{n+1} = u_i^n - a\tau\Delta u^n$$

By choosing an adequate value for σ we can obtain different schemes for the interpolation. The schemes of interest are the **upwind**(**downwind**) scheme

$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{\Delta x}(u_j^n - u_{j-1}^n) \quad \text{with} \quad \sigma = 1 \quad \text{(Upwind)}$$
$$u_i^{n+1} = u_i^n - \frac{a\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) \quad \text{with} \quad \sigma = -1 \quad \text{(Downwind)}$$

The value $c = a\frac{\Delta t}{\Delta x}$ is called **Courant number** and it represents the ratio between the speed of the signal and the "speed" of the grid. This coefficient will be used to determine the stability of the scheme. Asking this coefficient to be within 0 and 1 assures that the signal does not propagate faster than the grid, performing an interpolation between u_j^n, u_{j-1}^n and u_{j+1}^n.

Courant number and stability of the scheme

To determine if the aproximate solution we compute is good enough to represent the exact solution we need to evaluate the **convergency** of the scheme to the real solution. That is, if the global error:

$$\|\varepsilon(U^n, u^n, a, \Delta x, \Delta t)\| \to 0 \quad \text{for} \quad \Delta x, \Delta t \to 0$$

Without going too deep into the mathematical details, several theorems like the **Lax-Richtmyer theorem** assure us that for linear problem as in (1) the scheme is convergent if it is **stable** and **consistent**. This proof of convergency works both ways, so if the scheme is not stable it cannot converge to the solution.

We basically need to prove that:

$$\varepsilon^{n+1} = -\tau e^n + \tau S\varepsilon^n$$

with

- ϵ^n the truncation error at step n, must tend to 0 for $\Delta x, \Delta t \to 0$ to assure **consistency**.
- S an operator that must not amplify previous step errors, so it shall be bounded to assure **stability**.

Consistency - Approximation error

With the discretization of the domain we inevitably introduce an error on the solution. We need to evaluate the truncation error ϵ and assure that

$$\epsilon \to 0 \quad \text{for} \quad \Delta x, \Delta t \to 0$$

After several steps of derivation, we can find that the truncation error for the **Upwind** scheme is:

$$\epsilon_n = -a\frac{\Delta x}{2}(1 - c)u_{xx} + o(\Delta x^2)$$

So the scheme has an accuracy at the first order. The coefficient that drives the sign of the error is the **(1 - c)** term. If the coefficient multipling the second derivative is positive, an anti-diffusive phenomenon is present, meaning that the scheme amplifies the initial solution. We want a dissipative scheme, so we need to assure that:

$$(1 - c) > 0 \quad \text{that is} \quad 0 < c < 1$$

For the **Downwind** scheme we derive:

$$\epsilon = a\frac{\Delta x}{2}(1 + c)u_{xx} + o(\Delta x^2)$$

The Courant number is now negative being $a < 0$. The scheme is stable for $-1 < c < 0$. This analysis is exact for problems with a regular $u_0(x)$, but in the case of discontinuities, the convergency is inferior thus the truncation error affects the solution more. The Courant number must be chosen carefully to assure the stability of the scheme.

Stability

To check for the stability, we need to check that

$$\|u_j^{n+1}\| \leq \|u_j^n\|$$

for an adequate norm. By picking the L_1 norm, defined as:

$$L_1 \ : \ \|u\|_1 = \sum_{j=1}^{N} |u_j|$$

Substituting for the upwind method we find:

$$\sum_{j=1}^{N} |u_j^{n+1}| = \sum_{j=1}^{N} |(1-c)u_j^n + cu_{j-1}^n| \leq \sum_{j=1}^{N} (1-c)|u_j^n| + c|u_{j-1}^n|$$

last term valid because $0 < c < 1$. By using the useful property that the solution only depends on its initial value, we can write:

$$\sum_{j=1}^{N} |u_j^{n+1}| \leq \sum_{j=1}^{N} |u_j^n|$$

resulting verified the stability. The same analysis can be done for the downwind scheme, with the only difference that the Courant number is negative. So we can now assure that the scheme is stable and convergent, and the solution we compute is a good approximation of the real solution of the hyperbolic equation in ..

