Modeling Phase Separation in 1D Using Nonlinear Finite Element Techniques

Course : Non-Linear Finite Element Method

Assignment for summer term 2022

Documentation

NANDHA GOPAL MARIAPPAN

Matriculation number:66994

Coded in Matlab (v2022)

Theory:-

- From the given assignment, it is known that the given one-dimensional Cahn-Hilliard equation is of the strong form

$$\dot{c} = \left[M(c)[\theta_c \Psi(c) - \lambda c^2]\right] \quad \textbf{[1]}$$

- The original **strong form** of the Cahn-Hilliard equation, which involves **higher-order spatial derivatives**, was reformulated into a **weak form** using **test (weighting) functions**.

- This variational formulation allows the application of **Galerkin's method**, enabling the equation to be solved numerically using the **finite element method (FEM)**.

- In the weak form, the problem is decomposed into an **element routine**, which captures the local stiffness and residual contributions, and a **material routine**, which defines the constitutive behavior (free energy and mobility functions).

$$0 = \delta W = \int_0^L \left[\dot{c}\delta c + M(c)\,\partial_{cc}^2\Psi(c)\,c'\delta c' + \partial_c M(c)\,\lambda c''c'\delta c' + M(c)\,\lambda c''\delta c''\right] \mathrm{d}x + \left[\bar{H}^c\delta c + \left[M(c)\,\lambda c' - \bar{H}^\xi\right]\delta c'' - r\delta c'\right]_{x=0}^L \quad \textbf{[2]}$$

- Each 2-node element approximates the concentration field using **cubic Hermite shape functions**, allowing for a smooth representation of both the function and its derivatives.

$$[N]^{Herm}(\xi) = \left[\frac{1}{4}(2+\xi)(\xi-1)^2, \frac{h^e}{8}(\xi+1)(\xi-1)^2, \frac{1}{4}(2-\xi)(\xi+1)^2, \frac{h^e}{8}(\xi-1)(\xi+1)^2\right]$$

$$in\ \Omega_\Pi = \{\xi \in [-1,1]\} \quad \textbf{[3]}$$

- The **global coordinates** are approximated using **linear Lagrange polynomials**.

$$[N]^{Lagr}(\xi) = \left[\tfrac{1}{2}(1-\xi), \tfrac{1}{2}(1+\xi)\right]\ in\ \Omega_\Pi = \{\xi \in [-1,1]\} \quad \textbf{[4]}$$

- Using the defined **shape functions**, the **element routine** is derived. The **material routine** is based on the given **mobility** and **free energy functions**.
  Four Gauss quadrature points are used per element for numerical integration.

- The nonlinear system is solved using the **Euler implicit method**, with convergence ensured by the **Newton-Raphson method** and specified criteria.

- Condition 1:-

$$\left\| \blacktriangle \hat{\underline{\mathbf{c}}}_k \right\|_\infty < 10^{-5} * \left\| \hat{\underline{\mathbf{c}}} \right\|_\infty \quad \textbf{[5]}$$

- Condition 2:-

$$\left\| \hat{\underline{\mathbf{R}}} \right\|_\infty < 0.005 * \max\left(\left\| \hat{\underline{\mathbf{F}}}_{\text{int}} |_{k=0} \right\|_\infty, 10^{-8} \right) \quad \textbf{[6]}$$

- Once converged, the **evolved concentration profile over time** is plotted and compared to the **initial concentration**.

Program Structure :-

- The program begins by **taking user inputs** for: **Element size and Total simulation time.**
- The entered time value should be **greater than 6000** to observe the long-term evolution behavior.
- **Constants** required for the **element** and **material routines** are initialized next.
- The core of the program consists of **two nested loops**: **FOR loop 1** – Time stepping loop and **FOR loop 2** – Global element assembly loop
  - This loop is nested inside a **WHILE loop**, which continues until **convergence criteria** are met.
  - If convergence is not achieved, the concentration variable `ccap` is updated, and **FOR loop 2** runs again.
- After convergence is achieved for a time step, the program returns to **FOR loop 1** to process the **next time step**.This continues until the **end time** is reached.

- The **material routine** (including the **free energy function**, its derivatives, and the **mobility function**) is implemented as **separate functions** and called within the main loop.
- The computed **concentration results** are stored in the variable `Cglobnew`.

