Assignment in Numerical Analysis

Guy Raveh and Andrea Suárez Segarra

November 2025

1 Viscous Burgers equation 1D

1.1 Description - Andrea

The viscous Burgers equation in one dimension is a nonlinear equation described as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{1}$$

The vector-field $u(x,t)$ refers to the velocity of the fluid and ν is the kinematic viscosity. The left-side of equation (1) is the advection term, which describes the propagation of a wave, and the right-side is the diffusive term. When the viscosity ν is 0, the equation is known as inviscid Burgers and can exhibit shocks, which is not physically realistic. In this project we study equation (1), with positive viscosity and periodic boundary conditions.

1.2 Linearisation - Guy

In some contexts where the nonlinear behaviour of equation (1) poses a problem, we linearize it by assuming:

$$u(t,x) = U + \epsilon u_1(t,x), \;\; \textit{with } U, u_1, \partial_x u_1 = O(1) \tag{2}$$

Under this assumption, up to a term with order of magnitude $O(\epsilon^2)$, the Burgers equation gives the same behaviour as the linear advection-diffusion equation:

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{3}$$

with U now being a constant.

1.3 Relevance for Climate Modelling - Guy

The higher-dimensional equivalent of equation (1) serves as a simplified model of homogeneous, incompressible fluid dynamics, achieved by dropping the pressure term $-\nabla p$ in the momentum equation, as in equations (3.1) and (3.3) in [FG]:

$$\partial_t u + u \cdot \nabla u = -\nabla p + \nu \Delta u \tag{4}$$

In addition, it can be used directly to model phenomena that involve both nonlinear advection and diffusion. [Bas02] shows such a model for the behaviour of soil moisture, and [RLDL20] shows one for forecasting rainfall.

2 Chosen numerical methods

2.1 Discretisation and the FTCS Scheme - Andrea

We discretise the space with a set of points $\{x_j\}_{j=0}^{N_x}$. We will then work with the state of the solution in $N_x + 1$ spatial points, with $L := x_{N_x} - x_0$ being the length of the one dimensional space. The constant spatial resolution is denoted Δx, so that $x_{j+1} = x_0 + j\Delta x = j\Delta x$. The time

grid spans from $t = 0$ to the final time $T = N_t\,\Delta t$, using a fixed time step Δt and a variable number of time points N_t. The solution of the numerical scheme is $(u_j^{(n)})$, $j \in \mathbb{N}$, $n \in [0, N_t]$. The periodic boundary conditions can be expressed as:

$$u_k^{(n)} = u_{N_x+k}^{(n)}, \forall n \in [1, N_t], k \in \mathbb{Z} \tag{5}$$

We propose to implement an FTCS (Forward in Time, Centered in Space) scheme to solve this non-linear equation, described as follows. For each time step $n+1$, and each spatial point x_j, the velocity $u_j^{(n+1)}$ is computed as:

$$u_j^{(n+1)} = u_j^{(n)} - \frac{\Delta t}{2\Delta x} u_j^{(n)} (u_{j+1}^{(n)} - u_{j-1}^{(n)}) + \frac{\Delta t \cdot \nu}{(\Delta x)^2} (u_{j+1}^{(n)} - 2u_j^{(n)} + u_{j-1}^{(n)}) \tag{6}$$

$$= u_j^{(n)} - \frac{r}{2} \cdot u_j^{(n)} (u_{j+1}^{(n)} - u_{j-1}^{(n)}) + d(u_{j+1}^{(n)} - 2u_j^{(n)} + u_{j-1}^{(n)}) \tag{7}$$

Here, $r := \frac{\Delta t}{\Delta x}$ stands for the temporal-spatial ratio and $d := \frac{\Delta t \cdot \nu}{(\Delta x)^2}$ is the dimensionless kinematic viscosity. In the linearised case (3), we also denote $c := Ur$, the Courant number.

We chose this explicit scheme because it is mass-conservative, which was presented in the lecture notes [Wel25] as a desirable property of a numerical scheme.

