Numerical Simulations of the Modular Burgers Equation

Jeanne Lin, Dmitry E. Pelinovksy

July 2023

1 Modular Burgers Equation with Odd Parity

We are working with the viscous Burgers equation with modular nonlinearity

$$\begin{cases} u_t = (|u|)_x + u_{xx}, & t > 0, \\ u|_{t=0} = \phi(x). \end{cases} \tag{1}$$

We will prove that the odd parity of the initial data is preserved in the time evolution of the modular Burgers equation. As a result of this property, solutions of (1) can be restricted to the class of odd functions, i.e. $u(x,t) = -u(-x,t)$, on the interval $[0,\infty)$ subject to the Dirichlet boundary condition at $x = 0$.

The proof of the preservation of the odd parity is based on the following arguments. We write

$$u(x,t) = \begin{cases} u_+(x,t), & x > 0 \\ u_-(x,t), & x < 0 \end{cases}$$

For $x > 0$, we have from (1) that

$$\partial_x(|u_+(x,t)|) + \partial_x^2(u_+(x,t)) = \partial_t u_+(x,t).$$

Hence, for $x < 0$, we obtain

$$-\partial_x(|u_+(-x,t)|) + \partial_x^2(u_+(-x,t) = \partial_t u_+(-x,t).$$

Further, for $x < 0$, we have from (1) that

$$\partial_x(|u_-(x,t)|) + \partial_x^2(u_-(x,t)) = \partial_t u_-(x,t).$$

Adding these equations for $x < 0$ yields

$$\partial_x\left(|u_-(x,t)| - |u_+(-x,t)|\right) + \partial_x^2\left[u_-(x,t) + u_+(-x,t)\right] = \partial_t\left[u_-(x,t) + u_+(-x,t)\right].$$

If $u_-(x,0) = u_+(-x,0)$, then

$$\partial_t\left[u_-(x,t) + u_+(-x,t)\right]|_{t=0} = 0$$

so that $u_-(x,t) = -u_+(-x,t)$ holds for every $t \geq 0$. Hence, $u(x,t) = -u(-x,t)$ holds for $t > 0$ if $\phi(x) = -\phi(-x)$ for $t = 0$.

The purpose of this work is to confirm the precise scaling law of the finite-time extinction for odd initial data with multiple interfaces, which was conjectured in previous work [2] based on direct simulations of the modular Burgers equation (1) completed with the differential equation for the interface dynamics. Compared to the previously used approach, we introduce here a regularization for the modular nonlinearity and the finite-difference methods for numerical simulations without any additional equations for interface dynamics. The interface data is extracted from numerical data and used in the numerical computations to determine the power of the scaling law of the interface extinction.

2 Regularization

We observe that the modular Burgers equation in (1) can be rewritten as

$$\partial_t u = \operatorname{sgn}(u)\partial_x u + \partial_x^2 u, \tag{2}$$

where $\operatorname{sgn}(u)$ has a jump discontinuity at $u = 0$. We propose using a regularization term to smoothen out the jump. To do this, we rewrite $\operatorname{sgn}(u)$ and obtain

$$\operatorname{sgn}(u) = \frac{u}{|u|} = \frac{u}{\sqrt{u^2}}.$$

Then for some $\varepsilon > 0$, we define

$$f'_\varepsilon(u) := \frac{u}{\sqrt{\varepsilon^2 + u^2}}$$

in which $f'_\varepsilon(u) \to \operatorname{sgn}(u)$ as $\varepsilon \to 0$ for all $u \in \mathbb{R}$ pointwise. This yields a regularized Burgers' equation

$$\partial_t u_\varepsilon = \frac{u_\varepsilon}{\sqrt{\varepsilon^2 + u_\varepsilon^2}}\partial_x u_\varepsilon + \partial_x^2 u_\varepsilon. \tag{3}$$

Other possible regularizations of (2) can be constructed using the hyperbolic tangent and inverse tangent function, e.g.

$$\partial_t u_\varepsilon = \tanh\left(\frac{u_\varepsilon}{\varepsilon}\right)\partial_x u_\varepsilon + \partial_x^2 u_\varepsilon$$

and

$$\partial_t u_\varepsilon = \frac{2}{\pi}\arctan\left(\frac{u_\varepsilon}{\varepsilon}\right)\partial_x u_\varepsilon + \partial_x^2 u_\varepsilon.$$

Both converge to (2) as $\varepsilon \to 0$ for all u pointwise.

3 Initial Data

We construct initial data $u(x, 0) = \phi(x)$ for shocks and anti-shocks with the boundary condition $\phi(x) \to u_\pm$ as $x \to \pm\infty$. For the former, we have that $u_- < 0 < u_+$. It includes a monotone, steadily traveling shock, to which the evolution of small exponentially decaying perturbations converges [1]. For the latter, we have $u_- > 0 > u_+$. There exist no steadily traveling anti-shock solutions.

For the shock data, we construct an odd function $\phi(x)$ that satisfy the properties of $\phi(x) < 0$ for $0 < x < 1$, $\phi(x) > 0$ for $x > 1$, and $\phi(x) \to 1$ as $x \to +\infty$. We propose the following initial conditions to use in our numerical computations:

$$\phi(x) = \tanh(x)\left(1 - \frac{\cosh^2(\alpha)}{\cosh^2(\alpha x)}\right) \tag{4}$$

and

$$\phi(x) = \tanh(x)\left(1 - e^{\alpha(1-x^2)}\right), \tag{5}$$

where $\alpha > 0$ is a free parameter. The parameter $\alpha > 0$ can be used to construct slopes of the initial data at $x = 1$ where $\phi(1) = 0$. For the anti-shock data, we take the negative versions of (4) and (5). This gives us

$$\phi(x) = -\tanh(x)\left(1 - \frac{\cosh^2(\alpha)}{\cosh^2(\alpha x)}\right) \tag{6}$$

$$\phi(x) = -\tanh(x)\left(1 - e^{\alpha(1-x^2)}\right). \tag{7}$$

4 Finite-Difference Method

We rewrite the regularized Burgers' equation (3) in the equivalent form,

$$\partial_t u = \partial_x f_\varepsilon(u) + \partial_x^2 u, \quad f_\varepsilon(u) = \sqrt{\varepsilon^2 + u_\varepsilon^2} - \varepsilon, \tag{8}$$

where we have removed the subscript ε for the solution variable u. We will use the Crank-Nicolson method based on the trapezoidal rule of numerical integration to set up our numerical simulations for the regularized Burger's equation (8).