Results:-

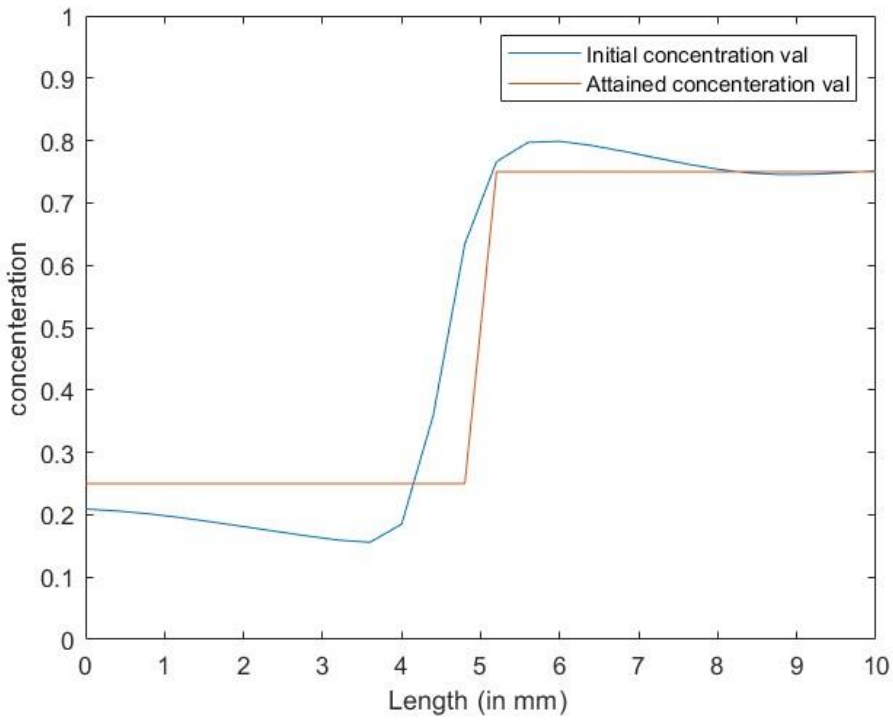

Figure.1: Evolved concentration vs. Element Length

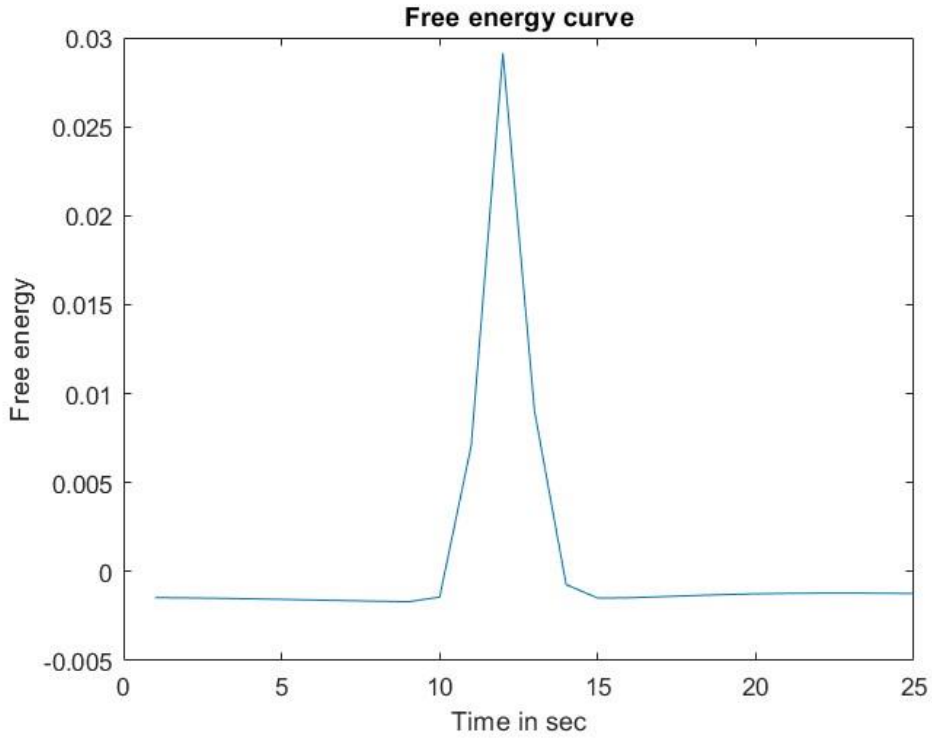

Figure .2: Free energy vs. Time

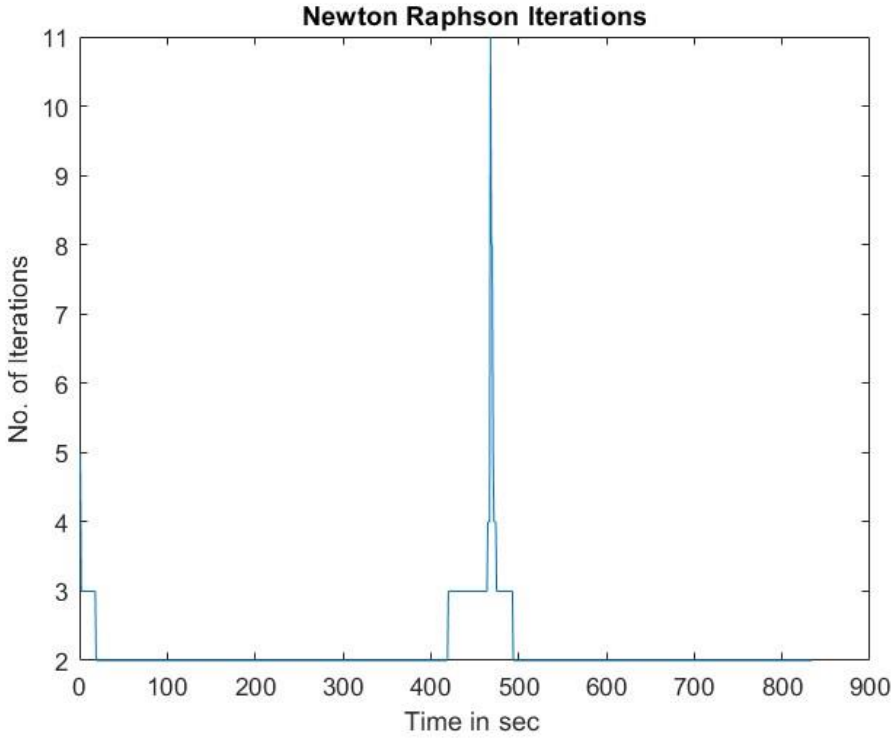

Figure .3: Newton-Raphson iterations vs. Time

- As per my matriculation number,t is greater than 6000 and the size of the element is h<0.11.

- The above graphs are plotted for the t=2000 sec and the size of the element (h) =0.4mm since the given desired time steps and size of the element are unable to plot the desired graph.

- Iterative solvers like **PCG** and **CGS** did not converge, and only MATLAB's **`mldivide` operator (\)** successfully solved the system, allowing the while loop to converge.