Mihai Research

Rongze Wei

2025.1.31

The equation that we have is

$$u_t = \delta u_{xx} - \delta^{-1} V'(u), V(u) = \frac{1}{4}(1-u^2)^2$$

So, by using the finite difference scheme, the second-order centred differences for partial derivative and the Euler method in time,

$$u_{xx} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}, u_t = \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

So, the equation becomes that, for $\forall j \in 1, 2, 3, \cdots, N-1 \quad \text{and} \quad \forall n \in 0, 1, \cdots, T-2$, we have

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \delta(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}) - \frac{1}{\delta}((u_j^n)^3 - u_j^n)$$

which is equivalent to

$$\begin{aligned} u_j^{n+1} &= u_j^n + \frac{\delta \Delta t}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) - \frac{\Delta t}{\delta}((u_j^n)^3 - u_j^n) \\ &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_j^n + \frac{\delta \Delta t}{h^2} u_{j+1}^n + \frac{\delta \Delta t}{h^2} u_{j-1}^n - \frac{\Delta t}{\delta}(u_j^n)^3 \end{aligned}$$

Since we have the Dirichlet boundary condiion that

$$u(0,t) = u(1,t) = 0, u(x,0) = u_+(x), u(x,T) = u_-(x)$$

so in our numerical simulation, we have 100×100 points in discretization, so $N = 99, T = 99$.

Note that u_j^n denotes the point at the j-th position at time n.

We only need to consider the case when $j = 1, 2, \cdots, 98$ and $n = 0, 1, \cdots, 97$.

Thus, the system of equations that we have is:
For $n = 0$,

$$\begin{cases} u_1^1 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^0 + \frac{\delta \Delta t}{h^2} u_2^0 + \frac{\delta \Delta t}{h^2} u_0^0 - \frac{\Delta t}{\delta}(u_1^0)^3 = (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^0 + \frac{\delta \Delta t}{h^2} u_2^0 - \frac{\Delta t}{\delta}(u_1^0)^3 \\ u_2^1 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_2^0 + \frac{\delta \Delta t}{h^2} u_3^0 + \frac{\delta \Delta t}{h^2} u_1^0 - \frac{\Delta t}{\delta}(u_2^0)^3 \\ \vdots & \\ u_{98}^1 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^0 + \frac{\delta \Delta t}{h^2} u_{99}^0 + \frac{\delta \Delta t}{h^2} u_{97}^0 - \frac{\Delta t}{\delta}(u_{98}^0)^3 = (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^0 + \frac{\delta \Delta t}{h^2} u_{97}^0 - \frac{\Delta t}{\delta}(u_{98}^0)^3 \end{cases}$$

For $n = 1$,

$$\begin{cases} u_1^2 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^1 + \frac{\delta \Delta t}{h^2} u_2^1 + \frac{\delta \Delta t}{h^2} u_0^1 - \frac{\Delta t}{\delta}(u_1^1)^3 = (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^1 + \frac{\delta \Delta t}{h^2} u_2^1 - \frac{\Delta t}{\delta}(u_1^1)^3 \\ u_2^2 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_2^1 + \frac{\delta \Delta t}{h^2} u_3^1 + \frac{\delta \Delta t}{h^2} u_1^1 - \frac{\Delta t}{\delta}(u_2^1)^3 \\ \vdots & \\ u_{98}^2 &= (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^1 + \frac{\delta \Delta t}{h^2} u_{99}^1 + \frac{\delta \Delta t}{h^2} u_{97}^1 - \frac{\Delta t}{\delta}(u_{98}^1)^3 = (1 + \frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^1 + \frac{\delta \Delta t}{h^2} u_{97}^1 - \frac{\Delta t}{\delta}(u_{98}^1)^3 \end{cases}$$

For $n = 97$,

$$\begin{cases} u_1^{98} &= (1+\frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^{97} + \frac{\delta\Delta t}{h^2}u_2^{97} + \frac{\delta\Delta t}{h^2}u_0^{97} - \frac{\Delta t}{\delta}(u_1^{97})^3 = (1+\frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_1^{97} + \frac{\delta\Delta t}{h^2}u_2^{97} - \frac{\Delta t}{\delta}(u_1^{97})^3 \\ u_2^{98} &= (1+\frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_2^{97} + \frac{\delta\Delta t}{h^2}u_3^{97} + \frac{\delta\Delta t}{h^2}u_1^{97} - \frac{\Delta t}{\delta}(u_2^{97})^3 \\ \vdots & \\ u_{98}^{98} &= (1+\frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^{97} + \frac{\delta\Delta t}{h^2}u_{99}^{97} + \frac{\delta\Delta t}{h^2}u_{97}^{97} - \frac{\Delta t}{\delta}(u_{98}^{97})^3 = (1+\frac{\Delta t}{\delta} - 2\frac{\delta \Delta t}{h^2})u_{98}^{97} + \frac{\delta\Delta t}{h^2}u_{97}^{97} - \frac{\Delta t}{\delta}(u_{98}^{97})^3 \end{cases}$$

