

Math 5604 Final Project

Jacob Hauck

May 5, 2024

1 Introduction

In this project, we consider the two-dimensional Allen-Cahn equation [1] with periodic boundary conditions, which is given by

$$u_t = \varepsilon^2 \Delta u - F(u), \quad (x, y) \in (0, 2\pi)^2, \quad t \in (0, T] \quad (1)$$

$$u(x, y, 0) = u_0(x, y), \quad (x, y) \in [0, 2\pi]^2 \quad (2)$$

$$u(x, 0, t) = u(x, 2\pi, t), \quad x \in [0, 2\pi], \quad t \in [0, T] \quad (3)$$

$$u(0, y, t) = u(2\pi, y, t), \quad y \in [0, 2\pi], \quad t \in [0, T] \quad (4)$$

where $\varepsilon > 0$ is a parameter, $F(u) = u^3 - u$, and $u_0(x, y) = 0.05 \sin(x) \sin(y)$.

This equation is a model of a reaction-diffusion system involving a mixture of two substances with different properties, such as oil and water or metals in certain alloys. It is meant to describe the phenomenon of phase separation, which is the process of the substances separating from a well-mixed initial state into coarser “cells” (see here for a visualization). The function u describes the relative density of one of the substances, with $u = 1$ corresponding to 100% density of one substance and $u = -1$ corresponding to 100% density of the other substance.

We test two temporal discretizations of the Allen-Cahn equation by comparing their accuracy and their ability to preserve the properties of solutions of the Allen-Cahn equation.

2 Discretization

We approximate $u(x, y, t)$ by the numerical solution $u_{i,j}^n$:

$$u(x_i, y_j, t_n) \approx u_{i,j}^n, \quad (5)$$

where we are using a uniform grid of $(M+1)^2$ spatial sample points $\{(x_i, y_j)\}$ defined by $x_i = hi$ for $i = 0, 1, \dots, M$ and $y_j = hj$ for $j = 0, 1, \dots, M$, where the spatial step size $h = \frac{2\pi}{M}$. We use N evenly-spaced time sample points $\{t_n\}$ defined by $t_n = kn$, where the temporal step size $k = \frac{T}{N}$.

We use the 5-point discretization of the Laplace operator, which is given in this case by

$$D_5 u_{i,j}^n = \frac{u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n}{h^2}, \quad i, j = 0, 1, \dots, M-1, \quad (6)$$

$$u_{-1,j}^n = u_{M-1,j}^n, \quad u_{0,j}^n = u_{M,j}^n, \quad j = 0, 1, \dots, M-1 \quad (7)$$

$$u_{i,-1}^n = u_{i,M-1}^n, \quad u_{i,0}^n = u_{i,M}^n, \quad i = 0, 1, \dots, M-1, \quad (8)$$

where the last four equations account for the periodic boundary conditions.

For the temporal discretization, we compare two different schemes. The first is the convex splitting scheme, given for the Allen-Cahn equation by

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{k} = \varepsilon^2 D_5 u_{i,j}^{n+1} + u_{i,j}^n - (u_{i,j}^{n+1})^3. \quad (9)$$

The second is the Crank-Nicolson scheme, which is given by

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{k} = \varepsilon^2 \frac{D_5 u_{i,j}^{n+1} + D_5 u_{i,j}^n}{2} + \frac{u_{i,j}^{n+1} + u_{i,j}^n}{2} - \frac{(u_{i,j}^{n+1})^3 + (u_{i,j}^n)^3}{2}. \quad (10)$$

3 Implementation

Let $\{a_{i,j}\}$ be a matrix with indices $i, j = 1, 2, \dots, M$. We pack this matrix into a vector $\{Va_m\}$, where $m = 1, 2, \dots, M^2$ by setting

$$Va_{m(i,j)} = a_{i,j}, \quad m(i,j) = (i-1)M + j. \quad (11)$$

Note that this is well-defined (that is, $m(i,j)$ is uniquely determined by the pair (i,j)) for $i, j = 1, 2, \dots, M$. In this case, it is easy to see that V is invertible with inverse

$$V^{-1}b_{i,j} = b_{m(i,j)}, \quad b \in \mathbf{R}^{M^2}, \quad i, j = 1, 2, \dots, M. \quad (12)$$