The L_1 norm gives us also a good indicator of the goodness of the algorithm. Infact we now know that this norm shall remain constant for each temporal step.

Another useful norm that will get computed is the **Total Variation** of the solution, defined as:

$$TV \ : \sum_{j=1}^{N} |u_j - u_{j-1}|$$

That measures, as its name says, the variation of the solution along the grid. This norm is useful to check the diffusion of the solution, and to check the dissipative property of the scheme to eliminate discontinuities in the solution.

Implementation and results

Study cases

We're going to test the upwind scheme on three different cases of initial conditions:

- A top hat function
- A triangular function
- A sine function

Those are chosen to test the scheme with several conditions on regularity and continuity of the initial condition. Grid dimension and temporal step are defined as:

$\Delta x = 0.01$

$\Delta t = 0.005$

a = 1

so that the Courant number is $c = 0.5$.

Each disturbance has length L = 1 and is propagated for 2000 steps, or 10s in real time.

Top hat function

The top hat function is defined as:

$$u(x,0) = \begin{cases} 2 & \text{if} \quad 1 \leq x \leq 2 \\ 1 & \text{elsewhere} \end{cases}$$

This function presents a jump discontinuity, thus not C^0 at $x = 1$ and $x = 2$, where the slope is undefined.

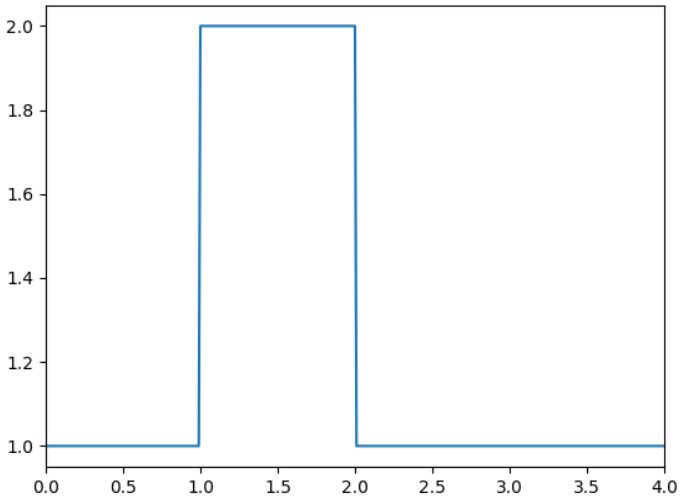

Figure 1: Top hat function

Triangular function

The triangular function is defined as:

$$u(x,0) = \begin{cases} 2x-1 & \text{if} \quad 1 \leq x \leq \frac{3}{2} \\ -2x+5 & \text{if} \quad \frac{3}{2} \leq x \leq 2 \\ 1 & \text{elsewhere} \end{cases}$$

This function is C^0 but not C^1 as it presents a jump in the first derivative at $x = \frac{3}{2}$.

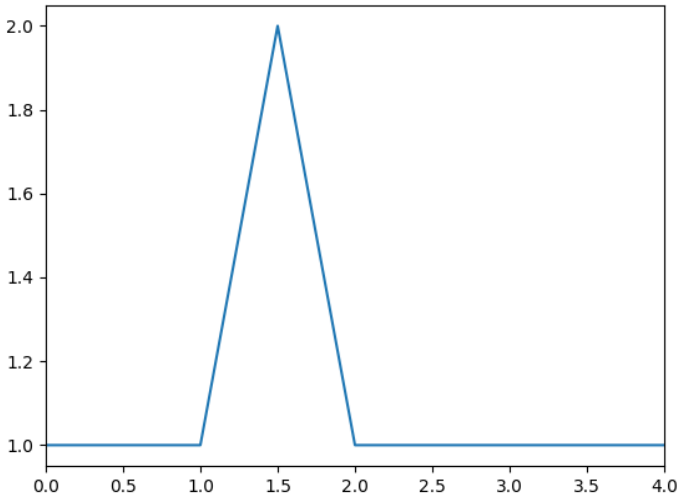

Figure 2: Triangular function

Sine function

The sine function is defined as:

$$u(x,0) = \begin{cases} -\sin(\pi x/L) + 1 & \text{if} \quad 1 \leq x \leq 2 \\ 1 & \text{elsewhere} \end{cases}$$

This function is C^∞ in its domain.