For numerical discretization, we first define the spatial domain $[0, L]$ partitioned into $(N+1)$ grid points with spatial step h and the time domain $[0, T]$ partitioned into M grid points with time step τ. Further, we let x_n for $0 \le n \le N$ be the spatial variable and t_m for $0 \le m \le M$ be the time variable. We set Dirichlet condition at $x = 0$ which yields $u_0^m = 0$ and Neumann condition at $x = L$. By using the virtual grid point $x_{N+1} > L$, the Neumann condition yields $u_{N+1}^m = u_{N-1}^m$.

The Crank-Nicolson method is based on the discretization rule,

$$u^{m+1} = u^m + \frac{\tau}{2h^2}\left[u_{n+1}^m - 2u_n^m + u_{n-1}^m + u_{n+1}^{m+1} - 2u_n^{m+1} + u_{n-1}^{m+1}\right] + \frac{\tau}{4h}\left[f_\varepsilon(u_{n+1}^m) - f_\varepsilon(u_{n-1}^m) + f_\varepsilon(u_{n+1}^{m+1}) - f_\varepsilon(u_{n-1}^{m+1})\right].$$

We need to solve N equations for N unknowns $\{u_n^{m+1}\}_{n=1}^N$ at each $1 \le m \le M$. Hence, we rearrange the discretization scheme to get the unknown variables on the left and the known variables on the right as

$$u_n^{m+1} + \frac{\tau}{h^2}u_n^{m+1} - \frac{\tau}{2h^2}\left(u_{n+1}^{m+1} + u_{n-1}^{m+1}\right) - \frac{\tau}{4h}\left[f_\varepsilon(u_{n+1}^{m+1}) - f_\varepsilon(u_{n-1}^{m+1})\right] = u_n^m + \frac{\tau}{2h^2}\left(u_{n+1}^m + u_{n-1}^m\right) - \frac{\tau}{h^2}u_n^m + \frac{\tau}{4h}\left[f_\varepsilon(u_{n+1}^m) - f_\varepsilon(u_{n-1}^m)\right]. \tag{9}$$

To simplify the expression, we use a predictor-corrector method (also known as Heun's method) that is constructed from Euler's method. The idea is to compute the solution at an initial point, u^m and use this to calculate an initial guess value of the next point $(u^*)^{m+1}$ (Euler's method). Heun's method then improves this initial guess value using the trapezoidal rule to determine a better estimate of the next term u^{m+1}.

To represent the predictor-corrector method, we introduce two matrices:

$$A_\pm = \begin{bmatrix} 1 \pm \frac{\tau}{h^2} & \mp\frac{\tau}{2h^2} & 0 & \cdots & 0 \\ \mp\frac{\tau}{2h^2} & 1 \pm \frac{\tau}{h^2} & \mp\frac{\tau}{2h^2} & \cdots & \vdots \\ \vdots & \mp\frac{\tau}{2h^2} & \ddots & \ddots & \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \mp\frac{\tau}{2h^2} & 1 \pm \frac{\tau}{h^2} & \mp\frac{\tau}{2h^2} \\ 0 & 0 & \cdots & \mp\frac{\tau}{h^2} & 1 \pm \frac{\tau}{h^2} \end{bmatrix}$$

where the $A_{\pm,N-1}$ elements are doubled due to the Neumann condition $u_{N+1}^m = u_{N-1}^m$. We also represent the regularized terms in matrix vector notion,

$$b^m = \begin{bmatrix} f_\varepsilon(u_2^m) \\ f_\varepsilon(u_3^m) - f_\varepsilon(u_1^m) \\ f_\varepsilon(u_4^m) - f_\varepsilon(u_2^m) \\ \vdots \\ f_\varepsilon(u_N^m) - f_\varepsilon(u_{N-2}^m) \\ 0 \end{bmatrix}$$

where $f_\varepsilon(0) = 0$ by construction of f_ε and the Dirichlet condition, and $f_\varepsilon(u_{N+1}^m) - f_\varepsilon(u_{N-1}^m) = 0$ by Neumann condition $u_{N+1}^m = u_{N-1}^m$. The prediction step is computed from (9) by Euler's method as

$$(u^*)^m = A_+^{-1}\left(A_- u^m + \frac{\tau}{2h} b(u^m)\right). \tag{10}$$

The prediction step is computed from (9) by Heun's method as

$$u^{m+1} = A_+^{-1}\left(A_- u^m + \frac{\tau}{4h} b(u^m) + \frac{\tau}{4h} b((u^*)^m)\right). \tag{11}$$

We now extract the interface position $\xi(t_m)$ from u^m at $t = t_m$ by finding the two adjacent grid points x_n and x_{n+1} where u_n and u_{n+1} are of opposite signs. By the straight line interpolation between (x_n, u_n) and (x_{n+1}, u_{n+1}), we obtain

$$u(x) = \left(\frac{u_{n+1} - u_n}{x_{n+1} - x_n}\right)(x - x_n) + u_n.$$

The value of $\xi(t_m)$ is obtained by finding the root of u as

$$\xi(t_m) = \frac{u_n x_{n+1} - u_{n+1} x_n}{u_n - u_{n+1}}. \tag{12}$$

5 Outcomes of Numerical Simulations

We performed iterations on the domain $[0, L]$ discretized with the step size $h = 0.01$. The time step was choosen to be $\tau = 0.0005$. We chose $\varepsilon = 10^{-16}$ to be as small as possible to maximize the accuracy of the regularized Burgers' equation (3) for the modular Burgers equation (2).

