Finite Element Methods: Kuramoto-Sivashinsky PDE Discretization to investigate Regularization Methods

Tahmid Tamim, RIN: 662088478

December 9, 2024

Abstract

This report presents the implementation of finite element methods to solve the 4^{th} order non-linear Kuramoto–Sivashinsky equation describing flame-front dynamics. The project employs finite element methods and implicit time-stepping discretization for solving the PDE. The objective of the project is to gain understanding of the Kuramoto-Sivashinsky equation, which will be potentially useful for researching regularization methods.

1 Background

The Kuramoto-Sivashinsky equation models flame-front dynamics. This work relies on Crank-Nicholson time-marching scheme and high-order Lagrangian finite elements to solve the equation over time in Python Dolfinx and PETSc. A 2D version of the Kuramoto-Sivashinsky solver is also implemented and simulated. The work done in this project is potentially useful to understanding and researching regularization methods authored by Professor Jason Hicken.

2 Methods

2.1 Original Partial Differential Equation

The Kuramoto-Sivashinsky equation:

$$u_t + u^{\mathrm{m}} u_x + \alpha u_{xx} + \beta u_{xxx} + \nu u_{xxxx} = 0 \tag{1}$$

2.2 Reduced Partial Differential Equation

Set $\beta = 0$, $\alpha = 1$, $\nu = 1$, $m = 1$ The Kuramoto-Sivashinsky equation:

$$u_t + u u_x + u_{xx} + \nu u_{xxxx} = 0 \tag{2}$$

Here:

- u: Flame-front amplitude
- ν: Diffusion parameter

- uu_x: Nonlinear advection term
- u_{xx}: 2nd order diffusion term
- u_{xxxx}: 4th order diffusion term

2.3 Weak Formulation

The weak form integrates the equation against a test function v, with stabilization to handle high-order terms:

$$\int_\Omega \left(\frac{u^{n+1} - u^n}{\Delta t} v + \theta \mathcal{L}(u^{n+1}) v + (1-\theta)\mathcal{L}(u^n) v + \mathcal{N}(u^{n+1}) v + \text{Stabilization} \right) dx = 0 \tag{3}$$

2.4 Numerical Implementation

- **Domain:** [0, 32] with 256 elements.
- **Function Space:** Continuous Galerkin (CG) with degree 4 polynomials.
- **Gaussian Initial Condition:**
$$u(x,0) = 4.0 \exp\left(-0.5 \left(\frac{x - 14.55}{2} \right)^2 \right) \sin\left(\frac{2\pi x}{32} \right) \tag{4}$$
- **Dirichlet Boundary Conditions:** $u(0) = u(L) = 0$.
- **Time-Marching:** Implicit Crank-Nicholson method with $\theta = 0.5$, $\Delta t = 0.005$, and a total simulation time $T = 100$.
- **Stabilization Term:**
$$h^2 \int_\Omega \nabla u_{n+1} \cdot \nabla v \, dx,$$
$$h = \frac{1}{N}$$
  where N is the number of elements in the grid.

2.5 Solver Configuration

- **Nonlinear Solver:** Newton's method.
- **Linearization Techniques:** GMRES with ILU preconditioner.
- **Convergence Criteria:** relative tolerance = 10^{-2}, absolute tolerance = 10^{-3}.

2.6 Initial Conditions

The initial condition stated above introduces an initial Gaussian perturbation into the system. This Gaussian perturbation models the initial stages of an evolving flame front.

3 1D Results

3.1 Evolution of the 1D Flame Front

The contour plot (Figure 1) illustrates the normalized amplitude over time and space. The displacements get smaller and smaller over time. Other notable features are periodic doubling pattern of the solution.