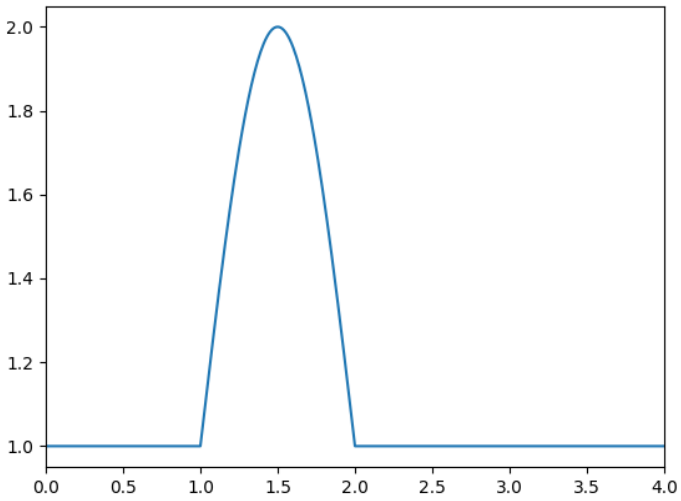

Figure 3: Sine function

Upwind scheme implementation

The upwind scheme is implemented in Python, using the numpy library for the array manipulation. The code is the following:

```
def propagate(u, dx, dt, lb, a=1, n_steps=1):
    c = a*dt/dx

    if np.abs(c) > 1:
        print(f"WARNING: CFL condition not met: {c}")
    u_prop = np.zeros((n_steps, len(u)))
    u_prop[0,:] = u
    for i in range(1, n_steps):
        u_new = u_prop[i,:]
        u_new[0] = lb
        for j in range(1,len(u)):
            u_new[j] = u[j] - c*(u[j] - u[j-1])
        u = u_new
    return u_prop
```

We can now test the scheme on the three cases we defined before.

Results

Top hat

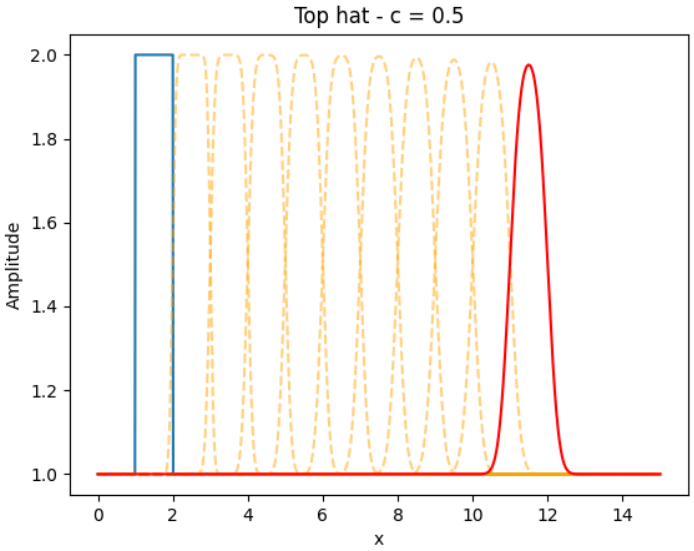

Figure 4: Top hat function after 10 steps

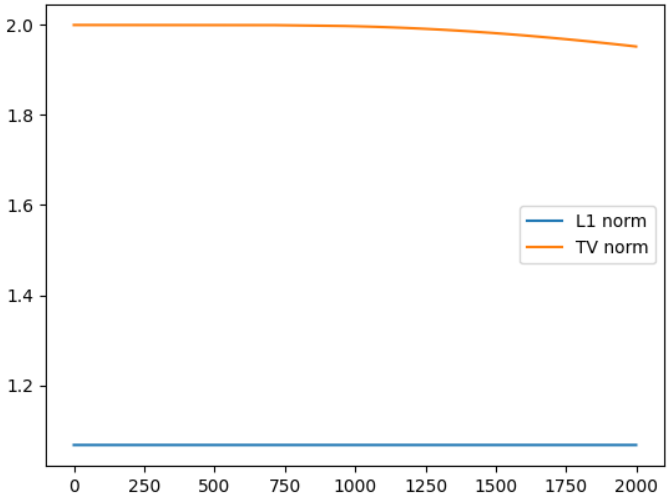

Figure 5: Total variation of the top hat function

From the top hat propagation, we observe how the flat top is diffused. The complete diffusions, meaning the elimination of every discontinuity, requires several steps. We can also see how the total variation is constant until the top hat is unflattened, then it starts to decrease. We infact note that in the top flat part, the second derivative

is null, so no diffusion of the peak occurs until the top hat is completely flattened.

