M2 MSIAM - Wavelet & differential calculus

Maxime Renard

School year 2023-2024

Study of a scientific article

Wavelet methods to eliminate resonances in the Galerkin-truncated Burgers and Euler equations.

Authors: R. M. Pereira, R. Nguyen van yen, M. Farge, and K. Schneider

Received December 10, 2012 || Published March 28, 2013

Key words: computational physics, wavelet analysis, filtering method, dual tree, CVS filtering, Burgers equation, Euler equation, Galerkin methods.

Contents

1 Introduction

1.1 Aim of the project

In the framework of this second year of Master, a scientific knowledge and methodology shall be developed and acquired. For this reason, the Wavelet course asks for the mathematical analysis of a recent article, including the reproduction of some of the numerical methods and of the results that are presented. In addition, a critic of the considered article shall be done.

1.2 Summary of the article

The main source for this report is due to R. M. Pereira, R. Nguyen van yen, M. Farge, and K. Schneider [1]. It presents a method to filter on the fly approximations to 1D Burgers and 2D Euler equations in their Galerkin-truncated formulation, using translation invariant complex valued wavelets called **dual trees** (or **Kingslets** after the name of N. Kingsbury, their inventor [2]) as well as real valued wavelets. Such a method is performed using a threshold which is updated at each time step, to discard so-called incoherent wavelet coefficient (CVS[1] method presented initially here [3]). Such filtering is in fact needed as the Galerkin-truncated Burgers and Euler equations, when solved numerically, makes some resonance phenomena appear on smooth parts of the approximations as shown in [4], which are purely due to numerical methods and shall be suppressed.

On the one hand, some complex valued wavelet are used, in order to benefit from their shift-invariant property. It enables to introduce a dissipation mechanism which the Galerkin-truncation method deletes but which is needed to kill resonance phenomenons. On the other hand, a similar method is suggested using real valued wavelets to avoid the high redundancy of dual trees, with the introduction of the so-called *safety zone*. This zone accounts for the microscopic oscillations as well as for the wave displacement and enables to get satisfying results.

To evaluate the performance of their method, authors chose to express relative error with respect to the known solution to 1D Burgers equation, whereas for 2D Euler equation, a quantitative observation on the Laplacian of the vorticity is made to witness filtering effects.

1.3 Galerkin truncation

As the reader is likely to have understood, resonance phenomena occurring when solving Burgers or Euler equations in their Galerkin-truncated version are purely due to numerical methods and shall be removed. This leads to asking what is this method and why it is used while posing that much problems, on equations that can be solved efficiently and with a good precision using other methods.

The method

Galerkin-truncation method consists in solving an altered version of a PDE, inside which the unknown function has been processed by a low pass filter discarding harmonics beyond a so-called **wave-number**, usually denoted K_G or K. This process is made through the so-called Galerkin projector[2] P_{K_G} defined as

$$P_{K_G} : u(x) = \sum_{k \in \mathbb{Z}} \hat{u_k} e^{2i\pi kx} \quad \longmapsto \quad P_{K_G} u(x) = \sum_{|k| \leq K_G} \hat{u_k} e^{2i\pi kx} \tag{1}$$

where $(\hat{u_k})_{k \in \mathbb{Z}}$ are the exact Fourier coefficients of u, which is thus assumed to be 1-periodic. Note that the Fourier transform is only applied in space, not in time. This means that the approximated solution which is searched is not u but its truncated spectrum version.

[1] Coherent Vorticity Simulation

[2] Proving P_{K_G} is a projector can be done using linearity and properties on products of the inverse Fourier transform, when developing $P_{K_G}\left(P_{K_G} u(x)\right)$ to get $P_{K_G}\left(P_{K_G} u(x)\right) = P_{K_G} u(x)$, point-wise with respect to x.

Interests

Despite the non negligible drawback of introducing non-physical resonance phenomena in the approximation, the Galerkin-truncation method is a well-used technique in thermodynamics. In fact, as it is explained in [4], the Galerkin projector enables to create a link between a conservative microscopic behaviour and the dissipative macroscopic level where energy can be lost in molecular motion. It acts as an energy sink for the low wave-number harmonics that end up vanishing in high modes depending on the chosen K_G, acting as a molecular viscosity.

This article also points out that spectral methods are within the best when it comes to the numerical integration of hydrodynamic equations, on the precision point of view. Spectral methods also benefit from efficient algorithm for their numerical uses.

2 Governing equations

Due to multiple issues during the realization of this project, only the inviscid Burger equation will be studied. Some diffusion models have been used to add material to this report as results were not conclusive at first, using the tools presented by the authors. Those additional models come from the part of the course done with S. Meignen.

2.1 1D inviscid Burgers equation

Before diving into the numerical aspects, it seems interesting to make a brief presentation of the 1D Burgers equation in its continuous and Galerkin-truncated formulations, and to develop a few interesting points about these.

2.1.1 Continuous formulation

Let Ω be an open spatial domain $\Omega \subset \mathbb{R}$ and a time domain $[0, +\infty[$; denote $\Omega_T = \Omega \times [0, T[$ for $T > 0$ such that $(x, t) \in \Omega_T$. Let $u : \Omega_T \to \mathbb{R}$ be a velocity function. The 1D inviscid Burgers equation is formulated as

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{1}{2}\dfrac{\partial u^2}{\partial x} = 0 \\ u(x,0) = u_0(x). \end{cases} \tag{2}$$

Note that this equation is said to be inviscid as the viscous term $\nu \frac{\partial^2 u}{\partial x^2}$ with $\nu > 0$ could be added on the right member to introduce dissipation. This equation can be used to model and study turbulence [5], traffic flows [6] or even shocks [7]. Burgers equation (2) can be solved using the method of characteristics as shown in appendix A.1, leading to the implicit formula for the solution

$$u(x,t) = u_0(x - tu(x,t)) \tag{3}$$

which can generate singularities, as shown in appendix A.2.

2.1.2 Galerkin-truncated formulation

The Galerkin-truncated version of Burgers equation is obtained by applying Galerkin projector to (2), both on the main equation and on the initial condition. Denoting v the solution to this truncated version of the Burgers equation, one can write that :

$$\begin{cases} \dfrac{\partial v}{\partial t} + P_{K_G}\left(\dfrac{1}{2}\dfrac{\partial v^2}{\partial x}\right) = 0 \\ v(x,0) = P_{K_G}\left(u_0(x)\right). \end{cases} \tag{4}$$

As shown in [8], the main equation of this truncated version can be written under a system of non linear ordinary differential equation using conventional rules on differentiation and Fourier series :

$$\begin{cases} \dfrac{dv}{dt} = \dfrac{-ik}{2} \cdot \displaystyle\sum_{\substack{k+p+q=0 \\ |p|,|q| \leq K_G}} v_{-p}v_{-q} \\ v(x,0) = P_{K_G}\left(u_0(x)\right). \end{cases} \tag{5}$$

where k corresponds to the modes appearing when performing differentiation on the Fourier transform.