In the numerical implementation of the scheme, L, T, N_t, N_x and ν are chosen, while Δx and Δt are computed inside the scheme, ensuring that the number of grid points are integer, and that the grid size and length of the experiment are also fixed. This allows to do consistent experiments.

2.2 Other Numerical Schemes - Guy

While our experiments where focused on FTCS, we implemented a few other schemes:

- A fully spectral scheme for the linearised equation 3, given by:

$$u(t, \cdot) = \mathcal{F}^{-1}\left[\mathcal{F}\left[u(0, \cdot)\right] e^{-\left(\nu k^2 + iUk\right)t}\right] \tag{8}$$

  where $\mathcal{F}$ is the discrete Fourier transform.

- A pseudo-spectral scheme where the Fourier transform is only used to compute spatial derivatives:

$$u^{(n+1)} = u^{(n)} - \Delta t \left(u^{(n)} \mathcal{F}^{-1}\left[ik\mathcal{F}\left[u^{(n)}\right]\right] - \nu \mathcal{F}^{-1}\left[-k^2 \mathcal{F}\left[u^{(n)}\right]\right]\right) \tag{9}$$

- A CTCS (leapfrog) scheme:

$$u_j^{(n+1)} = u_j^{(n-1)} - \frac{\Delta t}{\Delta x} u_j^{(n)} (u_{j+1}^{(n)} - u_{j-1}^{(n)}) + \frac{2\Delta t \cdot \nu}{(\Delta x)^2} (u_{j+1}^{(n)} - 2u_j^{(n)} + u_{j-1}^{(n)}) \tag{10}$$

  where the first time step is done by another customisable scheme.

Since the time frame of the assignment did not allow getting interesting results from these other methods, the results that follow are based on the FTCS scheme only, unless otherwise specified.

2.3 Expected Conservation properties - Andrea

As stated in subsection 2.1, the FTCS scheme is expected to preserve mass. Indeed, for this scheme, the total mass at time $n+1$ is:

$$\begin{aligned}
M^{(n+1)} &= \sum_{j=0}^{N_x-1} u_j^{(n+1)} \\
&= \sum_{j=0}^{N_x-1} u_j^{(n)} - \frac{r}{2} \sum_{j=0}^{N_x-1} u_j^{(n)}(u_{j+1}^{(n)} - u_{j-1}^{(n)}) + d \sum_{j=0}^{N_x-1} (u_{j+1}^{(n)} - 2u_j^{(n)} + u_{j-1}^{(n)}) \\
&= M^{(n)} - \frac{r}{2} \left(\sum_{j=0}^{N_x-1} u_j^{(n)} u_{j+1}^{(n)} - \sum_{j=0}^{N_x-1} u_j^{(n)} u_{j-1}^{(n)} \right) \\
&+ d \left(\sum_{j=0}^{N_x-1} (u_{j+1}^{(n)} - u_j^{(n)} - \sum_{j=0}^{N_x-1} u_j^{(n)} - u_{j-1}^{(n)}) \right) \\
&= M^{(n)} - \frac{r}{2}(u_{N_x-1}^{(n)} u_{N_x}^{(n)} - u_0^{(n)} u_{-1}^{(n)}) + d(u_{N_x}^{(n)} - u_{N_x-1}^{(n)} - u_0^{(n)} + u_{-1}^{(n)}) \\
&= M^{(n)}
\end{aligned}$$

Where the last equality holds due to the boundary conditions (Equation 5).

The energy, however, is not conserved. As stated in the literature [TD10], the dissipative term of equation 1 causes a decrease in kinetic energy.

2.4 Accuracy - Guy

Since we have the local truncation errors:

$$\partial_t u(t,x) = \frac{1}{\Delta t} \left(u(t+\Delta t, x) - u(t,x) \right) + O(\Delta t) \tag{11}$$

and:

$$\partial_x u(t,x) = \frac{1}{2\Delta x} \left(u(t, x+\Delta x) - u(t, x-\Delta x) \right) + O\left(\Delta x^2\right) \tag{12}$$

we expect the error $|u_{\text{scheme}}(T,x) - u_{\text{real}}(T,x)|$ to be bound by the maximum between a linear term in Δt, and a quadratic term in Δx.
If both resolutions are varied together with the relationship $\Delta t = C\Delta x^2$ for some constant C, we expect the error to be $O(\Delta x^2)$.