Figures 1 and 2 depict the outcomes of numerical simulations for the proposed initial data (4) and (5) with $\alpha = 1$ for which we take $L = 5$. It is observed that $\xi(t)$ goes to 0 in a finite time in which numerical computations can be continued, however, we stop them since we are only interested in the dynamics of multi-shock (non-monotone) solutions.

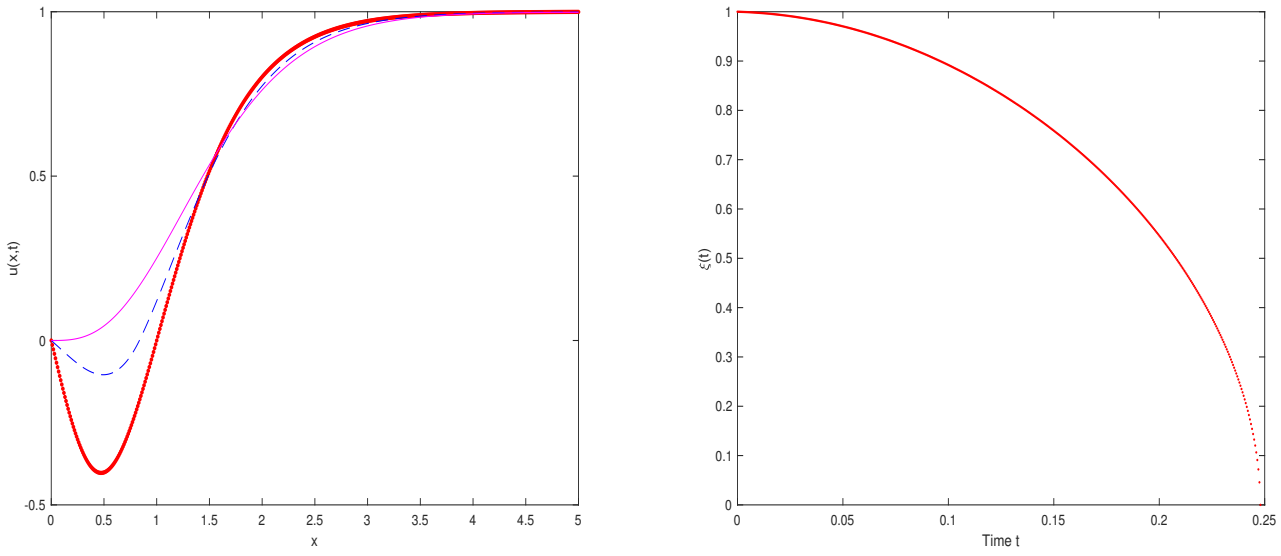

Figure 1: Evolution for the initial data in (4) with $\alpha = 1$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.1235$, and $t = 0.2475$. Right: evolution of $\xi(t)$ versus t.

We have also performed numerical simulations for initial data (5) with $\alpha = 4$ shown in Figure 3. For these simulations, we have taken $L = 10$ to avoid the boundary effects from the Neumann boundary condition at $x = L$. With smaller values of L, the solution decays below 1 at $x = L$ before the interface reaches 0. Although the initial data ϕ has large negative parts on $[0, 1]$, we observe that $\xi(t)$ goes to 0 in a finite time. Compared to 2, $\xi(t)$ is non-monotone as it expands first and then reduces towards 0.

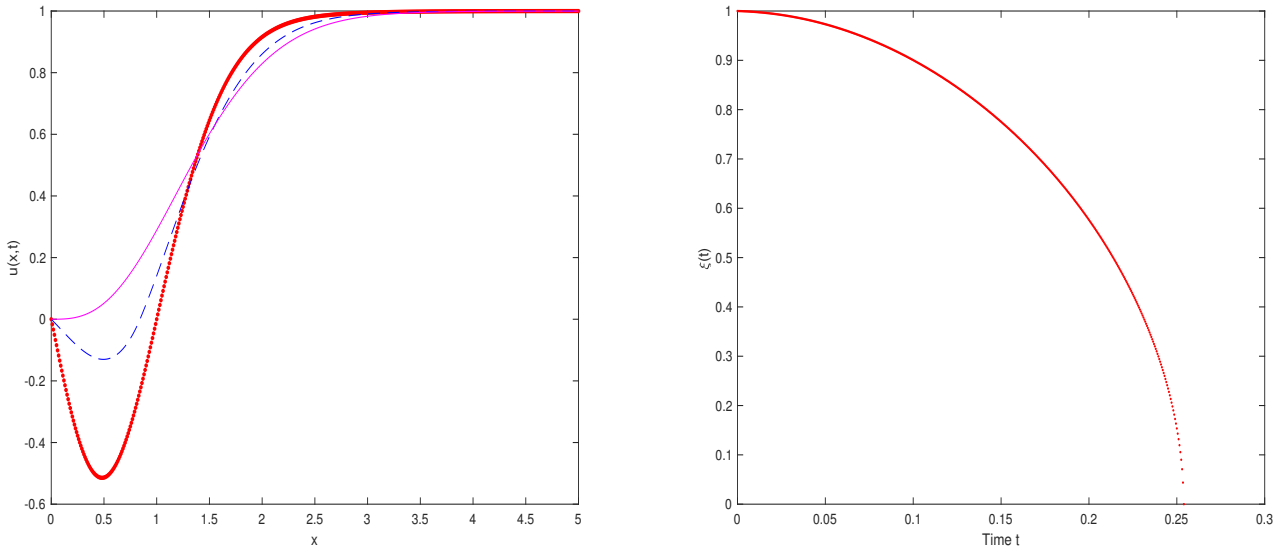

Figure 2: Evolution for the initial data in (5) with $\alpha = 1$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.126$, and $t = 0.2535$. Right: evolution of $\xi(t)$ versus t.