2.1.3 Fourier collocation method

Burger equation can be solved using a spectral method known as Fourier collocation method which makes use of a differential operator $D_N : \ell^2(\mathbb{R}) \to \ell^2(\mathbb{R})$ that can be defined as $D_N((u_k)_k) = \mathcal{F}^{-1}\left[i.(m_k)_k \mathcal{F}[(u_k)_k]\right]$ where $\mathcal{F}$ denotes the Fourier transform and $(m_k)_k$ the sequence of Fourier modes corresponding to the Fourier transform of the sequence $(u_k)_k$. According to [9], with such operator, Burger equation can be written as :

$$\frac{\partial u}{\partial t} + \frac{1}{3}D_N(u^2) + \frac{1}{3}uD_N(u) = 0. \tag{6}$$

2.2 Diffusion models

The issue authors try to tackle is a denoising problem on a signal. Some well known techniques are used in image processing, such as diffusion models, based on the heat equation. Here some of them are presented, that will be used later on.

2.2.1 Heat equation

One knows that the heat equation constitutes a good toy model for denoising signals as it has a tendency to smooth sharp gradient locations. The model for a signal u is expressed as

$$\frac{\partial u}{\partial t} = c\Delta u \tag{7}$$

where $c > 0$ and Δ the spatial differential operator (corresponding to $\frac{\partial^2}{\partial x^2}$ in 1D case). Some initial conditions have to be added, which are given by the shape of the signal profile to filter at each time step. Boundary conditions will be periodic as we look for such property on the Burger equation solution to filter. It lasts to decide the time period to use ; this will be decided experimentally[3]. To apply this model, an explicit finite difference scheme will be used as :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{c}{\Delta x^2}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right) \tag{8}$$

on the same grid that is used to discretize burgers equation. The only difference is that the time step will be changed to match the CFL condition $\Delta t \leq \frac{\Delta x^2}{2}$ such that $\Delta t = \frac{1}{2}\Delta x^2$.

2.2.2 Perona-Malik model

The Perona-Malik model tries to avoid smoothing regions where sharp gradient occur, which could be interesting for Burger equation with shocks. It is defined as

$$\frac{\partial u}{\partial t} = c\ div\left[g(\|\nabla u\|)\nabla u\right] \tag{9}$$

where $g : s \in \mathbb{R} \mapsto \dfrac{1}{1+\frac{s^2}{\lambda^2}}$. For large s, $g(s)$ is small meaning the smoothing is reduced, and inversely, for small s, $g(s)$ is close to 1 so that the smoothing is stronger. Same initial conditions and boundary conditions are used as for the heat equation model. The numerical scheme has to be changed to match this new term. One can witness that for $g \equiv 1$, the heat equation model is retrieved. The scheme that is used is the one proposed by Perona and Malik in 1990 [10], well explained by M. Wielgus [11], which is

$$\begin{cases} g_{j+\frac{1}{2}}^n &= g\left(\frac{u_{j+1}^n - u_j^n}{\Delta x}\right) \\ \frac{u_j^{n+1} - u_j^n}{\Delta t} &= \frac{c}{2\Delta x^2}\left(g_{j+\frac{1}{2}}^n(u_{j+1}^n - u_j^n) - g_{j-\frac{1}{2}}^n(u_j^n - u_{j-1}^n)\right) \end{cases} \tag{10}$$

for 1D case, with the addition of some celerity $c > 0$. This model can produce anti-diffusion if not well tuned.

2.2.3 Cote-Lions and Morel diffusion model

This model is really similar to the Perona-Malik one, with the difference that it convolves the gradient of the signal with a Gaussian kernel G_σ in the function g. The model is thus expressed as :

$$\frac{\partial u}{\partial t} = c\ div\left[g(\|\nabla u \star G_\sigma\|)\nabla u\right] \tag{11}$$

The scheme that is used is the same as in the case of the Perona-Malik one, with the addition of the convolution with a custom Gaussian kernel. Same initial and boundary conditions as before will be used. This model can also produce anti-diffusion if not well tuned.

[3] After some experiment, it has been decided to perform a single time step of this model to filter noise.

3 Numerical results

3.1 The code

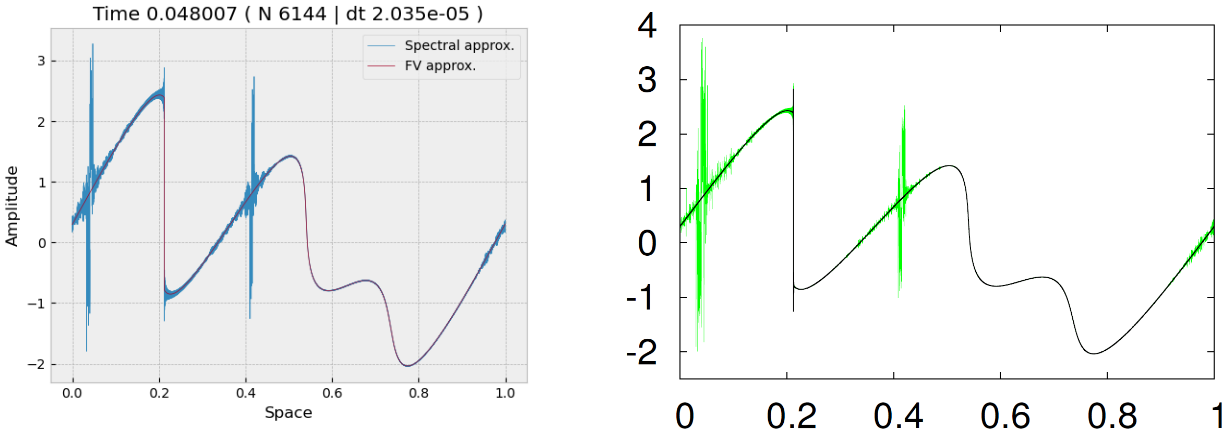
The code has been written all by myself from scratch. The file `main.py` contains the main time loop for Burgers equation, using the simulation parameters stored in `params.py`. As two possibles ways of performing time integration were found during the writing of the code, a special file `time_integration.py` hosts those methods (both Runge-Kutta of order 4, one using only explicit truncated Fourier transform, the other using Fourier collocation method). The main time loop also contains the filtering step that makes use of the methods located in `filtering_methods.py`.

Some post processing can be performed using the script `post_processing.py`, which avoids executing the main loop each time one wants to plot data. Two additional scripts can be found : `errors.py` to manage basic errors, and `burger_1D_FV_solver.py` which is the only code I did not write. It is a finite volume solver for Riemann problems on Burgers equation[4] that makes use of upwind interpolation and which is of second order. This script will be used as the reference approximation to burgers equation[5].

To reproduce results, the code can be parameterized at will with the `params.py` file. The command `python3 ./main.py` can be entered in the console to execute the program.

3.2 Using no filtering

Doing a project about spectral without knowing anything on them at first, it seems important to check that the Burger spectral solver works as expected, with no filtering. For this purpose, same parameters as in the article are used for space and time discretization. It seems important to note that the writing of this part of the code took a really large amount of time and effort. The following left plot was performed using only Galerkin-truncation with Runge-Kutta 4, it does not contain Fourier collocation.