2.5 Stability conditions for the linear advection-diffusion equation - Andrea

Von-Neumann stability analysis cannot be used in non-linear schemes. A common solution (see [VD18]) is to linearise Burgers' equation, as discussed in section 1.2, getting similar behaviour to the advection-diffusion equation (3). This second equation is linear so Von Neuman's stability analysis can be done and provide an approximation of the stability conditions for viscous Burgers' with nearly constant initial state.

The proposed FTCS scheme applied on the linear advection-diffusion euqation (3) is:

$$u_j^{(n+1)} = u_j^{(n)} - \frac{r}{2} \cdot U(u_{j+1}^{(n)} - u_{j-1}^{(n)}) + d(u_{j+1}^{(n)} - 2u_j^{(n)} + u_{j-1}^{(n)}) \tag{13}$$

Assume $u_j^{(n+1)} = A u_j^{(n)}$ and that at each time, the solution of the equation can be expressed as a sum of wavenumbers: $u_j^{(n)} = A^{(n)} \cdot \sum_{-\infty}^{+\infty} e^{ikj\Delta x}$. Then, doing some computations derived

from [VD18], we get the following condition for stability:

$$r^2U^2 \leq 2d \leq 1 \tag{14}$$

$$\implies \Delta t \leq \min\left\{\frac{2\nu}{U^2}, \frac{\Delta x^2}{2\nu}\right\} \tag{15}$$

Therefore, we expect our scheme to be stable for nearly constant initial velocity $u^{(0)}$ and parameters $\Delta t, \Delta x, \nu$ verifying inequality (15).

3 Experiments and Results

3.1 Stability in the Linearised Case - Guy

We keep a constant N_t while varying N_x and ν. Taking values of c and d from a grid, we calculate the corresponding N_x, round it to an integer, calculate ν, and recalculate c according to the following equations which stem from the definitions of c and d:

$$c = U\frac{TN_x}{LN_t} \tag{16}$$

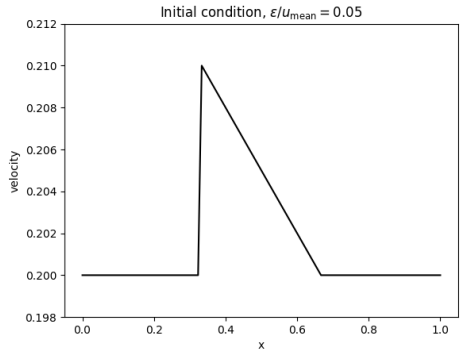
$$\nu = d\frac{L^2N_t}{TN_x^2} \tag{17}$$

For each such pair of N_x and ν, we run the scheme and check the relative increase in total variation of the solution:

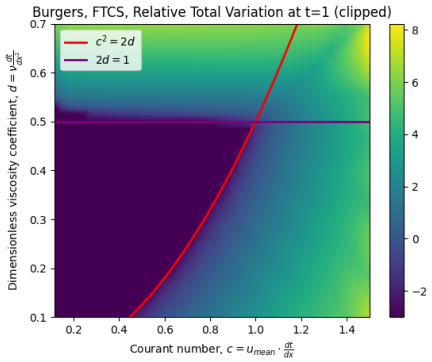
$$v_{rel} = \frac{v_T - v_0}{v_0} \tag{18}$$

$$v_t := \Sigma_{\text{cyc}} |u(t, x_{k+1}) - u(t, x_k)| \tag{19}$$

Figure 1: Total variation of solution for different values of c and d



(a) Initial condition $u(0, x) = U(1 + \epsilon u_1(x))$, where $U = 0.2$, $\epsilon = 0.05$, and u_1 goes linearly from 1 to 0 on the interval $\left(\frac{1}{3}, \frac{2}{3}\right)$ and is 0 outside it. Length scale $L = 1$.