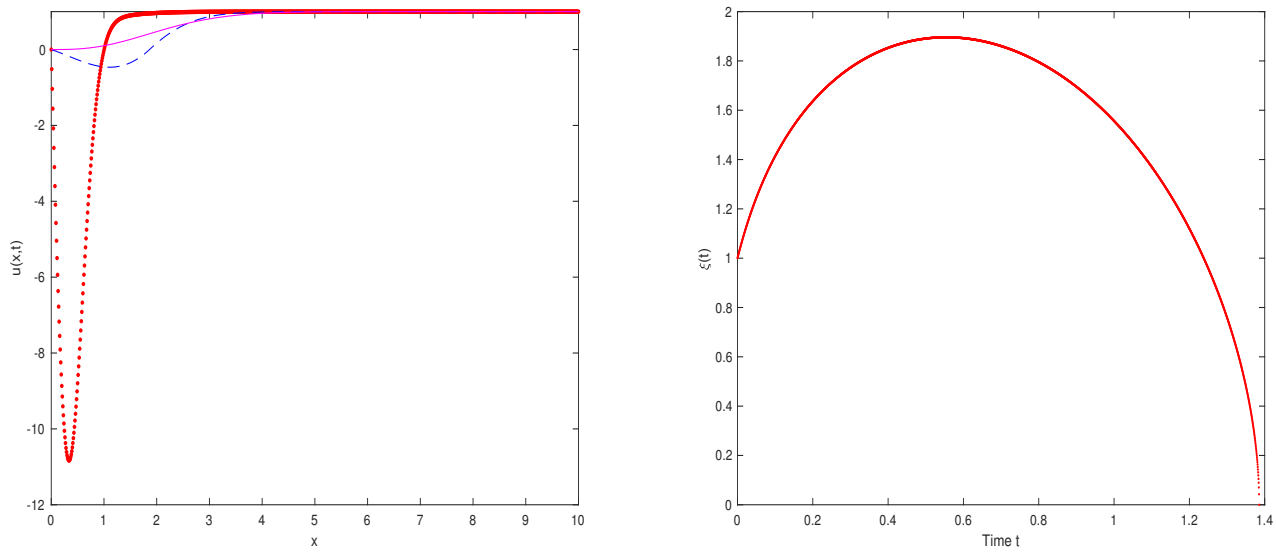

Figure 3: Evolution for the initial data in (5) with $\alpha = 4$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.692$, and $t = 1.385$. Right: evolution of $\xi(t)$ verse t

6 Scaling Laws Describing Finite-Time Extinction

In previous work [2], it was shown numerically that the scaling law of extinction

$$\xi(t) \sim \sqrt{t_0 - t} \tag{13}$$

holds as $t \to t_0$ in the case of odd viscous shocks for the modular Burgers equation (2). Here, we have confirmed this scaling law numerically for the regularized Burgers equation (8).

For the interface data from our previous numerical simulations on Figures 1-3, we use linear regression in the log-log variable to approximate the power of the scaling law. That is, we use

$$\log \xi(t) \text{ versus } c_1 \log (t_0 - t) + c_2 \tag{14}$$

where the coefficient c_1 represents the power of the scaling law. We use linear regression and obtain the best fit by minimizing the approximation error.

The outcomes of these computations are depicted in Figures 4-6 where the left panel shows the power verses t_0 and the right panel shows the corresponding approximation error verses t_0. The minimal error for initial data (4) is attained at $t_0 = 0.2479$ and this value of t_0 corresponds to $c_1 = 0.5069$. The minimal error for initial data (5) is attained at $t_0 = 0.2538$ and this value of t_0 corresponds to $c_1 = 0.5068$. The minimal error for initial data (5) at $\alpha = 4$ is attained at $t_0 = 1.3853$ and this value of t_0 corresponds to $c_1 = 0.5127$. In all cases, the power is close to the claimed value of $\frac{1}{2}$ in [2]. We note that the finite time of extinction is longer for $\alpha > 1$ than for $\alpha = 1$.