(a) Result of the code for spectral approximation of the Burgers equation, $K = 2048$, using no filtering, at time $t \approx 0.048$.

(b) From [1] : solutions of the truncated inviscid Burgers equation at time $t = 0.48$ with no filtering (green) and CVS filtering (black) for $K = 8192$.

Figure 1: Comparison of the approximations to the truncated inviscid Burgers equation with no filtering, at time 0.048

The same behaviour is obtained as the solver used in the article. Resonance appear at same locations, but with a slightly smaller amplitude as the one from the article, on places where the signal is supposed to be smooth. Nevertheless, this is a good thing to start investigating some filtering methods to discard those resonances.

[4] https://zingale.github.io/comp_astro_tutorial/advection_euler/burgers/burgers-methods.html

[5] I tried using classical non linear solvers to approximate solutions to Burger equation, but it suffered from serious numerical errors.

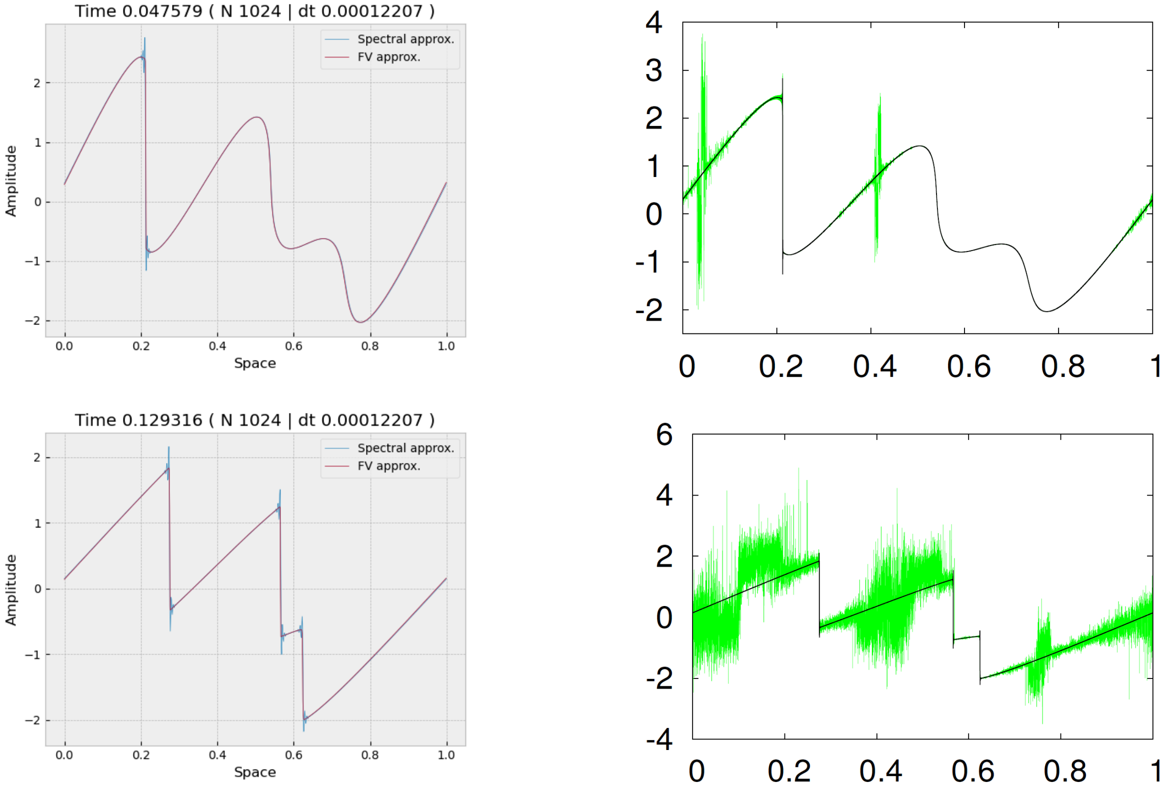
3.3 CVS filtering with Kinglsets

From now on, only the method containing Fourier collocation and Galerkin truncation will be used inside Runge-Kutta 4, to match article results. To make computations faster, we choose a space grid of $N = 1024$ points (meaning $K = 341$), and we keep other parameters as defined in the article. We use Near-Symmetric 13,19 tap filters for first level of the dual tree transform and Q-Shift 14,14 tap filters for the other levels, as it is done in the code of the authors of the article. For the filtering part, a threshold is defined as follows :

$$\begin{cases} T_{n+1} = q\sigma\left[\tilde{u}^n\right], \\ T_0 = q\sqrt{\frac{E}{N}} \end{cases} \tag{12}$$

where $q = 5$ as defined in the article (this is a choice), E is the initial total energy of the signal, N is the number of space points, $\sigma[\cdot]$ is the standard deviation of a vector of values and $\tilde{u}^n$ is the vector of the discarded coefficients at time n i.e. the ones below threshold T_n.

Figure 2: Result of my code for Burgers equation with kingslets filtering (left), result of article [1] using CVS filtering with kinglsets - black (right).

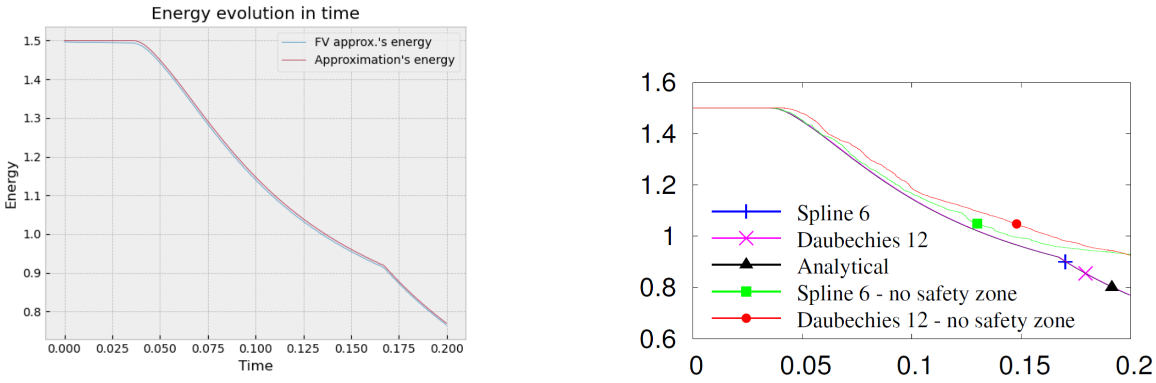


The results do not seem as perfect as those presented in the article, but the overall behaviour is reproduced. In fact, the resonance is filtered on the entire signal, and some peaks can be observed, as well as for the article results, on the locations where shocks occur.