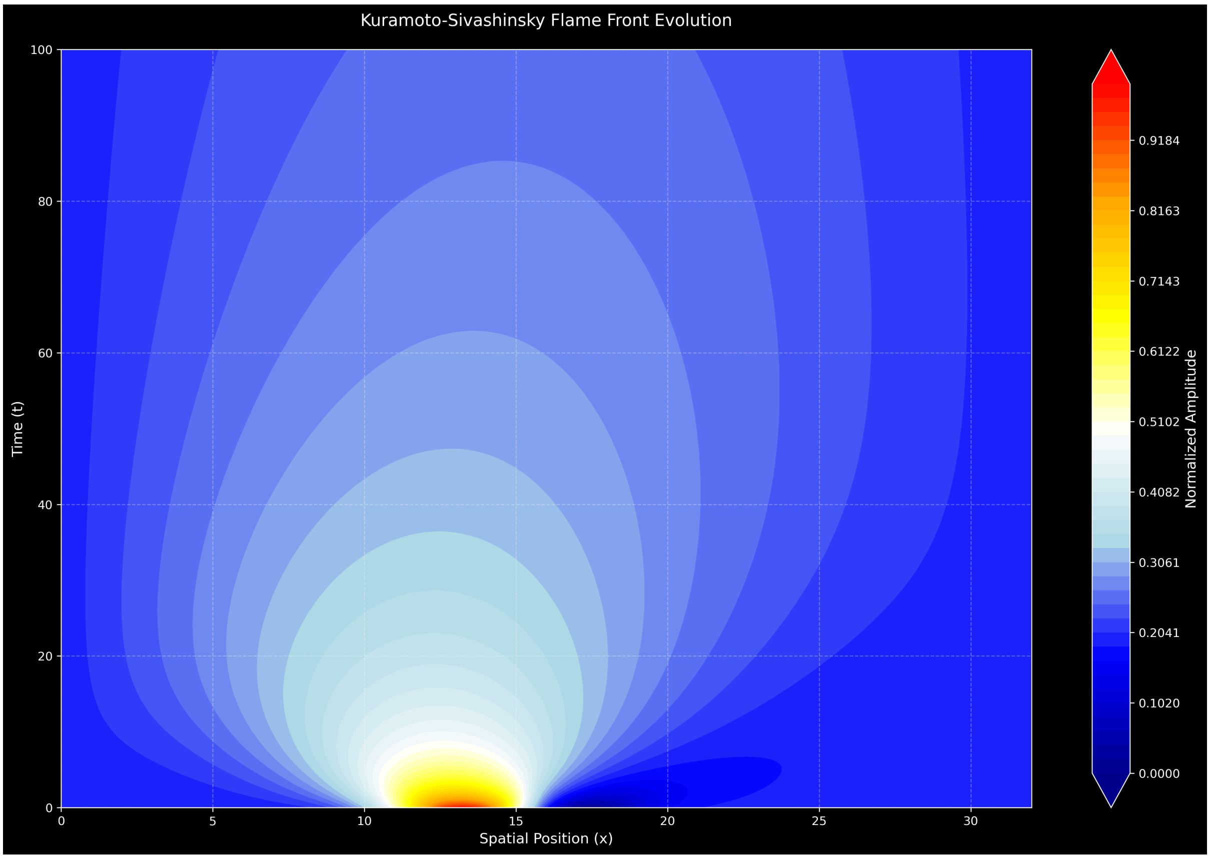

Figure 1: Kuramoto-Sivashinsky Flame Front Evolution.

3.2 Statistics

- Maximum Displacement: 1.6673
- Minimum Displacement: -0.3922
- Final time reached: 100.0 s

3.3 Snapshots of the 1D Solution

Figure 2 shows flame-front snapshots at selected times (0, $T/3$, $2T/3$, T). It highlights how the flame front perturbation evolves over time in shape and magnitude.