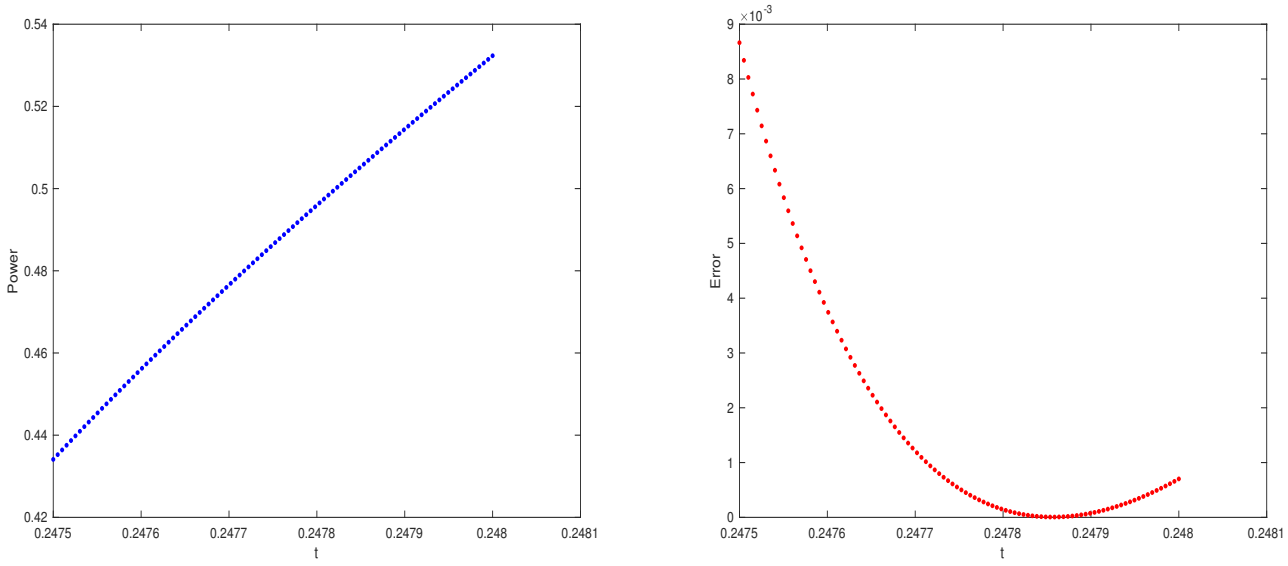

Figure 4: Left: power of the linear regression for Figure 1. Right: approximation error verses t_0.

7 Anti-Shock Numerical Simulations

We have also considered anti-shock initial data with $u_- > 0 > u_+$ and performed the same numerical simulations as before. Figures 7 and 8 depict the outcomes of the boundary-value problem for initial data (6) and (7) with $\alpha = 1$. The interface position $\xi(t)$ goes to 0 monotonically, similarly to the computations for the shock initial data shown in Figures 1 and 2.

We have also performed numerical simulations for initial data (7) with $\alpha = 4$ shown in Figure 9. Similarly to Figure 3, we observe that $\xi(t)$ first expands and then reduces towards 0.

Figures 10 and 11 show the approximate power of the scaling law and approximation error verses t_0 for the computations shown in Figures 7 and 8. The minimum error for initial data (6) is attained at $t_0 = 0.3208$ and this value of t_0 corresponds to the power $c_1 = 0.4893$. The minimum error for initial data (7) with $\alpha = 1$ is attained at $t_0 = 0.3284$ and this value of t_0 corresponds to the power $c_1 = 0.4846$. We

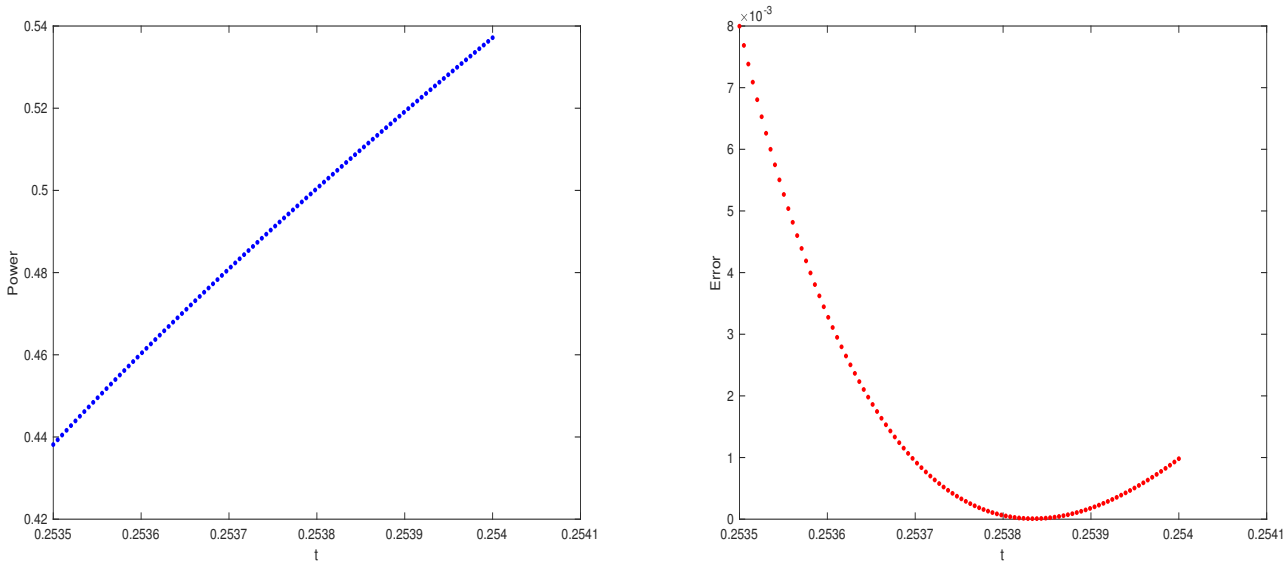

Figure 5: Left: power of the linear regression Figure 2. Right: approximation error verses t_0.