Defining the matrix of unknowns $U_{i,j}^n = u_{i-1,j-1}^n$ for $i, j = 1, 2, \dots, M$, we can rewrite our two schemes in vector form by applying V to both sides. In order to put everything in terms of VU^n , we must translate the action of D_5 into vector terms. Define D'_5 so that $D'_5 U_{i,j}^n = D_5 u_{i-1,j-1}^n$ by setting

$$D'_5 U_{i,j}^n = \frac{U_{i+1,j}^n + U_{i-1,j}^n + U_{i,j+1}^n + U_{i,j-1}^n - 4U_{i,j}^n}{h^2}, \quad i, j = 1, 2, \dots, M, \quad (13)$$

where we take $U_{0,j}^n = U_{M,j}^n$, $U_{M+1,j}^n = U_{1,j}^n$, $U_{i,0}^n = U_{i,M}^n$, and $U_{i,M+1}^n = U_{i,1}^n$ to match the periodic boundary conditions in D_5 . Furthermore, if we define $VD'_5 = V \circ D'_5 \circ V^{-1}$, then the convex-splitting scheme becomes

$$VU^{n+1} - VU^n = k\varepsilon^2 VD'_5 VU^{n+1} + kVU^n - k(VU^{n+1})^3, \quad (14)$$

where $n = 0, 1, \dots, N-1$, and the Crank-Nicolson scheme becomes

$$VU^{n+1} - VU^n = \frac{k\varepsilon^2}{2} (VD'_5 VU^{n+1} + VD'_5 VU^n) + \frac{k}{2} (VU^{n+1} + VU^n) - \frac{k}{2} ((VU^{n+1})^3 + (VU^n)^3), \quad (15)$$

where $n = 0, 1, \dots, N-1$.

In order to solve these two nonlinear systems for VU^{n+1} , we apply Newton's method using VU^n as the initial guess. For this, we need to define a function f_n whose root is VU^{n+1} and calculate its Jacobian. We can define these functions as follows:

$$f_n^{\text{CS}}(z) = z - k\varepsilon^2 VD'_5 z + kz^3 - (1+k)VU^n, \quad z \in \mathbf{R}^{M^2}, \quad (16)$$

and

$$f_n^{\text{CN}}(z) = \left(1 - \frac{k}{2}\right)z - \frac{k\varepsilon^2}{2} VD'_5 z + \frac{k}{2} z^3 - \frac{k\varepsilon^2}{2} VD'_5 VU^n - \left(1 + \frac{k}{2}\right)VU^n + \frac{k}{2}(VU^n)^3, \quad z \in \mathbf{R}^{M^2}. \quad (17)$$

Calculating the Jacobians of these functions is straightforward except for the $VD'_5 z$ terms:

$$Df_n^{\text{CS}}(z) = I - k\varepsilon^2 D(VD'_5 z) + \text{diag}(3kz^2), \quad (18)$$

and

$$Df_n^{\text{CN}}(z) = \left(1 - \frac{k}{2}\right)I - \frac{k\varepsilon^2}{2} D(VD'_5 z) + \text{diag}\left(\frac{3k\varepsilon^2}{2} z^2\right). \quad (19)$$

To evaluate $VD'_5 z$, we make use of (12) and the definition of D'_5 :

$$VD'_5 z_{m(i,j)} = V(D'_5 V^{-1}(z))_{m(i,j)} = (D'_5 V^{-1}(z))_{i,j} \quad (20)$$

$$= \frac{V^{-1}z_{i+1,j} + V^{-1}z_{i-1,j} + V^{-1}z_{i,j+1} + V^{-1}z_{i,j-1} - 4V^{-1}z_{i,j}}{h^2} \quad (21)$$

$$= \frac{z_{m(i+1,j)} + z_{m(i-1,j)} + z_{m(i,j+1)} + z_{m(i,j-1)} - 4z_{m(i,j)}}{h^2} \quad (22)$$

for $i = 1, 2, \dots, M$, where we account for the periodic boundary conditions by setting $m(0, j) = m(M, j)$, $m(M+1, j) = m(1, j)$, $m(i, 0) = m(i, M)$, and $m(i, M+1) = m(i, 1)$.