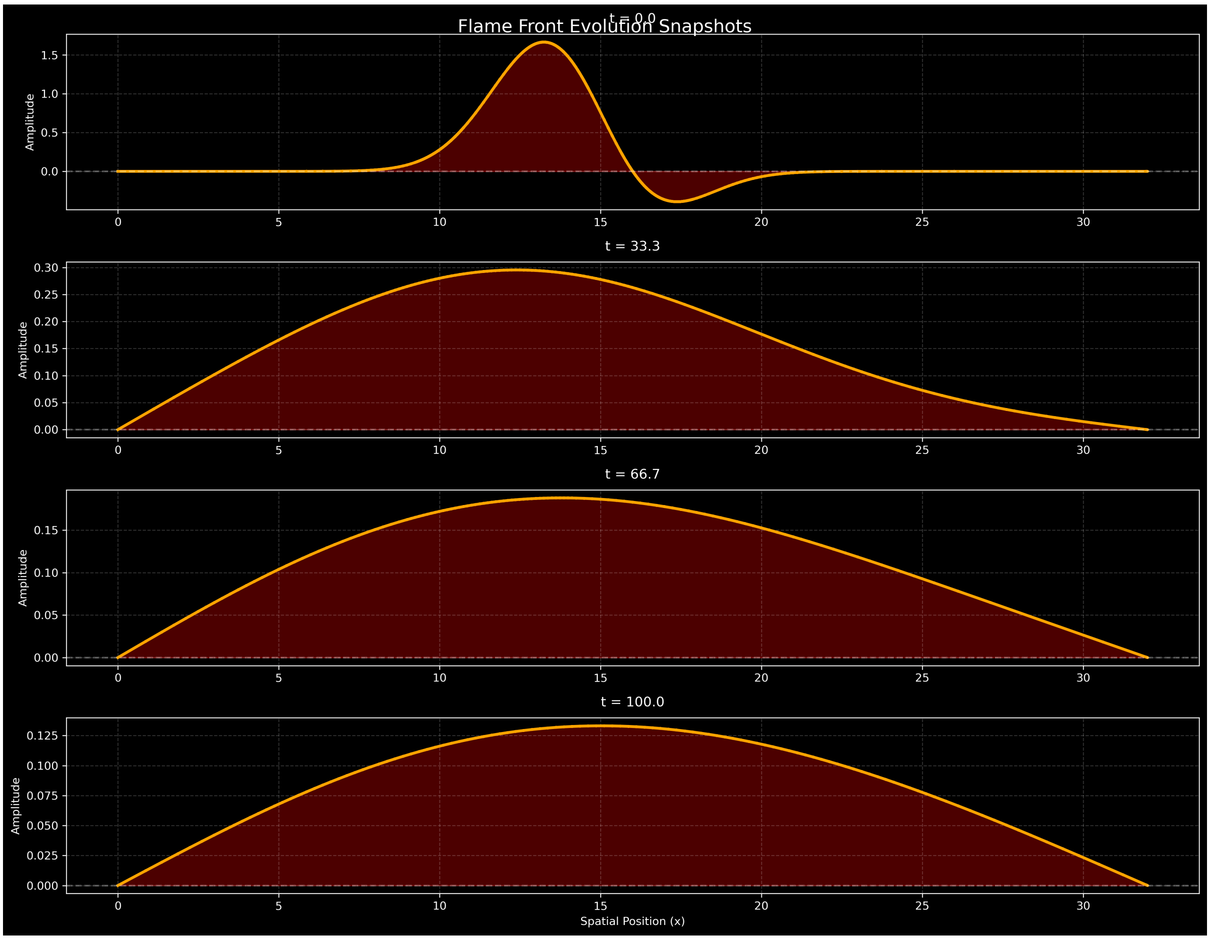

Figure 2: Kuramoto-Sivashinsky Flame Front Snapshots.

The vertical displacements gradually decrease in amplitude as demonstrated by the solution snapshots plotted over time. Due to the boundary condition constraints set at $u(0) = 0$ and $u(L) = 0$, we get a nice sinusoidal plot to interpret the results. To visualize more chaos inherent in the Kuramoto-Sivashinsky equation, it's recommended to lift the boundary constraints at either $x = 0$ or $x = L$. Please check github to see more chaotic plots.

3.4 Verification - 1D Energy Plot

The energy at time t is tentatively defined as:

$$E(t) = \int_{\Omega} u(x,t)^2 \, dx,$$

where Ω is the spatial domain.

The energy equation is an indirect measure of the energy within the system. It's useful to visualize how the energy evolves over time to see if and where it converges.