This has been a time consuming part of the code to write, as at first, I tried writing the filtering algorithm as presented in the article, but it was not working for weeks. The authors of the article accepted to send their code, which helped me understanding that the spirit of the CVS method is not to apply a threshold, but to actually choose ranges of scales within the dual tree decomposition and to assume their role with respect to the macroscopic behaviour. For instance, signal's variance is only estimated on some part of the coarsest coefficients, and the scaling coefficients are always kept while filtering. It seems important to note that the part of the code about this CVS filtering method is greatly inspired from the `matlab` code used by the authors. It has then been adapted to the needs and my understanding, for this project.

In their article, authors present the energy profile of the filtered approximation compared to the theoretic one. Here is the comparison of the energy profile of the approximation done with my code, which is compared to the finite volume approximation, known to be of a better accuracy and without noise issues.

Figure 3: Comparison of the energy profiles issued from my code (left) and the article (right)



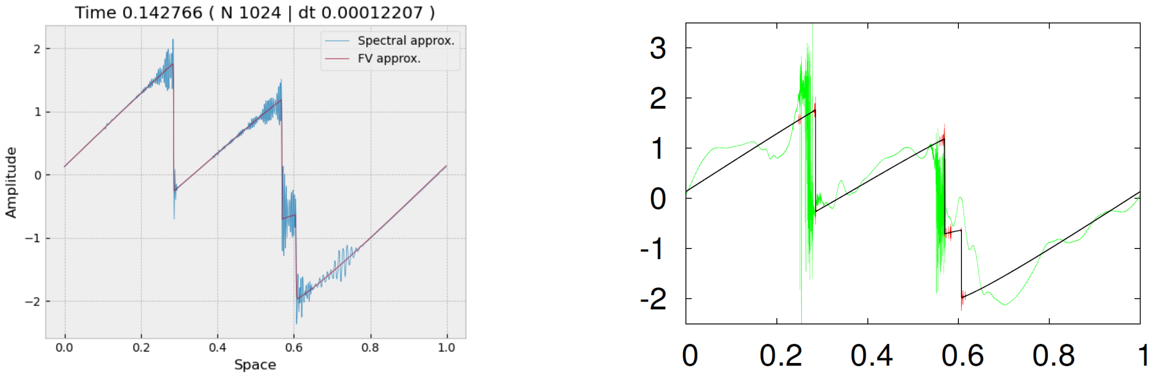
Note that for this computation, the simulation runs for a numerical time $t_f = 0.2$. We can see that a very similar profile is obtained. Starting by a plateau, followed by a slight slopes and finally an acceleration of the decrease around 0.175. It can be seen that my code seems to slightly overestimate the energy level as the blue plot on the left is below the red one. In any case, it is quite remarkable to see that the CVS filtering with dual trees manage to filter without perturbing the dynamics of the problem. Overall, this seems to confirm that the code is working well and does filter the resonance properly.

3.4 CVS filtering with real-valued orthogonal wavelets

Authors then focus on the use of real-valued orthogonal wavelet in order to avoid using highly redundant wavelets as Kinglsets. This is done at the cost of loosing the translation invariance needed for the filtering, which they counteract with the introduction of a so called *safety-zone*. This zone tells the filter that neighbouring wavelet coefficients to a coherent one should not be discarded. This helps introducing some stability in the amplitude and scales of the wavelet coefficients. Unfortunately, I have not been able to reproduce this safety zone in my code. This means that the following results have been produced using the same method as before, even if the code is different as the wavelet transform does not generate a dual tree in this case.

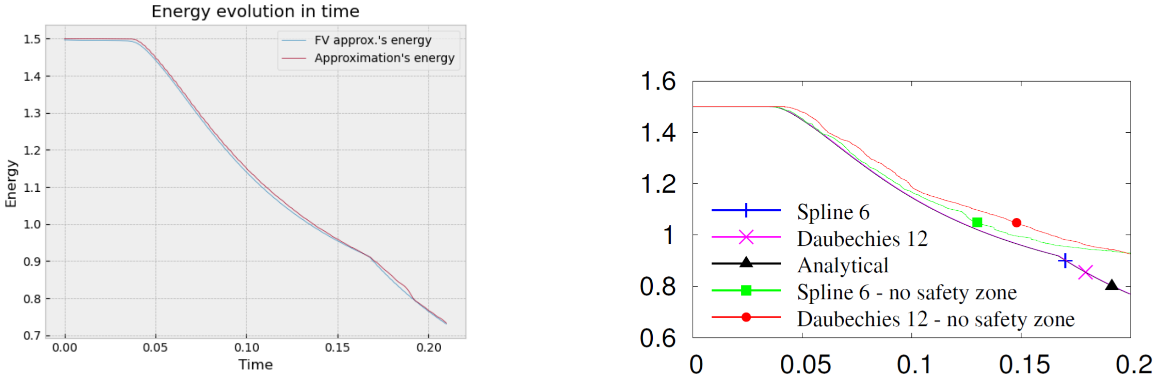
Daubechies 12 The first try performed by the authors is with a Daubechies 12 basis. Note that the time for the plots are changed to match those of the article.

Figure 4: Comparison of the approximations to the truncated inviscid Burgers equation with CVS Daubechies 12 filtering, my code (left), article results (right-green) at time 0.067



For the right plot, coming from the article, one should look at the green curve with is computed with real-valued wavelet and no safety zone. It is strange to see that my code manage to filter the signal slightly better in some sense at the author's one, but some large oscillations can still be seen on the shock locations. This result indeed shows that my implementation for the real-valued wavelet filtering does not work as expected.

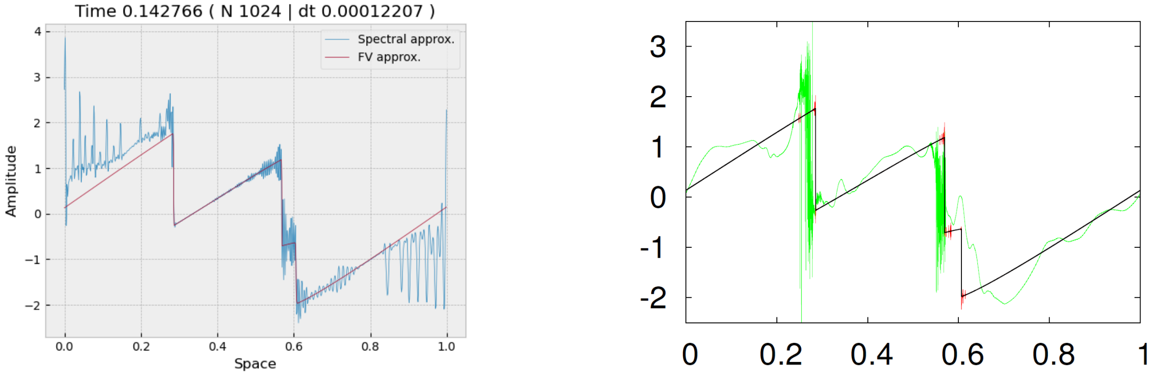
Figure 5: Comparison of the energy profiles issued from my code using Daubechies 12 (left) and the article (right)



One can see that even if the filtering is not as good as expected, the energy profile remains quite accurate compared to the finite volume approximation, meaning there is no energy drift.