Now we can compute the Jacobian of $z \mapsto VD'_5 z$, which is a constant matrix:

$$D(VD'_5)_{m(i,j),m(i',j')} = \begin{cases} \frac{1}{h^2} & i = i' \text{ and } (j = j' + 1 \text{ or } j = j' - 1 \text{ or BC}) \\ \frac{1}{h^2} & j = j' \text{ and } (i = i' + 1 \text{ or } i = i' - 1 \text{ or BC}) \\ -\frac{4}{h^2} & i = i' \text{ and } j = j' \end{cases}$$

for $i, j, i', j' = 1, 2, \dots, M$. Since VD'_5 is linear, it follows that $VD'_5 z = D(VD'_5)z$, so in our implementation we only need to construct the constant (sparse) matrix $D(VD'_5)$, and then we can use sparse matrix-vector multiplication to compute $VD'_5 z$. See the comments and code in `dvd5.m` for details on the construction of the matrix $D(VD'_5)$. The rest of the code for both the convex splitting scheme and the Crank-Nicolson scheme can be implemented straightforwardly by calling a Newton's method subroutine at each time step; we define this subroutine in `newton.m`. The time-stepping code is defined in wrapping functions in `ac_convex_splitting.m` and `ac_crank_nicolson.m`. Note the dependence on `v.m` and `v_inv.m`, which wrap the `reshape` function in MATLAB for consistency with the notation used above.

4 Numerical Experiments

4.1 Qualitative evaluation

Qualitatively, solutions of the Allen-Cahn equation tend to approach either -1 or 1 as $t \rightarrow \infty$ depending on whether the initial value was negative or positive, with a smooth transition, whose typical scale is ε , between negative and positive regions. This behavior can be observed clearly in Figure 1, in which the solution starts nearly 0 everywhere, with some regions slightly positive (yellow) and some regions slightly negative (blue). As time passes, the positive parts approach 1 and the negative parts approach -1 , with a thin but smooth transition between regions.

4.2 Accuracy

The convex splitting scheme theoretically has first-order accuracy in time, and the Crank-Nicolson scheme theoretically has second-order accuracy in time. We verify these theoretical facts numerically by fixing a small spatial step size h and evaluating the error at time $t = 1$ of the two schemes for different temporal step sizes k .

To compute the error we generate a numerical reference solution (using the Crank-Nicolson scheme) using an even smaller spatial step size and a temporal step size much smaller than the smallest step size used in the experiment. As we see in Table 1, the convex splitting scheme empirically has first-order accuracy in time, which agrees with the theoretical result. As we see in Table 2, the Crank-Nicolson scheme empirically has second-order accuracy in time, which agrees with the theoretical result.

The tables were produced by running `test_accuracy.m`, and the raw outputs can be found in `outputs4.2.txt`.

k	L^∞ error	L^∞ rate	L^2 error	L^2 rate
1/8	7.602018e-03	-	2.391305e-02	-
1/16	4.005657e-03	0.924343	1.260021e-02	0.924351
1/32	2.058425e-03	0.960498	6.474971e-03	0.960502
1/64	1.043727e-03	0.979797	3.283138e-03	0.979798
1/128	5.255761e-04	0.989773	1.653247e-03	0.989773

Table 1: First-order convergence in time of the convex splitting scheme. Note that $h = \frac{2\pi}{64}$, and $\varepsilon = 0.01$; additionally, $k = \frac{1}{1024}$ for the reference solution generated using the Crank-Nicolson method.

4.3 Numerical maximum principle

Based on the physical interpretation of u as a relative density, it makes sense to consider initial data $u_0(x, y)$ such that $|u_0(x, y)| \leq 1$ for all x, y . In fact, it can be shown that if u_0 satisfies this condition, then the