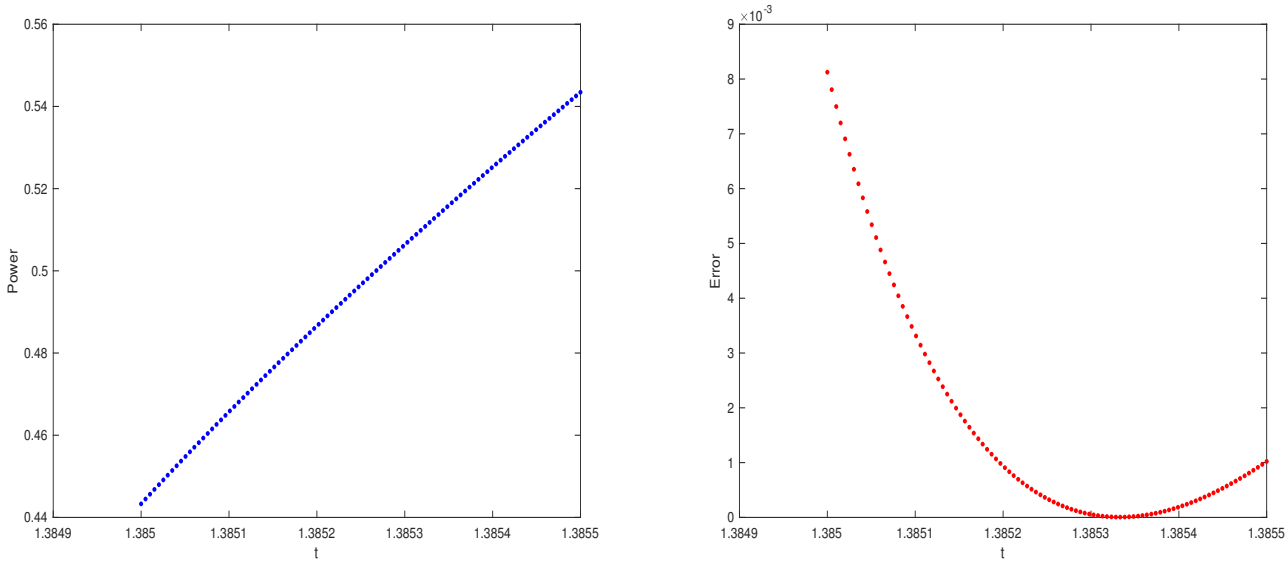

Figure 6: Left: power of the linear regression Figure 3. Right: approximation error verses t_0.

observe that these values are also close to $\frac{1}{2}$ and thus, the scaling law is shown numerically to hold for anti-shock data as well.

Figure 12 shows the approximate power of the scaling law and approximation error verses t_0 for the computations shown in Figure 9. The minimum error is attained at $t_0 = 1.4459$ and this value of t_0 corresponds to $c_1 = 0.4884$ which is also close to $\frac{1}{2}$.

The finite time of extinction is longer for the anti-shock data compared to that of the shock data both for $\alpha = 1$ and $\alpha = 4$. In both cases, the difference of t_0 is about 0.07 independently of the values of α.

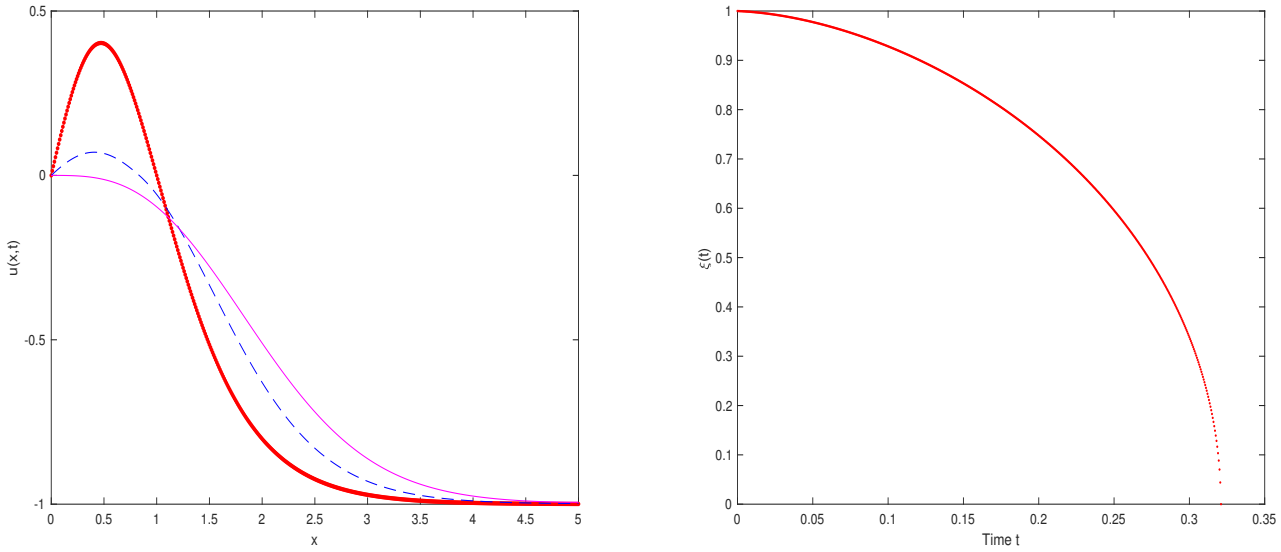

Figure 7: Evolution for the initial data in (6) with $\alpha = 1$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.16$, and $t = 0.3205$. Right: evolution of $\xi(t)$ versus t.