If we write the iteration in the form of a linear system, we would have,

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_{98}^{n+1} \end{bmatrix} = \begin{bmatrix} 1+\frac{\Delta t}{\delta} - 2\frac{\delta\Delta t}{h^2} & \frac{\delta\Delta t}{h^2} & 0 & 0 & \cdots & 0 \\ \frac{\delta\Delta t}{h^2} & 1+\frac{\Delta t}{\delta} - 2\frac{\delta\Delta t}{h^2} & \frac{\delta\Delta t}{h^2} & 0 & \cdots & 0 \\ 0 & \frac{\delta\Delta t}{h^2} & 1+\frac{\Delta t}{\delta} - 2\frac{\delta\Delta t}{h^2} & \frac{\delta\Delta t}{h^2} & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \frac{\delta\Delta t}{h^2} & 1+\frac{\Delta t}{\delta} - 2\frac{\delta\Delta t}{h^2} & \frac{\delta\Delta t}{h^2} \\ 0 & 0 & 0 & 0 & \frac{\delta\Delta t}{h^2} & 1+\frac{\Delta t}{\delta} - 2\frac{\delta\Delta t}{h^2} \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ u_{98}^n \end{bmatrix} + \begin{bmatrix} (u_1^n)^3 \\ (u_2^2)^3 \\ (u_3^n)^3 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ (u_{98}^n)^3 \end{bmatrix}$$

After discuss with Prof Mihai, we decided to set

$$u_+(x) = u(x,0) = \sin(\pi x), u_-(x) = -\sin(\pi x)$$

, and we discretize the space-time domain $[0,1]\times[0,1]$ with sizes $\Delta x = \frac{1}{99}$, $\Delta t = \frac{1}{99}$.
To make our life easier, we just try to construct the function $u(x,t)$ on the domain as simple as possible with the boundary condition above. So, we set

$$\begin{aligned} u(x,t) &= -t\sin(\pi x) + (1-t)\sin(\pi x) \\ &= -2t\sin(\pi x) + \sin(\pi x) \\ &= (1-2t)\sin(\pi x) \end{aligned}$$

Firstly I use Matlab to plot the 3D plot of this function on the domain. And then I get the matrix U, where the last row of U is $u(x,0)$ while the top row is $u(x,1)$.

```
x = linspace(0, 1, 100);
t = linspace(0, 1, 100);

U = zeros(100, 100);

for idx_t = 1:length(t)
    current_t = t(idx_t);
    U(101 - idx_t, :) = (1 - 2 * current_t) * sin(pi * x);
end

[X, T] = meshgrid(x, t);

figure;
surf(X, T, flipud(U));
xlabel('x');
ylabel('t');
zlabel('u(x,t)');
title('u(x,t) = (1 - 2t)sin(\pi x)');
shading interp;
colorbar;
```

And then I get the matrix U, where the row on the top is $u(x,0)$ while the row in the bottom is $u(x,1)$.
Next we try to find matrix P where P is the numerical approximation to

$$p(x,t) = u_t - \delta u_{xx} + \delta^{-1} V'(u)$$

The formula is given by

$$P_{i,j+\frac{1}{2}} = \frac{U_{i,j+1} - U_{i,j}}{\Delta t} + \delta^{-1} V'\left(\frac{U_{i,j+1} + U_{i,j}}{2}\right) - \frac{\delta}{2}\left(\frac{U_{i+1,j+1} - 2U_{i,j+1} + U_{i-1,j+1}}{\Delta x^2} + \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta x^2}\right)$$

where $i = 1, \cdots, 98, \quad j = 0, \cdots, 98, \quad \delta = 0.05$, $\Delta x = \frac{1}{99}$, $\Delta t = \frac{1}{99}$ and recall that

$$V(u) = \frac{1}{4}(1-u^2)^2, V'(u) = u^3 - u$$

After we get P, we calculate $S_T(U)$ which is defined as

$$S_T(U) = \frac{1}{2}\Delta x \Delta t \sum_{i=1}^{98} \sum_{j=0}^{98} P_{i,j+\frac{1}{2}}^2$$