Wizard hat

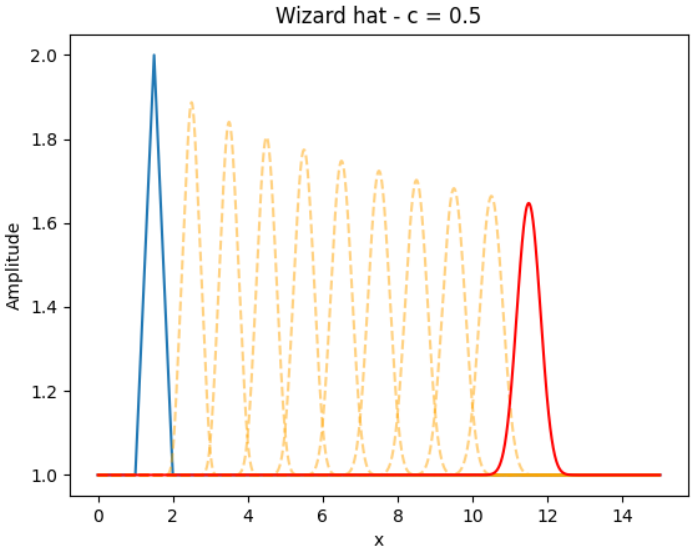

Figure 6: Top hat function after 10 steps

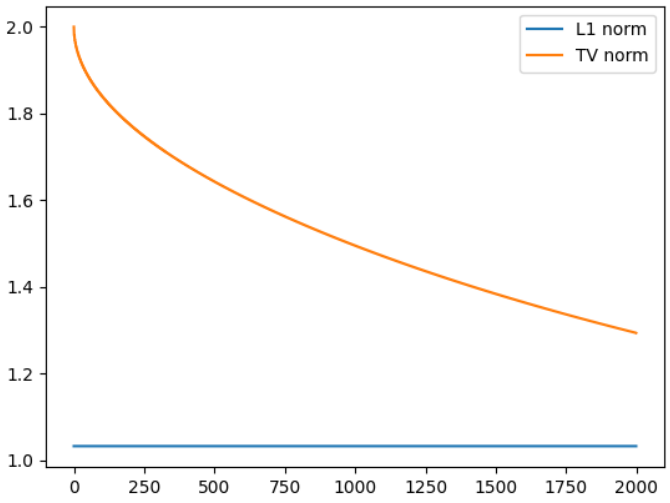

Figure 7: Total variation of the top hat function

From the wizard hat propagation instead, the peak is instantly diffused in the first step. We note that the total variation in this case is almost vertical at the start, infact the second derivative at the top tends to infinity, being a cusp. The diffusion of the peak is then immediate, and the total variation decreases very rapidly.