Symlets 12 The authors also test their filtering method on a Spline 6 wavelet basis, which the package `pywt` does not contain and which I have not manage to add. Thus, I wanted to present a meaningful result using an other type of orthogonal real wavelet. Symlets seem to be a good choice as they are an altered version of Daubechies wavelet, to which some symmetry has been added. Thus, it could be interesting to witness the effect of the loss of the asymmetry that worked quite nicely using the Daubechies 12.

Figure 6: Comparison of the approximations to the truncated inviscid Burgers equation with CVS Symlets 12 filtering, my code (left), article results (right-green) at time 0.067



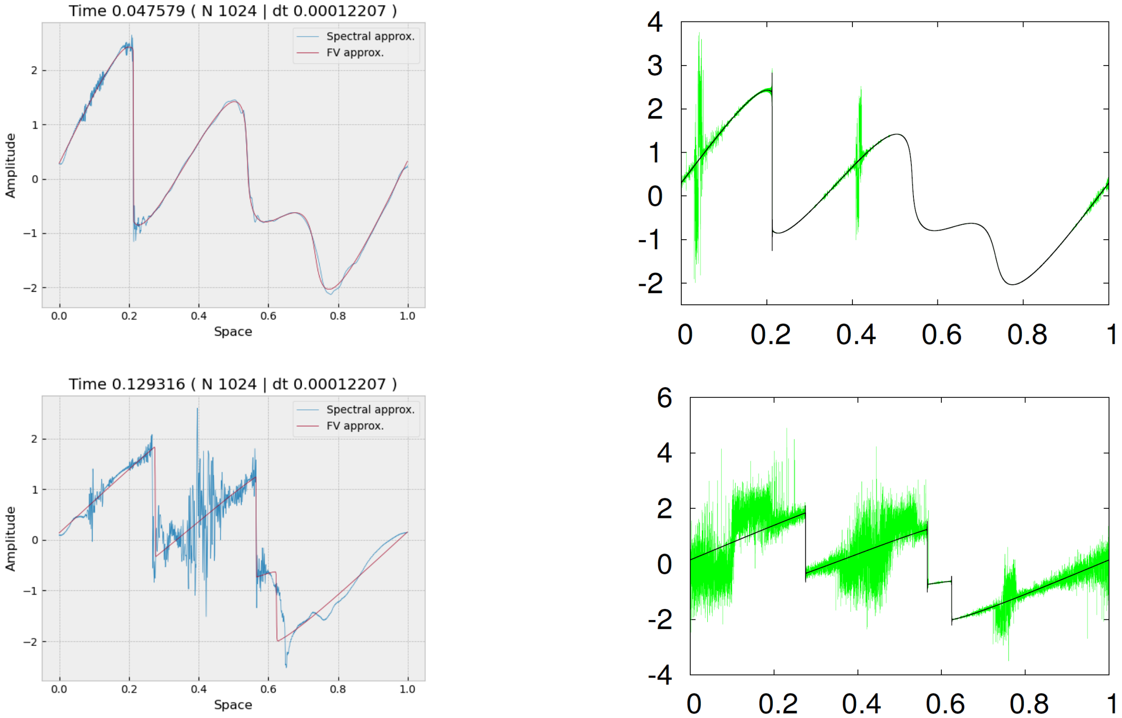
In fact, we see that it does not work very well. it seems that the asymmetry that Daubechies 12 were introducing helped in the stability of the filtering. Using Symlets 12, an energy drift can be observed right from the start at numerical time $t \approx 5.10^{-3}$ with my code.

Here we clearly see that the profile of Burgers spectral approximation is almost following a line between abscissa 0 and 0.3, but overestimating the profile. This may be due to the lack of shift invariance that is introduced with the use of real-valued wavelet without the addition of the safety zone.

3.5 Using wavelets (cf. Lab 2)

As it was frustrating to work on these Kingslets with no results for over two months, it has been decided at some point to give a try to a method seen in the Wavelet course, which uses a similar threshold method, seen in Lab 2. This threshold uses an estimator of the true noise variance[6] $\hat{\sigma}$ to have a threshold $T_n = \hat{\sigma}\sqrt{2 \log N}$ which is thus computed at each filtering step. The threshold is apply in a hard way.

Figure 7: Result of my code for Burgers equation with lab2-like filtering for $K = 341$ (left), result of the article [1] using CVS filtering (black) for $K = 8192$ (right).



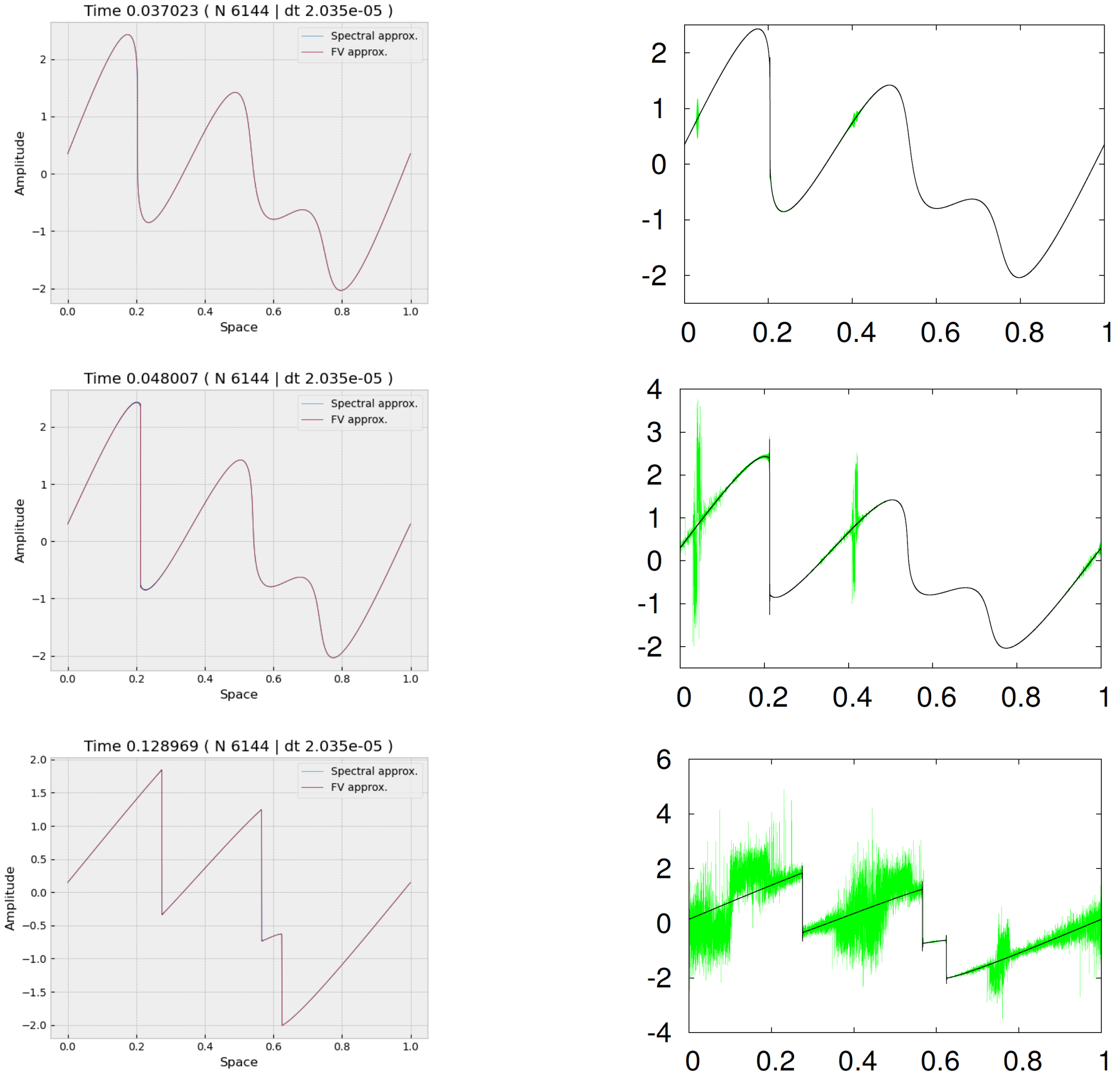
The filtering has been performed with Daubechies basis 4. Results for moderate numerical time seem fine as very few oscillating part can be seen in the signal. However, this method seems inefficient for larger simulation time, as the signal becomes totally noisy and would not be usable in a serious framework. Thus, it seems that this method does not perform well enough to be kept in mind. Some other possibilities could be to investigate other wavelet basis or other ways to define the threshold value.