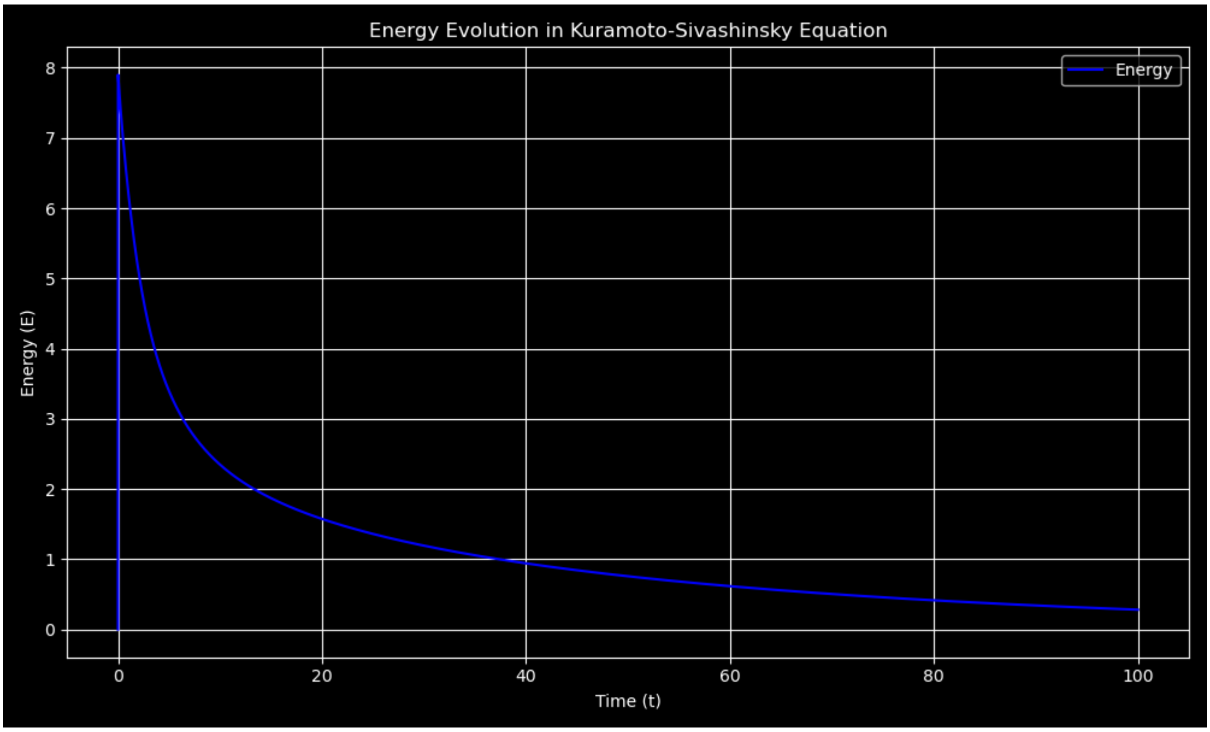

Figure 3: Kuramoto-Sivashinsky Flame Front Energy Convergence.

The energy starts at peak at the initial condition, then it gradually decreases to 0 as expected due to the forcing term being set to 0. Fixed Dirichlet boundary conditions on both ends also help to dissipate oscillations or perturbations that might otherwise display chaotic behavior. Eventually, the energy converges to 0 as the solution $u(x,t)$ becomes zero everywhere (trivial steady state). Thus, the integral $E(t) = \int_{\Omega} u(x,t)^2 \, dx$ converges to 0.

4 2D Results

4.1 Evolution of the 2D Flame Front

The Kuramoto-Sivashinsky equation is also solved in 2D using 2nd-order Triangular Continuous Galerkin Finite Elements. This is additionally implemented based on project proposal feedback from the TA. Similar to the 1D case, the 2D case also has an initial Gaussian perturbation which models the initial stages of an evolving flame front.