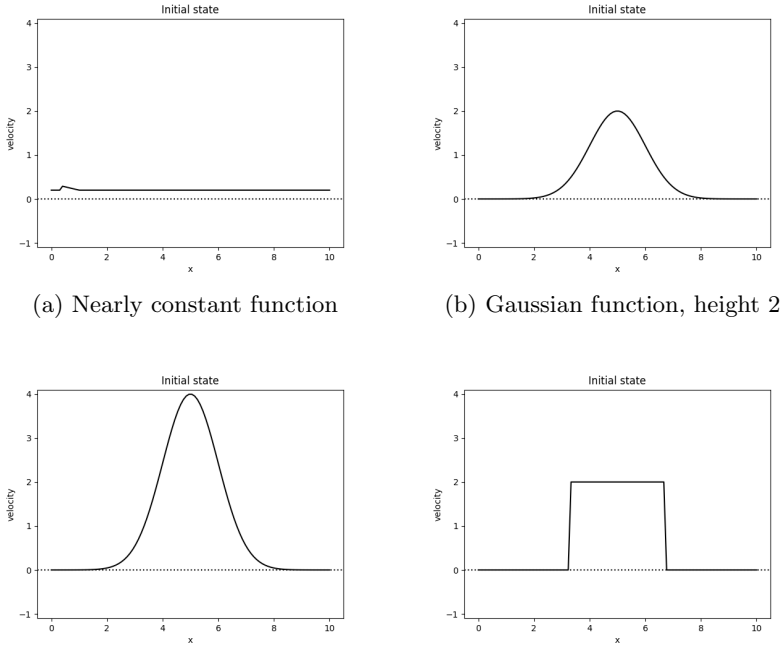
(b) Heatmap of $log(v_{rel})$, taken at time $T = 1$ and clipped to the interval $[-3, 10]$. Negative v_{rel} treated as 0. c taken at 40 equally spaced points between 0.1 and 1.5, d taken at 40 equally spaced points between 0.1 and 0.7.

Note that in this case, the initial condition does not satisfy the bound on the derivative in equation (2). Nevertheless, the small amplitude of the perturbation is enough in this case to get the behaviour predicted for the linear case.

3.2 Stability in the general viscous Burgers - Andrea

As previously mentioned, Von-Neumann stability analysis cannot be done for non linear numerical schemes. To analyse the stability conditions in the general viscous Burgers equation and visualise their dependence on the initial state, we propose two experiments. These experiments were done on the following four initial states:

Figure 2: Different initial conditions $u^{(0)}$ for the experiment



(a) Nearly constant function

(b) Gaussian function, height 2

(c) Gaussian function, height 4

(d) Step function, height 2

Parameters: $T = 20,\ L = 10,\ N_x = 100$.
(a) $u(0, x) = 0.2$ except for $x \in 1/3, 1$, where $u(0, x) = 0.35 - 0.15x$. (b) $u(0, x) = 2 * \exp(-(x - 5)^2/2)$.
(c) $u(0, x) = 4 * \exp(-(x - 5)^2/2)$. (d) $u(0, x) = 0$ except for $x \in \{L/3, 2L/3\}$, where $u(0, x) = 2$.

3.2.1 Experiment 1 : Intuition on non-linear boundaries for stability

The first experiment aims to compute numerically the stability conditions of scheme (7), in terms of the viscosity coefficient and the temporal and spatial resolutions, for the chosen initial states.

Description:

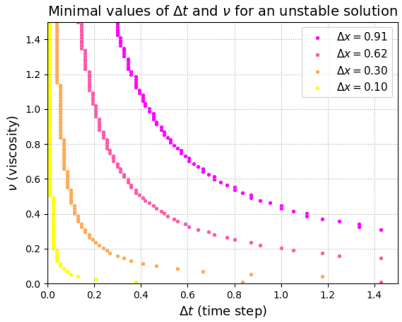
- Fix L and T, the grid length and simulation time for the experiment
- Choose a set of initial state functions with different shapes
- Choose a list of resolution values $\widehat{\Delta x}$ to test
- Compute the corresponding approximated list of N_x values, as the closest integers to $L/\widehat{\Delta x}$, and then recompute the true list of resolution values Δx that will be used in the experiment
- For each true Δx, compute the minimal pairs $(\nu, \Delta t)$ from which the solution is unstable
- Plot the obtained pairs, creating a stability contour for each Δx