[6] The estimator is given by the median of the wavelet coefficients divided by the experimentally determined coefficient 0.6745.

3.6 Using the heat equation diffusion model

As wavelet denoising was not a common method to use for me, I decided to give a try to more classical methods for denoising, using PDEs. The diffusion models presented by S. Meignen in his differential calculus course will be used. Let us first focus on the heat equation model, as it has been presented before.

Figure 8: Result of my code for Burgers equation with heat equation diffusion filtering for $K = 341$ (left), result of the article [1] using CVS filtering (black) for $K = 8192$ (right).



As the reader might see, this method is quite incredible as almost no noise can be seen on the left plots (some still appears around $x = 0.2$ for $t \approx 0.048$). No side effect can be witnessed, shocks are perfectly captured, and the time duration does not seem to be an issue with respect to the reliability of the results. It is the best method that was used so far.

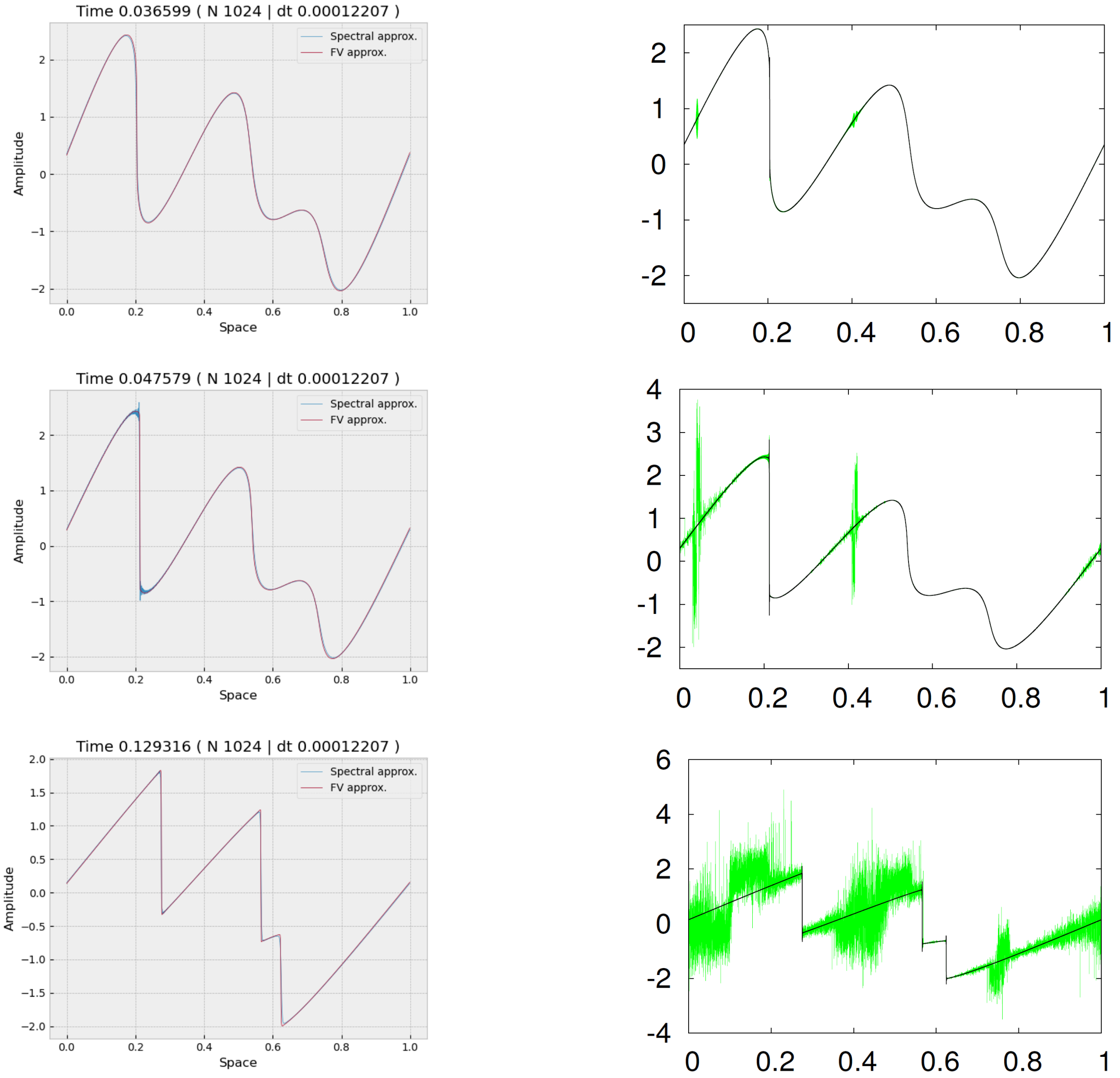
Without surprise, the energy profile matches the finite volume approximation very well, which seems to emphasize on the good performance of this method.

The heat diffusion model has been tuned to produce good results on this application. The time step for the heat diffusion is chosen such that $dt/dx^2 = 0.5$, and the propagation speed of the model is set to $c = 0.2$. Moreover, a single time step of the model is performed to smooth noise.

3.7 Perona-Malik diffusion model

Recall that this model makes use of a special function inserted in the divergence to modulate the smoothing effect when the gradient is sharp so that it conserves the global profile of the signal.

Figure 9: Result of my code for Burgers equation with Perona-Malik diffusion model for $K = 341$ (left), result of the article [1] using CVS filtering (black) for $K = 8192$ (right).