And since the boundary condition, S_T is actually a fucntion of variables $U_{i,j}$ where $i = 1, \cdots, 98$, $j = 1, \cdots, 98$, and the gradient of S_T with respect to each variale is given by

$$\begin{aligned}
\frac{\partial S_T}{\partial U_{i,j}} &= \left(2\Delta x + \frac{\Delta x \Delta t}{\delta} V''\left(\frac{U_{i,j-1} + U_{i,j}}{2}\right) + \frac{2\delta \Delta t}{\Delta x}\right) P_{i,j-\frac{1}{2}} \\
&- \left(2\Delta x - \frac{\Delta x \Delta t}{\delta} V''\left(\frac{U_{i,j} + U_{i,j+1}}{2}\right) - \frac{2\delta \Delta t}{\Delta x}\right) P_{i,j+\frac{1}{2}} \\
&- \frac{\delta \Delta t}{\Delta x}\left(P_{i-1,j-\frac{1}{2}} + P_{i-1,j+\frac{1}{2}} + P_{i+1,j-\frac{1}{2}} + P_{i+1,j+\frac{1}{2}}\right)
\end{aligned}$$

Well, actually the function S_T is a function of the whole matrix U, so there should be 100×100 random variables, but since the boundary condition, the boundarie are constant so their derivatives are 0.
So the gradient is

$$\nabla S_T(U) = \begin{bmatrix} \mathbf{0} \\ 0 \\ \frac{\partial S_T}{\partial U_{1,1}} \\ \vdots \\ \frac{\partial S_T}{\partial U_{98,1}} \\ 0 \\ 0 \\ \frac{\partial S_T}{\partial U_{1,2}} \\ \vdots \\ \frac{\partial S_T}{\partial U_{98,2}} \\ 0 \\ \vdots \\ \\ 0 \\ \frac{\partial S_T}{\partial U_{1,98}} \\ \vdots \\ \frac{\partial S_T}{\partial U_{98,98}} \\ 0 \\ \mathbf{0} \end{bmatrix}$$

and the **BFGS** method is presented as follows,
U_0 = U
For $k = 0, 1, 2, ...,$

$U_{k+1} = U_k + \alpha_k p_k$
where the step length α_k is chosen by the wolfe condition

$$S_T(U_{k+1}) \leq S_T(U_k) + c_1\alpha_k \nabla S_T(U_k)^T p_k$$

and

$$\nabla S_T(U_{k+1})^T p_k \geq c_2 \nabla S_T(U_k)^T p_k$$

with $0 < c_1 < c_2 < 1$
The descent direction p_k is given by

$$p_k = -H_k \nabla f_k$$

where H_k is defined recursively by

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

where $s_k = U_{k+1} - U_k$, $y_k = \nabla S_T(U_{k+1}) - \nabla S_T(U_k)$, and $\rho_k = \frac{1}{y_k^T s_k}$.
For H_0, we firstly consider

$$H_0 = \gamma_1 I$$

where $\gamma_1 = \frac{y_0^T s_0}{y_0^T y_0}$

The algorithm is follows

Algorithm 1 BFGS Method

1: **Input:** Initial matrix $U_0 = U$, tolerance $\epsilon = 10^{-4}$, $\delta = 0.05$, $\Delta x = \frac{1}{99}$, $\Delta t = \frac{1}{99}$.
2: **Initialize:** $H_0 = \gamma_1 I$, where $\gamma_1 = \frac{y_0^T s_0}{y_0^T y_0}$.
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Compute descent direction: $p_k = -H_k \nabla S_T(U_k)$.
5: Choose step length α_k satisfying the Wolfe conditions:

$$S_T(U_{k+1}) \leq S_T(U_k) + c_1\alpha_k \nabla S_T(U_k)^T p_k,$$
$$\nabla S_T(U_{k+1})^T p_k \geq c_2 \nabla S_T(U_k)^T p_k,$$

where $0 < c_1 < c_2 < 1$.
6: Update U: $U_{k+1} = U_k + \alpha_k p_k$.
7: Compute $s_k = U_{k+1} - U_k$ and $y_k = \nabla S_T(U_{k+1}) - \nabla S_T(U_k)$.
8: Compute $\rho_k = \frac{1}{y_k^T s_k}$.
9: Update H_k by:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T.$$