Notes:

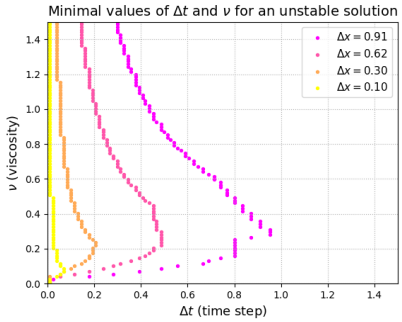
- As initial states, we chose the ones presented in Figure 2
- Our criterion for instability is in terms of relative variation, with respect to the initial condition variation (formula 18). When this value gets larger than a certain threshold, chosen to be 1 in this experiment, (or becomes nan), then we consider the scheme unstable

Results:

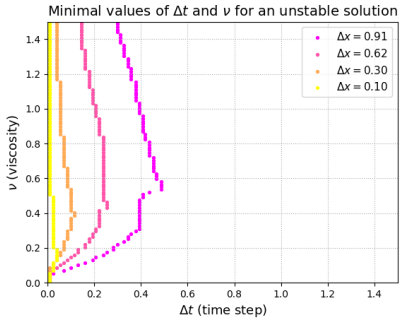
Figure 3: Stability boundaries for different initial conditions $u^{(0)}$



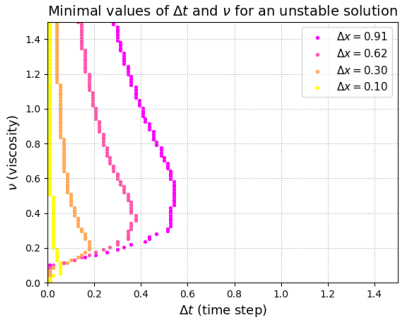
(a) Nearly constant function (Fig.2a)



(b) Gaussian function, height 2 (Fig.2b)



(c) Gaussian function, height 4 (Fig.2c)



(d) Step function, height 2 (Fig.2d)

Parameters: $T = 20,\ L = 10,\ \widehat{\Delta x} \in \{0.1, 0.3, 0.6, 0.9\}$, 100 values of $\Delta t \in [0.01, 1.5]$, 100 values of $\nu \in [0.01, 1.5]$

The results in Figure 3 qualitatively indicate some relation between parameters $\Delta x, \Delta t$ and ν that enable stability of the solution. When the spatial resolution Δx decreases, Δt must also decrease. Also, for large variance initial conditions, or if they have discontinuities, it seems to be necessary to have large viscosity values to ensure stability. These two observations are qualitatively aligned with the boundary conditions for the advection-diffusion equation (15), and a motivation for the next experiment.

3.2.2 Experiment 2: the role of linear and non-linear terms in the equation

The goal of this experiment is to verify if the stability analysis done for the linear advection-diffusion equation can provide information about the conditions of stability in the non-linear viscous Burgers. It follows from the previous experiment.

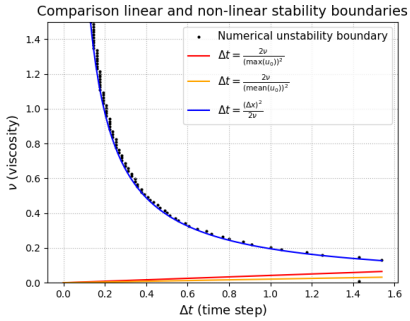
Description:

- Fix L and T, the grid length and simulation time for the experiment
- Choose a set of initial state functions with different shapes
- Choose a Δx value for which, according to experiment 1, there are some pairs $(\Delta t, \nu)$ for which the scheme is stable.
- Plot the stability boundary in a plot $\Delta t, \nu$, as done in the previous experiment
- Add boundary $\Delta t = \frac{\Delta x^2}{2\nu}$ for stability in the linear advection-equation
- Add boundaries $\Delta t = \frac{2\nu}{mean_j \left(u_j^{(0)}\right)^2}$ and $\Delta t = \frac{2\nu}{max_j \left(u_j^{(0)}\right)^2}$