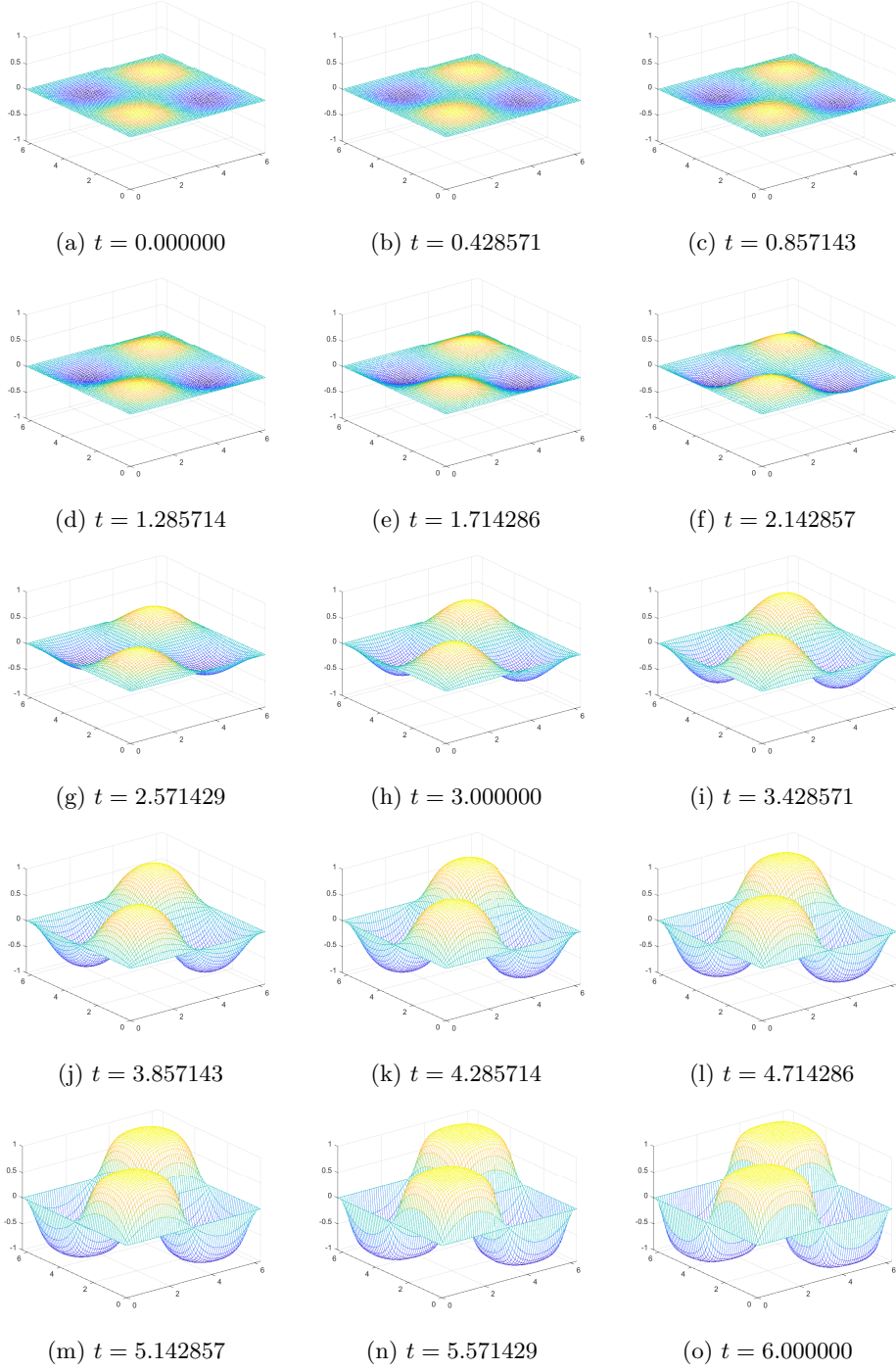


Figure 1: A typical solution of the Allen-Cahn equation.

k	L^∞ error	L^∞ rate	L^2 error	L^2 rate
1/8	1.564656e-04	-	5.200769e-04	-
1/16	3.903657e-05	2.002948	1.297311e-04	2.003201
1/32	9.747477e-06	2.001725	3.239257e-05	2.001790
1/64	2.429460e-06	2.004393	8.073391e-06	2.004416
1/128	6.002199e-07	2.017073	1.994561e-06	2.017104

Table 2: Second-order convergence in time of the Crank-Nicolson scheme. Note that $h = \frac{2\pi}{64}$, and $\varepsilon = 0.01$; additionally, $k = \frac{1}{1024}$ for the reference solution generated using the Crank-Nicolson method.

solution $u(x, y, t)$ satisfies $|u(x, y, t)| \leq 1$ for all $t \geq 0$ and all x, y . Thus, a good numerical scheme should be able to preserve this maximum/minimum principle of the Allen-Cahn equation.