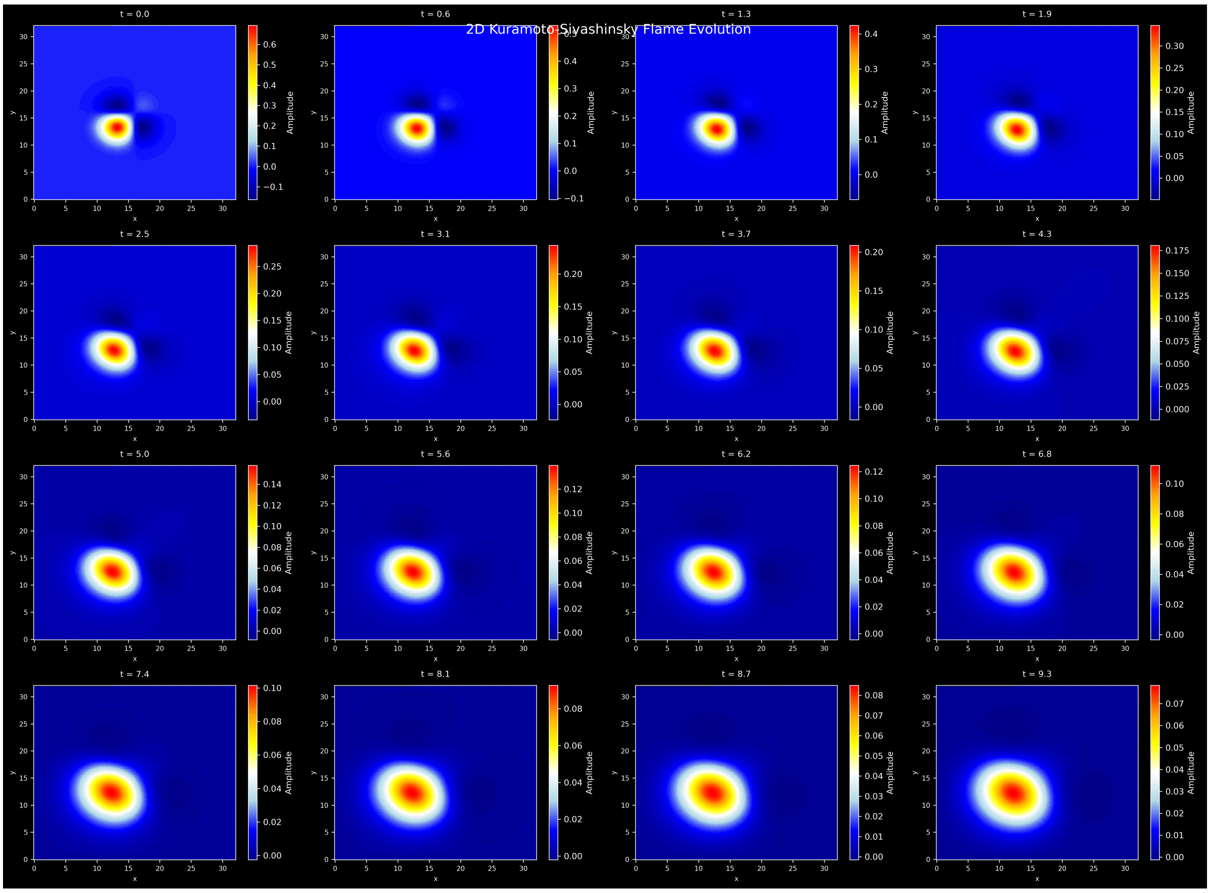

Figure 4: Kuramoto-Sivashinsky 2D Flame Front Evolution

The 2D results are similar to 1D results which shows the flame dissipating over time. However, the 2D results are more interesting as while the flame is dissipating with its core intact, it also spreads out over the spatial dimensions as it gradually burns out or goes to 0.

5 Conclusion

This project demonstrates how finite element methods are used in Dolfinx in solve the Kuramoto-Sivashinsky partial differential equation. The results provide useful insights into the evolving dynamics of a flame front perturbation.

6 File Appendix

1. `ks.py` - File contains PDE solver
2. `ks_energy_evolution.png` - Energy Evolution Plot.
3. `ks_flame_evolution_dolfinx.png` - Flame Evolution Plot.
4. `ks_flame_snapshots_dolfinx.png` - Flame Snapshots at Selected times.
5. `ks_flame_evolution_dolfinx_u0=0.png` - Flame Evolution for Dirichlet boundary condition only at $u(0) = 0$
6. `ks_flame_evolution_dolfinx_uL=0.png` - Flame Evolution for Dirichlet boundary condition only at $u(L) = 0$
7. `ks_2d_flame_evolution.png` - Shows flame evolution in 2D.

7 How to Run Code

1. Github Repository - `https://github.com/h7474/finite-elements-project`
2. Run Kuramoto-Sivashinsky solver with **python ks.py**