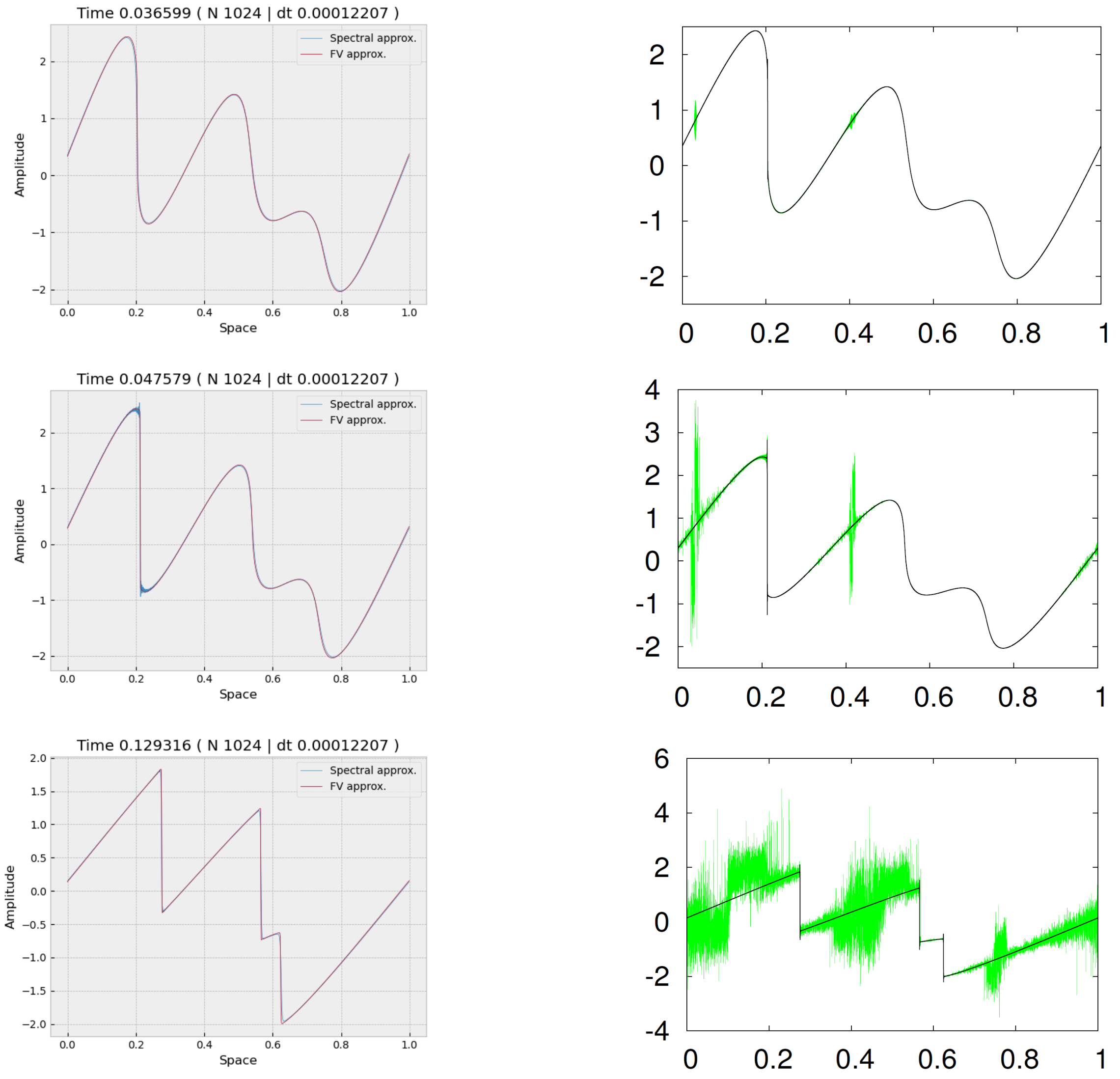
It is possible to witness that the results are quite good with this approach. Some oscillations still can be seen around the first shock at time $t \approx 0.048$ but it does not seem to introduce instability as the profile at time $t \approx 0.129$ is almost identical to the finite volume approximation.

This derived heat equation model has some good performance but is not as efficient as the pure heat equation model that has been used in the previous paragraph. It is important that the model has been tuned to produce within the best results. As for the heat equation model, this one has a time step chosen such that CFL is reached ($dt/dx^2 = 0.5$), and the propagation speed is changed to $c = 0.05$. The coefficient for modulation function g is set as $\lambda = 1000$, and 10 time steps of this diffusion model are performed to denoise the signal. These parameters have been chosen to improve the result, but might not be optimal for this application.

3.8 Cote-Lions-Morel diffusion model

This model makes use of the same scheme as the Perona-Malik one, with the only difference that it convolves the gradient of the signal with a Gaussian kernel before applying the modulation function.

Figure 10: Result of my code for Burgers equation with Cote-Lions-Morel diffusion model for $K = 341$ (left), result of the article [1] using CVS filtering (black) for $K = 8192$ (right).



We see that the results are extremely similar as the one obtained with Perona-Malik model. It seems to have a great stability over time, and to match almost perfectly the finite volume approximation.

Even if the performance of this model are also quite good, it is not as good as what we managed to obtain with the heat equation model. Again, it is important that the model has been tuned to produce within the best results. As for the heat equation model, the time step is chosen such that $dt/dx^2 = 0.5$), the propagation speed is changed to $c = 0.05$, coefficient in modulation function g is set as $\lambda = 1000$, and 10 time steps of this diffusion model are performed to denoise the signal. As before, these parameters have been chosen to improve the result, but might not be optimal for this application.

4 Conclusion

First, let us mention that this project has taken far more time than expected when choosing the article. As I was not familiar with spectral method, nor dual tree transforms, it took me a serious amount of time to get into the details and write the code.

On the critic of the article, it seems important to mention that the filtering algorithm as presented in the article never worked for me i.e. the one with an explicit threshold that is passed from a time step to the following. This can be due to my ignorance or mistakes, as I was not familiar with the methods used in the article. Hopefully, the authors accepted to share their code, which helped a lot in the realization and the understanding of this project. It is still slightly deceiving not to have been able to reproduce the results without exterior help.

On the reproduction of the results, I focused on the 1D Burgers equation as it was complex enough to get familiar with spectral methods for 1D problems. I have managed to reproduce the results in the case of the CVS filtering with dual trees, even if it seems that the accuracy of the code is not as good as the authors' one. For the real valued wavelets filtering, it is not a success as I could not make the safety zone work in my code. But the extension of the study with diffusion models coming from S. Meignen course showed that not only spectral methods can be used to filter approximations. In fact, the classical heat equation model is the method that produced the better results of this project when coming to matching the finite volume approximation.

Finally, this project has taken a tremendous amount of time for not a large amount of results, but it brought me to studying spectral methods, which I knew nothing about before, have fun with Burgers equation, which I already studied some years ago, and also to get in touch with researchers and speak about their work. This was overall, a really nice experience, and I would like again to thank M. Farge and R. M. Pereira for their collaboration.