According to Yang et al. [1] both the convex splitting and Crank-Nicolson schemes are capable of preserving this principle numerically in the sense that $|u_{i,j}^n| \leq 1$ for all i, j, n . The Crank-Nicolson scheme, however, only satisfies this property conditionally, with a sufficient condition being that $k \leq \min\{\frac{1}{2}, \frac{h^2}{2\varepsilon^2}\}$. In practice this requirement is not a huge issue; since ε can be interpreted as the typical length of the interface between the two materials, it makes sense to take $h \approx \varepsilon$, in which case the requirement on the Crank-Nicolson scheme effectively reduces to $k \leq \frac{1}{2}$.

Nevertheless, we can still observe the difference between the convex splitting scheme and the Crank-Nicolson scheme numerically by taking the time step very large. In Table 3, we see that the Crank-Nicolson scheme fails to satisfy the maximum condition numerically as the time step increases, but the convex splitting scheme always satisfies the maximum condition. We also see that the sufficient condition $k < \min\{\frac{1}{2}, \frac{h^2}{2\varepsilon^2}\}$ for the Crank-Nicolson scheme to preserve the maximum condition is not necessary, as the failure occurs when k is quite a bit larger than the sufficient value (note that we use different time intervals to ensure a fixed number of steps (100), so it may also be that the maximum condition would be violated after more time steps).

The table was produced by running `test_maximum.m`, and the raw outputs can be found in `outputs4.3.txt`.

k	$\sup_{n,i,j} u_{i,j}^n $	
	Convex Splitting	Crank-Nicolson
1/4	0.998772	0.998772
1/2	0.998772	0.998772
1	0.998772	0.998772
2	0.998773	1.057909
4	0.999466	1.049927
8	0.999508	1.062186

Table 3: The Crank-Nicolson method fails to satisfy the numerical maximum condition for large time steps, but the convex splitting method does not. Note that $h = \frac{2\pi}{64}$, and $\varepsilon = \frac{1}{4}$, which puts the sufficient condition value around 0.077.

5 Conclusion

In this project I solved the Allen-Cahn equation using the 5-point discretization of the Laplacian and two different time-stepping schemes: the convex splitting scheme and the Crank-Nicolson scheme. Using these implementations, I was able to compare the two time-stepping schemes in terms of accuracy and ability to preserve an important physical and mathematical property, a sort of minimum/maximum principle, of solutions of the Allen-Cahn equation.

I was able to observe the first-order accuracy in time of the convex splitting scheme and the second-order accuracy in time of the Crank-Nicolson scheme. I was also able to observe the unconditional preservation of the maximum/minimum condition for the convex splitting scheme and the conditional preservation for the Crank-Nicolson scheme. I also observed that the numerical solutions behaved qualitatively in a physically plausible manner.

6 Bonus

For fun, I created a video similar to the one linked earlier. You can find the video [here](#). For this I used the parameter settings $\varepsilon = 0.075$, $T = 40$, $h = \frac{2\pi}{256}$, and $k = 0.01$. The initial condition u_0 is white noise, so that $u_0(x, y) \sim \mathcal{N}(0, \sigma^2)$ for each $(x, y) \in [0, 2\pi]^2$, where the set of random variables $\{u_0(x^{(i)}, y^{(i)})\}_{i=1}^V$ are independent for any finite set of points $\{(x^{(i)}, y^{(i)})\}_{i=1}^V$. I chose $\sigma = 0.25$ (so that the value of $u_0(x, y)$ is very likely between -1 and 1). The code to generate the video can be found in `fun.m`.

The result is a solution that rapidly becomes smooth (due to the diffusion term), separating into randomly-shaped, smooth regions of positive and negative values which then approach 1 or -1 . The boundary between the two regions becomes smoother as time goes on.

References

- [1] Jiang Yang, Qiang Du, and Wei Zhang. Uniform Lp-Bound of the Allen–Cahn Equation and its Numerical Discretization. *International Journal of Numerical Analysis and Modeling*, 15(1):213–227, January 2018.