10: **if** $\|\nabla S_T(U_{k+1})\| \leq \epsilon$ **then**
11: **Stop** and return U_{k+1}.
12: **end if**
13: **end for**

Then, we move to finding the initial boundary condition for $u_+(x,0)$ and $u_-(x,0)$. These two functions come from the stable equilibrium states of the function

$$\mathbb{E}[u] = \frac{1}{2}\int_0^1 (\delta u_x^2 + 2\delta^{-1}\mathbb{V}(u))dx$$

Similarly, we set $q(x) = \delta u_x^2 + 2\delta^{-1}\mathbb{V}(u)$, so the energy function can be written as

$$\mathbb{E}[u] = \frac{1}{2}\int_0^1 q(x)dx$$

We use the midpoint rule to compute the temporal integral, and obtaining

$$E(U) = \frac{1}{2}\Delta x \sum_{i=1}^{I-1} P_i^2$$

and since here we have 100 points so $I = 100$, and P_i is the numerical approximation of $q(x_i)$ by central finite difference

$$P_i = \delta^{-1}\mathbb{V}(U_i) + \delta(\frac{U_{i+1} - U_{i-1}}{2\Delta x})^2$$

For the boundary condition of U, we have $U_0 = U_{100} = 0$ and the BFGS method requires the gradient, which is given by

$$\frac{\partial E}{\partial U_1} = \frac{1}{2}\Delta x \left[\delta^{-1}\frac{1}{2}(1 - U_1^2)(-2U_1) + 2\delta(\frac{U_3 - U_1}{2\Delta x})(-\frac{1}{2\Delta x})\right]$$

$$\frac{\partial E}{\partial U_2} = \frac{1}{2}\Delta x \left[2\delta\frac{U_2}{2\Delta x}(\frac{1}{2\Delta x}) + \delta^{-1}\frac{1}{2}(1 - U_2^2)(-2U_2) + 2\delta(\frac{U_4 - U_2}{2\Delta x})(-\frac{1}{2\Delta x})\right]$$

$$\frac{\partial E}{\partial U_i} = \frac{1}{2}\Delta x \left[2\delta(\frac{U_i - U_{i-2}}{2\Delta x})(\frac{1}{2\Delta x}) + \delta^{-1}\frac{1}{2}(1 - U_i^2)(-2U_i) + 2\delta(\frac{U_{i+2} - U_i}{2\Delta x})(-\frac{1}{2\Delta x})\right], \quad \text{for} \quad i = 3, \cdots, 97$$

$$\frac{\partial E}{\partial U_{98}} = \frac{1}{2}\Delta x \left[2\delta(\frac{U_{98} - U_{96}}{2\Delta x})(\frac{1}{2\Delta x}) + \delta^{-1}\frac{1}{2}(1 - U_{98}^2)(-2U_{98}) + 2\delta\frac{-U_{98}}{2\Delta x}(-\frac{1}{2\Delta x})\right]$$

$$\frac{\partial E}{\partial U_{99}} = \frac{1}{2}\Delta x \left[2\delta(\frac{U_{99} - U_{97}}{2\Delta x})(\frac{1}{2\Delta x}) + \delta^{-1}\frac{1}{2}(1 - U_{99}^2)(-2U_{99})\right]$$

And the BFGS method present as follow. $\mathbf{U}_0 = \sin(\pi x), H_0 = I$
The BFGS method defines the next one as

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \alpha_k p_k$$

, where α_k is chosen to satisfy the wolfe conditions

$$E(U_{k+1}) \leq E(\mathbf{U}_k) + c_1\alpha_k \nabla E(\mathbf{U}_k)^T p_k$$

and

$$\nabla E(\mathbf{U}_{k+1})^T p_k \geq c_2 \nabla E(\mathbf{U}_k)^T p_k$$

with $0 < c_1 < c_2 < 1$
The search direction p_k is given by

$$p_k = -H_k \nabla E_k$$

where H_k is defined recursively by

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

, where

$$s_k = E(\mathbf{U}_{k+1}) - E(\mathbf{U}_k), \quad y_k = \nabla E_{k+1} - \nabla E_k, \quad \rho_k = \frac{1}{y_k^T s_k}$$

and finally the iteration would stop if the norm of the gradient is smaller than $\epsilon = 10^{-4}$.