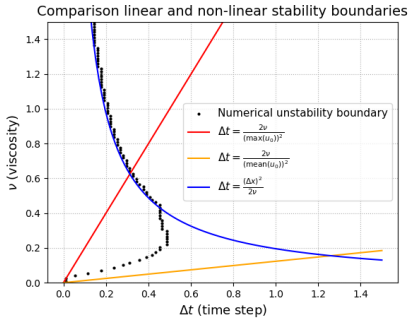
The two last boundaries in the plot represent the stability conditions (15) of the linear-advection diffusion, taking the mean and the maximal values of $u^{(0)}$ as representatives of the shape of the initial condition, that influences the advection term.

Results:

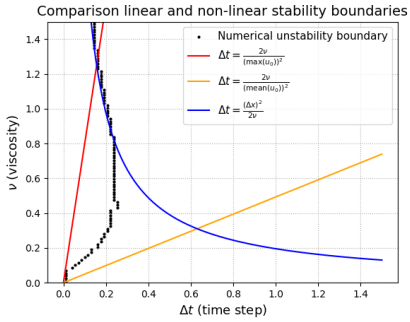
Figure 4: Comparison linear and non linear stability boundaries for different initial conditions $u^{(0)}$



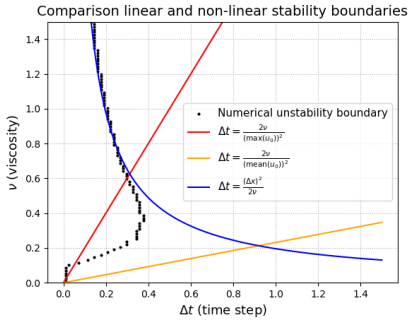
(a) Nearly constant function (Fig.2a)



(b) Gaussian function, height 2 (Fig.2b)



(c) Gaussian function, height 4 (Fig.2c)



(d) Step function, height 2 (Fig.2d)

Parameters: $T = 20,\ L = 10,\ \widehat{\Delta x} = 0.6,\ 100$ values of $\Delta t \in [0.01, 1.5],\ 100$ values of $\nu \in [0.01, 1.5]$

The results in Figure (4) illustrate some properties of the non-linear Viscous Burgers.
On the one hand, it can be seen that for large kinematic viscosity values, the boundaries for the linear (blue line) and non linear (black dots) version of the equation seem to match, regardless of the initial condition. We interpret that this is due to the fact that this boundary is related to the diffusion term of the equation, that dominates for large ν. This term is linear on the velocity, and thus the non constancy of the initial state does not affect this condition much. Going further, we could say that, when the viscosity is strong, it takes care of discontinuities and peaks by smoothing the velocity profile, which ensures stability.

On the other hand, for small ν values, the numeric boundary for stability seems to fall between the two lines used as reference for stability in the linear case. One can see that the larger the variance of the initial state, or the more discontinuous it is, the less the real boundary agrees with the line $\Delta t = \frac{2\nu}{mean_j\left(u_j^{(0)}\right)^2}$. This seems coherent with the fact that the advection term is directly linked to the non linearity of the equation so, when it dominates (for small viscosity values), the boundary is strongly affected by the properties of the initial state.

Experiments 1 and 2 done on the FTCS (eq. (7)) scheme of the non linear viscous Burgers equation illustrate the dependencies among the parameters that enable its stability. For numerical implementation purposes, it seems relevant to present the following hypothesis on a non-tight stability condition:

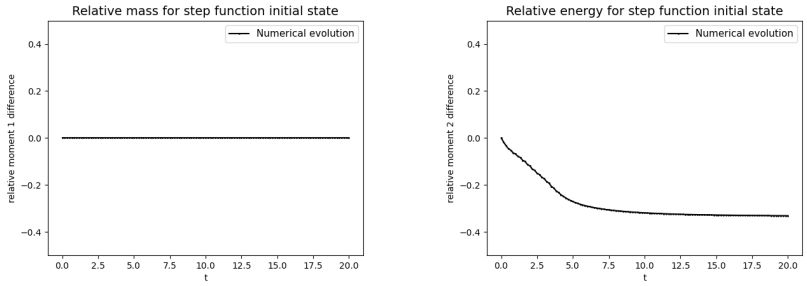
$$\Delta t \leq \min \left\{ \frac{2\nu}{max_j \left(u_j^{(0)} \right)}, \frac{\Delta x^2}{2\nu} \right\} \tag{20}$$

Where $\{u_j^{(0)}\}_{j=0}^{N_x}$ is the initial state.