A Properties of Burgers equation

A.1 Exact solution

Recall the equation with initial condition to the Burgers problem is given by

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{1}{2}\dfrac{\partial u^2}{\partial x} = 0 \\ u(x,0) = u_0(x). \end{cases} \tag{13}$$

We consider the method of characteristics i.e. we are looking for lines along which the solution is constant. It is important to recall that this method enables to find solutions for smooth initial condition. Let define such a line in the space-time plane by $[X(s), T(s)]$ for s a real number[7]. We define $Z(s) = u[X(s), T(s)]$ which represents the parametrized characteristic lines so that :

$$\frac{dZ}{ds} = 0. \quad \Longleftrightarrow \quad \frac{\partial u}{\partial t}\frac{dT}{ds} + \frac{\partial u}{\partial x}\frac{dX}{ds} = 0.$$

To build again the initial equation, it is clear that we need to impose :

$$\begin{cases} \dfrac{dT}{ds} = 1, \\ T(t) = t, \\ \dfrac{dX}{ds} = u(x,t), \\ X(t) = x. \end{cases} \quad \Longrightarrow \quad \begin{cases} T(s) = s, \\ X(s) = (s-t)u(x,t) + x. \end{cases} \tag{14}$$

Condition for an evaluation at t of X and T come from the framework imposed by the method of characteristics. Moreover, note that $u\frac{\partial u}{\partial x} = \frac{1}{2}\frac{\partial u}{\partial x}$, explaining the value we chose for the derivative of X.

Recall that the main assumption of the method of characteristics is that $\frac{dZ}{ds} = 0$, meaning that Z is a constant function of the real value s. Notice that,

$$\begin{cases} Z(0) = u[X(0), T(0)] = u(x - tu(x,t), 0) = u_0(x - tu(x,t)), \\ Z(t) = u[X(t), T(t)] = u(x,t). \end{cases}$$

Those two relations thus provide an implicit relation on the solution to Burgers equation :

$$u(x,t) = u_0(x - tu(x,t)). \tag{15}$$

Numerically, this expression can be exploited using non linear solvers such as the bisection algorithm, or Newton's and quasi-Newton methods for example.

A.2 Singularity formation time

Looking at the expression of $X(s)$ in (14), one can notice that characteristics could intersect at some given time t^* which can easily be computed. For instance, consider two distinct characteristic lines that go though points $(x_1, 0)$ and $(x_2, 0)$ respectively, in the space-time plane, and that intersect at time t^* :

$$X_1(0) = x_1 - t^* u_0(x_1) \quad \text{and} \quad X_2(0) = x_2 - t^* u_0(x_2).$$

Setting these two quantities to be equal provides a set of values for t^*, on which we choose to pick the smallest possible value as it will be the first time characteristics cross i.e. singularities occur.

$$t^* = \inf_{x_1, x_2 \in \Omega} \left\{ -\frac{x_2 - x_1}{u_0(x_2) - u_0(x_1)} \,\middle|\, \frac{x_2 - x_1}{u_0(x_2) - u_0(x_1)} < 0 \right\}.$$

[7] we can assume the characteristics enable to reverse the displacement induced by burgers equation as no diffusion term is present, thus no entropy is created. However, it is known that characteristics can cross, so this assumption is only valid before the time at which characteristics do not merge (entropy creation thus information loss and irreversibility). Setting $s \geq 0$ is thus physically consistent and enables to let s be as big as possible, but does not permit as much reversibility.

References

[1] R. M. Pereira et al. "Wavelet methods to eliminate resonances in the Galerkin-truncated Burgers and Euler equations". In: *Phys. Rev. E* 87 (3 Mar. 2013), p. 033017. DOI: 10.1103/PhysRevE.87.033017. URL: https://link.aps.org/doi/10.1103/PhysRevE.87.033017.

[2] Nick Kingsbury. "Complex Wavelets for Shift Invariant Analysis and Filtering of Signals". In: *Applied and Computational Harmonic Analysis* 10.3 (2001), pp. 234–253. ISSN: 1063-5203. DOI: https://doi.org/10.1006/acha.2000.0343. URL: https://www.sciencedirect.com/science/article/pii/S1063520300903439.

[3] Marie Farge and Kai Schneider. "Coherent Vortex Simulation (CVS), A Semi-Deterministic Turbulence Model Using Wavelets". In: *Flow, Turbulence and Combustion* 66 (2001), pp. 393–426. DOI: https://doi.org/10.1023/A:1013512726409. URL: http://wavelets.ens.fr/PUBLICATIONS/ARTICLES/PDF/151.pdf.

[4] Samriddhi Sankar Ray et al. "Resonance phenomenon for the Galerkin-truncated Burgers and Euler equations". In: *Phys. Rev. E* 84 (1 July 2011), p. 016301. DOI: 10.1103/PhysRevE.84.016301. URL: https://link.aps.org/doi/10.1103/PhysRevE.84.016301.

[5] Sergey G. Chefranov and Artem S. Chefranov. "Exact solution to the main turbulence problem for a compressible medium and the universal law turbulence spectrum of breaking waves". In: *Physics of Fluids* 33.7 (2021). ISSN: 1089-7666. DOI: 10.1063/5.0056291. URL: http://dx.doi.org/10.1063/5.0056291.

[6] Andronikos Paliathanasis. "Symmetry Analysis for the 2D Aw-Rascle Traffic-Flow Model of Multi-Lane Motorways in the Euler and Lagrange Variables". In: *Symmetry* 15.8 (2023). ISSN: 2073-8994. DOI: 10.3390/sym15081525. URL: https://www.mdpi.com/2073-8994/15/8/1525.

[7] Dongho Chae and Seung-Yeal Ha. "On the formation of shocks to the compressible Euler equations". In: *Communications in Mathematical Sciences* 7 (2009), pp. 627–634. URL: https://api.semanticscholar.org/CorpusID:55817112.

[8] Andrew J. Majda and Ilya Timofeyev. "Remarkable Statistical Behavior for Truncated Burgers-Hopf Dynamics". In: *Proceedings of the National Academy of Sciences of the United States of America* 97.23 (2000), pp. 12413–12417. ISSN: 00278424. URL: http://www.jstor.org/stable/123862 (visited on 11/27/2023).

[9] Romain Nguyen van yen et al. "Wavelets meet Burgulence: CVS-filtered Burgers equation". In: *Physica D: Nonlinear Phenomena* 237.14 (2008). Euler Equations: 250 Years On, pp. 2151–2157. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2008.02.011. URL: https://www.i2m.univ-amu.fr/perso/kai.schneider/PDF-FILES/nfksk_physd_final08.pdf.

[10] P. Perona and J. Malik. "Scale-space and edge detection using anisotropic diffusion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639. DOI: 10.1109/34.56205.

[11] Maciek Wielgus. *Perona-Malik equation and its numerical properties.* 2014. arXiv: 1412.6291 [cs.NA].