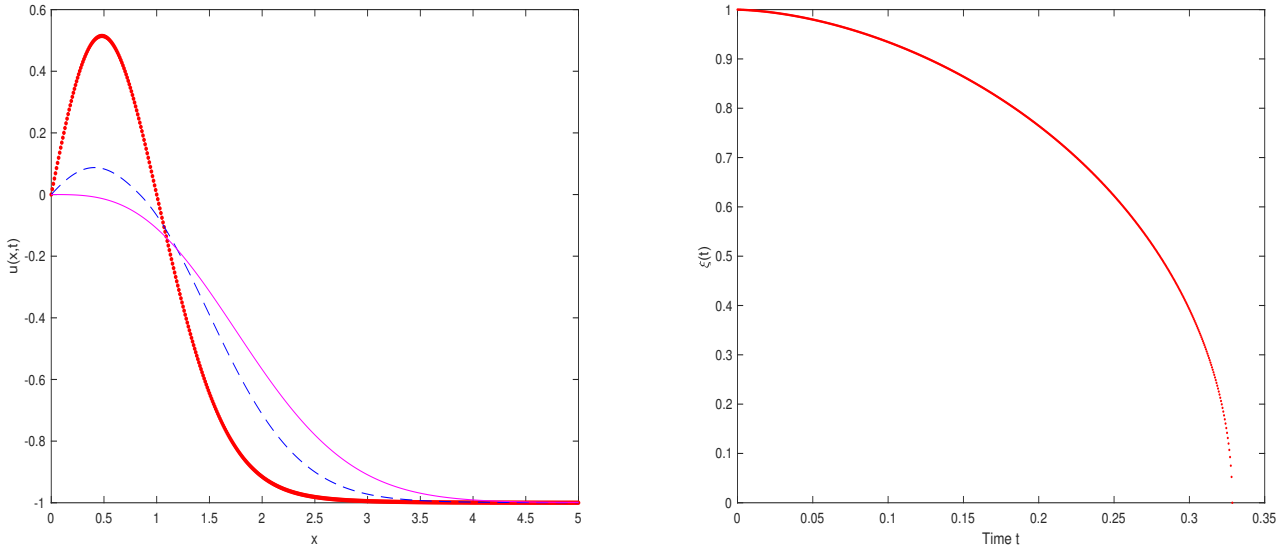

Figure 8: Evolution for the initial data in (7) with $\alpha = 1$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.1635$, and $t = 0.328$. Right: evolution of $\xi(t)$ versus t.

References

[1] U. Le, D.E. Pelinovsky, and P. Poullet, "Asymptotic stability of viscous shocks in the modular Burgers equation", Nonlinearity **34** (2021) 5979–6016

[2] D.E. Pelinovsky and B. de Rijk, "Extinction of multiple shocks in the modular Burgers equation",

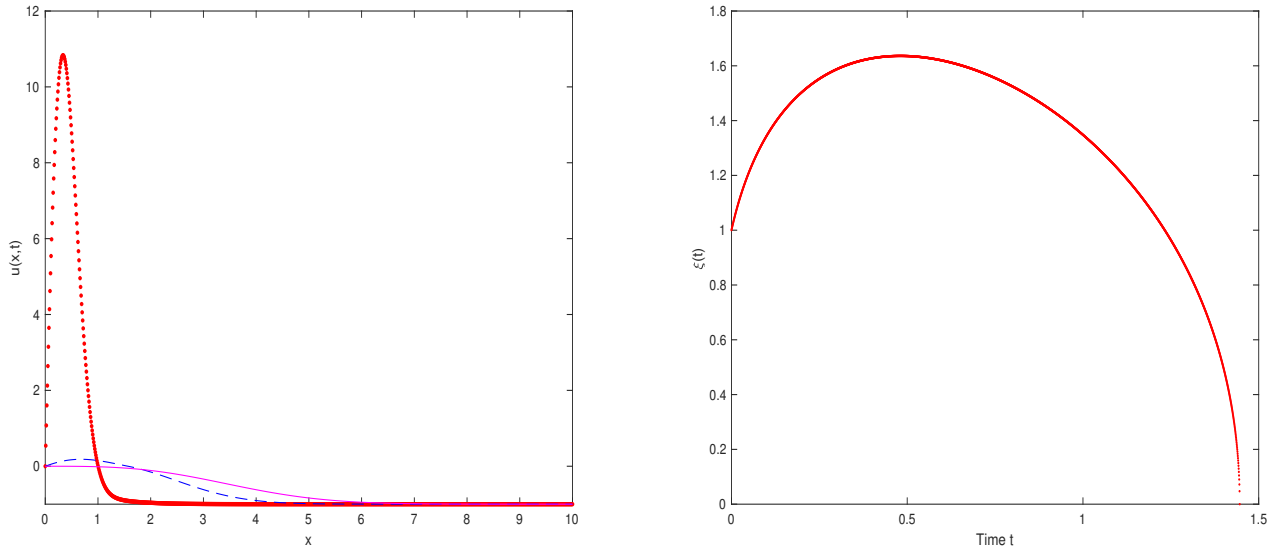

Figure 9: Evolution for the initial data in (7) with $\alpha = 4$. Left: $u(x,t)$ versus x for times $t = 0$, $t = 0.7225$, and $t = 1.4455$. Right: evolution of $\xi(t)$ versus t.

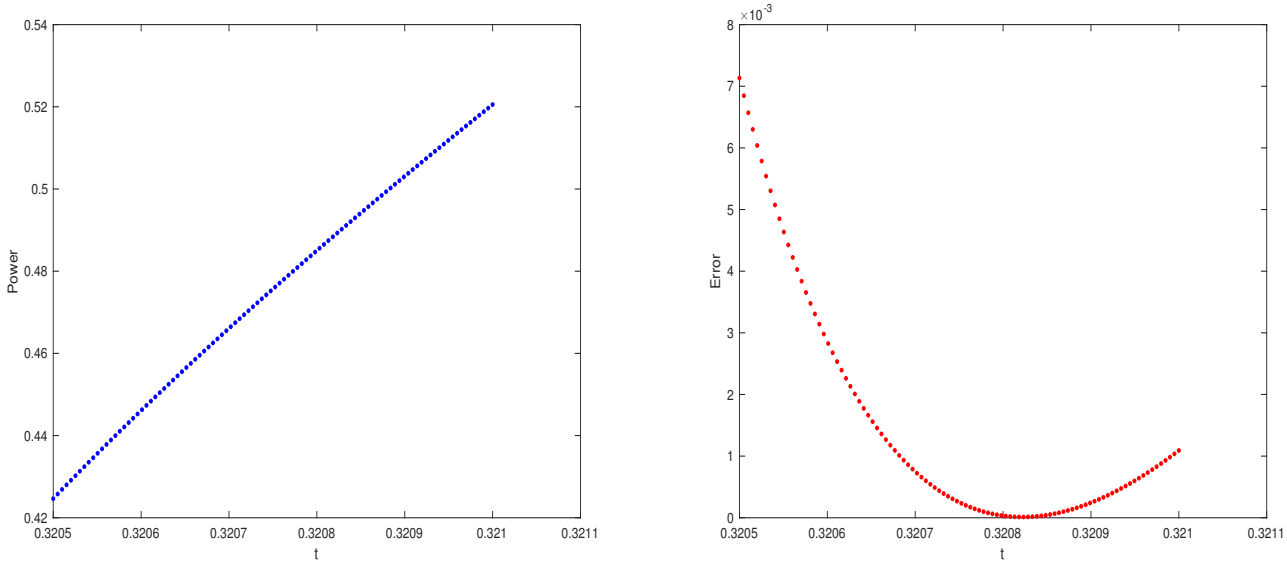

Figure 10: Left: power of the linear regression for Figure 7. Right: approximation error verses t_0.