3.3 Conservation - Andrea

We expect to numerically verify that the scheme is indeed mass conservative for stable solutions, independently of the shape of the initial state, but non energy conservative. The experiment consists in running the numerical scheme and plotting the mass and energy evolution over time, relative to the initial mass. Parameters are chosen so that the scheme is stable for the chosen initial condition.

Figure 5: Relative mass and energy evolution over time, for step function in Fig. 2d



(a) Relative mass evolution (b) Relative energy evolution

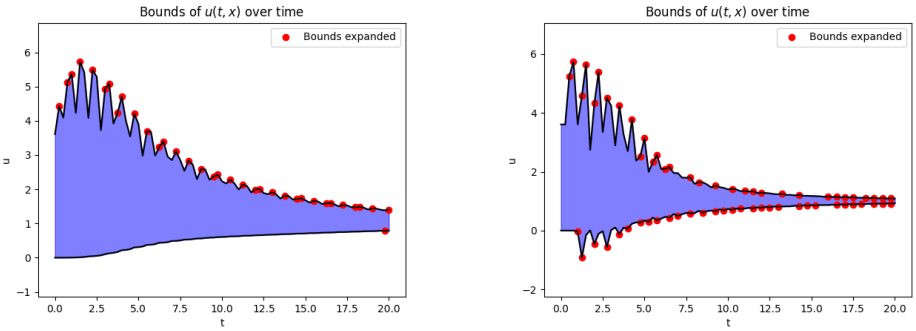
Parameters: $T = 20,\ L = 10,\ N_x = 30,\ N_t = 120,\ \nu = 0.2$, initial state from Fig. (2d): $u(0, x) = 0$ except for $x \in \{L/3, 2L/3\}$, where $u(0, x) = 2$.

Figure 5a is indeed consistent with mass conservation for stable solutions, as expected (subsection 2.3). Likewise, Figure 5b shows that energy is not conserved.

3.3.1 Boundedness Is Not Conserved - Guy

Starting with the initial condition in figure 2c, we run our FTCS scheme 7 as well as the Leapfrog scheme 10 (with FTCS as first step). We plot the maximal and minimal value of u as a function of time, and mark places where the maximum grows relative to the previous step, or the minimum gets smaller.

Figure 6: Evolution of the maximum and minimum of u over time



(a) FTCS scheme (b) Leapfrog scheme

Parameters: $T = 20,\ L = 10,\ \nu = 0.3,\ N_x = 11,\ N_t = 80$. Initial condition: $u(0, x) = 4\exp(-(x - 5)^2/2)$.

We see that neither method conserves bounds on u.

Note that we do not know what theoretical behaviour to expect from the real solution in this regard.

3.4 Accuracy - Guy

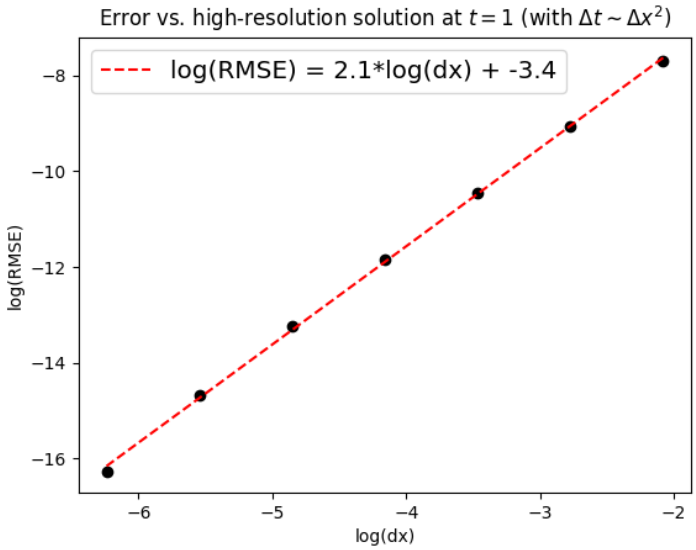
Using the sequence of spatial resolutions $N_x = 8, 16, 32, \ldots, 512$, we start from a gaussian initial condition and compute the root mean square error at final time:

$$\varepsilon_{RMS} = \sqrt{\Sigma_j (u(T, j\Delta x) - u_{ref}(T, j\Delta x))^2} \tag{21}$$

where the error is relative to a reference solution u_{ref} computed with the same scheme using a higher resolution $N_x = 1024$. We take $N_t = \left\lceil \frac{\nu L}{T^2} \right\rceil N_x^2$ for both the test and the reference solutions.

We then plot the error as a function of Δt on a log-log scale, and fit a line to it where we expect the slope to be the order of accuracy.

Figure 7: RMSE error relative to reference solution



Parameters: $T = 1,\ L = 1,\ \nu = 0.1$, initial condition $u(0, x) = \frac{1}{\sqrt{2\pi}} \exp(-(x - 0.5)^2/2)$.

As seen in the plot, we indeed get an order of accuracy approximately equal to 2. However, since the reference solution is computed by the same scheme, this only shows the rate of convergence of the scheme, and not necessarily that it converges to the correct solution.

4 Conclusion

This report presents an implementation of the FTCS numerical scheme for solving the viscous Burgers equation in one dimension, and the experiments done to test its properties. We have focused our work on analysing stability, accuracy, and conservation properties of the scheme.

4.1 Further work - Andrea

Following the experiments presented in Section 3.2, it could be interesting to create a function that, given a scheme, an initial state $u(0, x)$, and a spatial resolution Δx, outputs intervals $[\Delta t_{min}, \Delta t_{max}]$ and $[\nu_{min}, \nu_{max}]$ where the scheme is stable. This would allow testing new properties of the scheme for different initial conditions without the risk of instability.

Regarding the stability analysis, the criterion for instability could also be improved. In the current implementation, the condition is that the total variation does not exceed 1, which is an arbitrary threshold based on observations. However, oscillations do not necessarily imply instability; a more robust condition could be considered.

Finally, if time allowed, I would personally like to work more on learning efficient programming methods, code modularization, and Git.

4.2 Further work - Guy

A natural next step would be the implementation of a scheme that has second-order accuracy in both time and space. A suggested candidate is the Crank-Nicolson scheme, which is implicit and would require careful consideration of how to perform each time step efficiently. One can then check if the higher accuracy in time can be fully utilised without running into instability issues.

The code used for this project can be improved upon, specifically by making it more generic (e.g. allowing any class of parameters to be correctly used in every experiment) or more flexible (e.g. passing a halting condition to schemes that compute time steps sequentially, to save resources in case of detected instability).

In terms of further experiments, one can look at the behaviour of the higher moments of the function over time, and compare them to the theoretical behaviour.

References

[Bas02] H. A. Basha. Burgers' equation: A general nonlinear solution of infiltration and redistribution. *Water Resources Research*, 38(11):29–1–29–9, 2002.

[FG] Rudolf Friedrich and Sergei Gurevich. Numerische methoden für komplexe systeme i/ii – chapter: Burgers equation. Lecture notes, Wintersemester 2009/10.

[RLDL20] Soorok Ryu, Geunsu Lyu, Younghae Do, and GyuWon Lee. Improved rainfall nowcasting using burgers' equation. *Journal of Hydrology*, 581:124140, 2020.

[TD10] Chuong V. Tran and David G. Dritschel. Energy dissipation and resolution of steep gradients in one-dimensional burgers flows. *Physics of Fluids*, 22(3):037102, 03 2010.

[VD18] Trung Vo Duy. One dimensional burgers equation. September 2018.

[Wel25] Hilary Weller. Numerical modelling of the atmosphere and ocean. Lecture notes, October 2025. Dr Hilary Weller.