Sine function

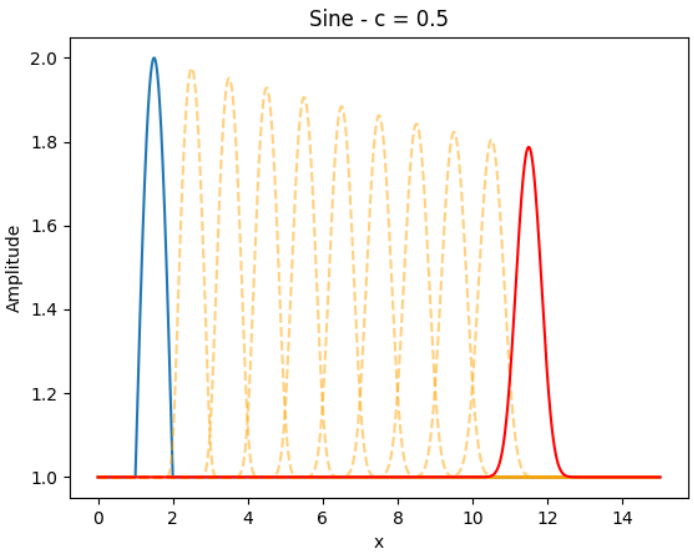

Figure 8: Sine function after 10 steps

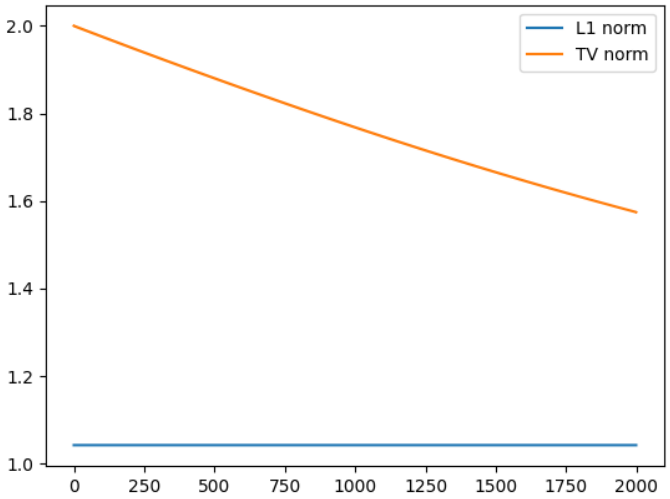

Figure 9: Total variation of the sine function

Comments on the results

By observing the results, it is clear how the dissipative "speed" of the scheme is related to the reguarity of the function, in particular on the second derivative.

The **top hat** wont start to diffuse its peak until this is unflattened, meaning that we need to wait for the second derivative to be not null to start the peak diffusion. This is evident in the total variation plot, where the value is constant until the peak is unflattened.

The **wizard hat** instead, with its cusp at the top, starts to diffuse the peak immediately, and the total variation decreases very rapidly, almost vertically at the beginning. This because the second derivative tends to infinity at the peak (infinite speed of change of the first derivative where a jump is present)

The **sine function** instead, being C^∞, has a smooth second derivative, and the diffusion of the peak is immediate, but not as immediate as the wizard hat. We can infact observe the total variation decrease almost linearly.

Effect of the Courant number

We can now test the effect of the Courant number on the solution, using the upwind scheme on the sine function, due to its regularity, with different Courant numbers. We're going to use the not yet introduced simmetric property of the Courant number about the value 0.5, so we test several values of c in the range [0.1,0.4].
In order to perform a sensible analysis, we fix the number of steps and timestep, fixing then the total time of the simulation.
We choose:

$\Delta t = 0.005$

a = 1

n_steps = 2000

then, $t_{tot} = 10$s and the grid is defined as:

$$\Delta x = \frac{\Delta t}{c}$$

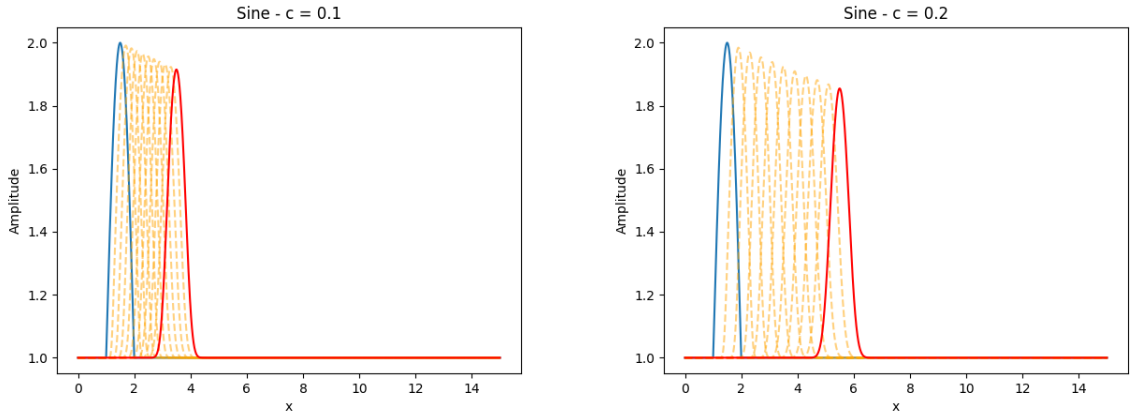

Figure 10: Sine function after 10 steps with c = 0.1 and c = 0.2

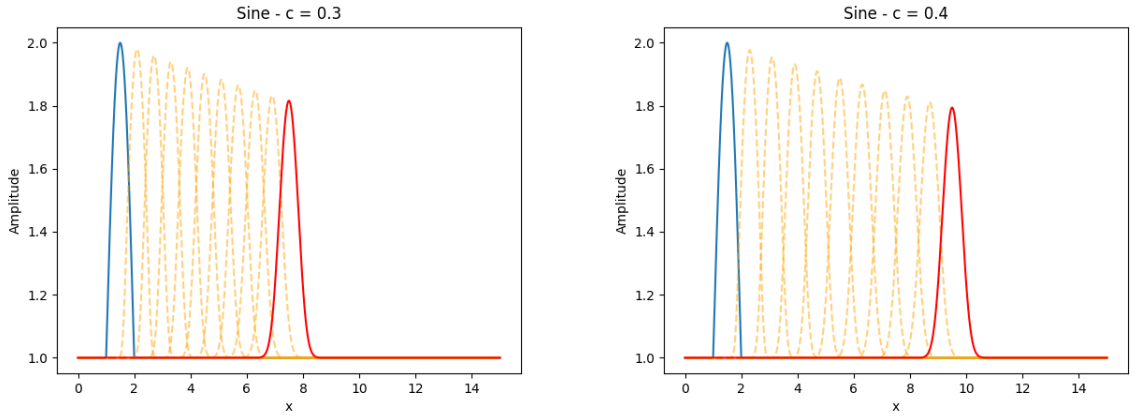

Figure 11: Sine function after 10 steps with c = 0.3 and c = 0.4

Without introducing the Von Neumann stability analysis, but just by observing the results, we can note how the signal propagates faster and diffuses more as the Courant number approaches $\frac{1}{2}$.

Nonlinear transport equation

Euler equations

We now put on the big boy pants and we start approaching the theory to the real world. We're going to analyze the **Euler equations**, a system of hyperbolic equations that modelize the flow of an ideal inviscid fluid, under the hypotesis of isentropicity.

Mass conservation

The first equation of the system is the mass conservation equation, that is:

$$\frac{D\rho}{Dt} + \rho\nabla \cdot u = 0$$

By using the Reynolds trasport theorem, we can rewrite the equation as:

$$\rho_t + u\rho_x + \rho u_x = 0$$

Momenta conservation

The second equation of the system is the momenta conservation equation, that is:

$$\rho\frac{Du}{Dt} + \nabla p = 0$$

that can be rewritten as:

$$\rho(u_t + uu_x) + p_x = 0$$

Energy conservation

For an isentropic fluid, the entropy conservation states that:

$$\frac{Ds}{Dt} = \frac{Dp}{Dt} - a^2\frac{D\rho}{Dt} = 0$$

where s is the entropy of the fluid, a is the speed of sound and p is the pressure. We can rewrite the equation as:

$$p_t + up_x - a^2\rho u_x = 0$$

Euler equations

We can now write the system of equations as:

$$\begin{cases} \rho_t + u\rho_x + \rho u_x = 0 \\ u_t + uu_x + \frac{1}{\rho}p_x = 0 \\ p_t + up_x + \rho a^2 u_x = 0 \end{cases}$$

We can rewrite the system in a matrix form with:

$$U = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \quad A = \begin{bmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \gamma p & u \end{bmatrix}$$

where γ is the adiabatic index of the fluid. The system can be rewritten in an hyperbolic form as:

$$U_t + A(U)U_x = 0$$

By solving the eigenvalue problem on A(U), we can find three distinct and real eigenvalues, that are:

$$\lambda_1 = u - a \quad \lambda_2 = u \quad \lambda_3 = u + a$$

These are the characteristic velocities of the system, and they represent the speed of the signal in the fluid. The system is hyperbolic, and the solution of the system is a wave-like phenomenon, where the perturbation in the fluid propagates with a finite velocity. By searching for the eigenvectors, we define the wave variables as:

$$d\sigma_1 = dp - \rho a du \quad \text{along u-a} d\sigma_2 = d$$