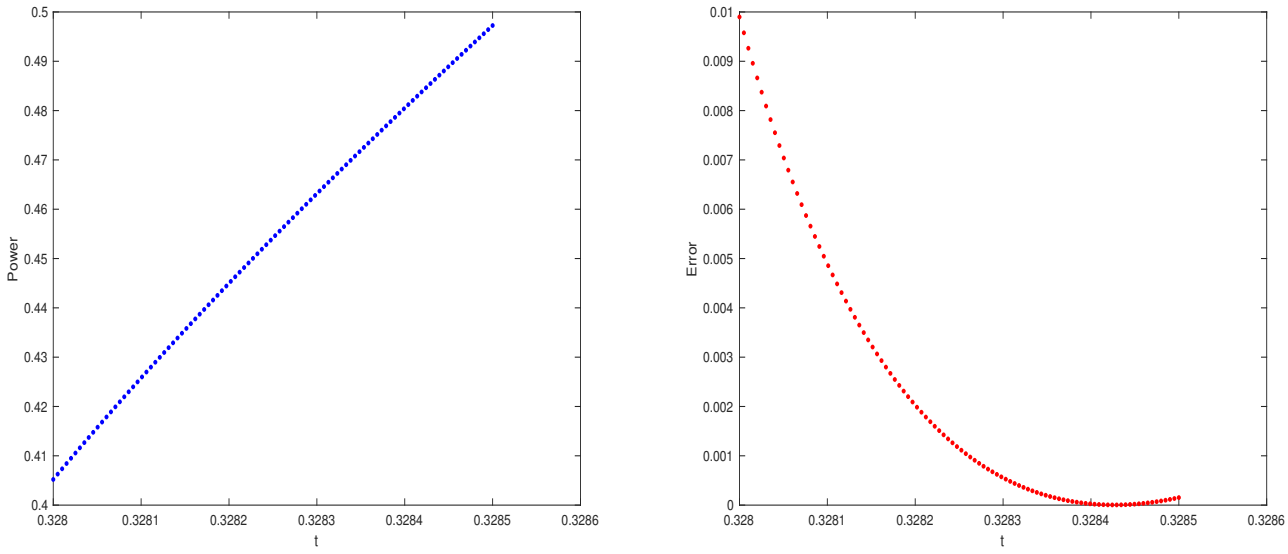

Figure 11: Left: power of the linear regression for Figure 8. Right: approximation error verses t_0.

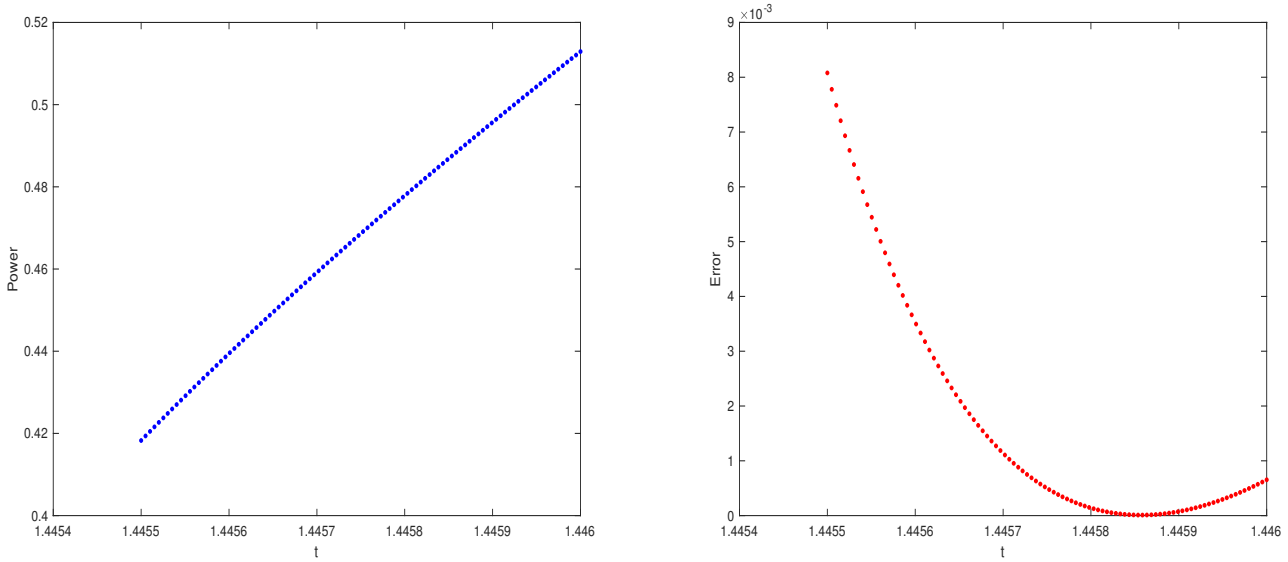

Figure 12: Left: power of the linear regression for Figure 9. Right: approximation error verses t_0.