Mathematical Methods in the Applied Sciences

WILEY

RESEARCH ARTICLE

A Positivity-Preserving Numerical Method for the Ternary Allen–Cahn Equation

Soobin Kwak | Seokjun Ham | Junseok Kim

Department of Mathematics, Korea University, Seoul, Republic of Korea

Correspondence: Junseok Kim (cfdkim@korea.ac.kr)

Received: 14 May 2025 | **Revised:** 6 January 2026 | **Accepted:** 28 February 2026

Keywords: finite difference method | positivity-preserving | ternary Allen–Cahn equation

ABSTRACT

We propose a positivity-preserving computational method for the ternary Allen–Cahn (tAC) equation. To solve the tAC equation, we apply the operator splitting method. The tAC equation is decomposed into nonlinear and linear components. The nonlinear term is discretized using a finite difference method (FDM) combined with an explicit–implicit scheme. The resulting cubic polynomial admits a unique real solution under appropriate time step conditions. The unique solution is obtained by applying the analytical formula for solving cubic polynomials. The linear term, representing the diffusion equation, is discretized using a fully implicit FDM. The resulting tridiagonal matrix is solved by using the Thomas algorithm, which is a direct method for such systems. We analytically prove the positivity-preserving property of the proposed numerical method. To validate the performance of the proposed algorithm, several standard computational tests are conducted, and the results confirm the accuracy and effectiveness of the proposed algorithm.

1 | Introduction

We propose a positivity-preserving computational scheme for the ternary Allen–Cahn (tAC) equation [1]. The tAC equation plays an important role in modeling and simulating multi-phase systems involving three distinct components. The classical Allen–Cahn (AC) equation [2] characterizes phase separation and interface evolution in binary systems. In recent years, researchers have made significant progress in the numerical analysis of the AC equation, and they have refined both its theoretical foundations and its computational methodologies. Lee et al. [3] conducted a comprehensive review of numerical approaches for binary AC equations and provided a solid theoretical reference for the efficient computation of related models. Kang et al. [4] proposed an unconditionally stable numerical scheme tailored to higher-order AC equations, which markedly improves both the stability and efficiency of the solution process. Concurrently, the AC equation has demonstrated remarkable adaptability in multiphysics modeling and engineering applications. Lai et al. [5] incorporated the AC equation into a phase-field framework and developed a dual-energy-driven topology optimization method for multi-material systems, which enables efficient structural design across heterogeneous materials.

Abbreviations: AC, Allen–Cahn; FDM, finite difference method; OSM, operator splitting method; tAC, ternary Allen–Cahn.

For the binary AC equation, Baharlouei et al. [6] investigated the global existence and asymptotic behavior of strong solutions to the three-dimensional isentropic compressible Navier–Stokes/Allen–Cahn system under small, spatially decaying initial perturbations. Their analysis demonstrated that the phase variable exhibits diffusion-wave behavior with exponentially decaying amplitude, and that the fluid density and momentum follow a generalized Huygens principle. Xia et al. [7] employed the AC equation to construct a phase-field model that simulates fiber-scale thermal diffusion and phase transition in fused deposition modeling, successfully revealing the microstructural evolution governed by thermal-phase coupling. Lai et al. [8] integrated the AC equation with the phase-field approach to numerically investigate thermo-mechanical-phase interactions in laser melting-based additive manufacturing, which provides clear insights into the underlying multiphysics coupling mechanisms of the process. However, many practical applications involve more complex scenarios that require the analysis of systems with more than two phases.

The tAC equation extends the original framework by incorporating multiple order parameters and allows the representation of three-phase coexistence and interactions. A primary advantage of the tAC equation is its ability to capture complex interfacial dynamics among three different phases while preserving mass conservation through appropriate constraints. This makes it particularly useful in materials science, where the formation of microstructures in alloys, ceramics, and composites involves three-phase boundaries. Furthermore, it has been applied in biological systems to model multi-cellular interactions [9] and in fluid dynamics for describing emulsions and multiphase flows [10]. From a computational perspective, the tAC equation with mass conservation provides a relatively simple yet effective alternative to more complex models like the ternary Cahn–Hilliard equation [11]. Let us consider the following total energy functional $\mathcal{E}$:

$$\mathcal{E}(\mathbf{c}(\mathbf{x},t)) = \int_\Omega \sum_{k=1}^{3} \left[\frac{F\big(c_k(\mathbf{x},t)\big)}{\epsilon^2} + \frac{1}{2}|\nabla c_k(\mathbf{x},t)|^2 \right] d\mathbf{x}, \tag{1}$$

where $\mathbf{x} = (x, y)$, $F(c_k) = 0.25c_k^2(c_k - 1)^2$, and ϵ is an interfacial transition width related parameter. The L^2-gradient flow for Equation (1) is the following tAC equation: For $k = 1, 2, 3$,

$$\frac{\partial c_k(\mathbf{x},t)}{\partial t} = -\frac{F'(c_k(\mathbf{x},t))}{\epsilon^2} + \Delta c_k(\mathbf{x},t) + \beta(\mathbf{c}(\mathbf{x},t)), \text{ for } \mathbf{x} \in \Omega, \ t > 0, \tag{2}$$

where $\beta(\mathbf{c}(\mathbf{x},t)) = \sum_{k=1}^{3} F'(c_k(\mathbf{x},t))/(3\epsilon^2) = c_1(\mathbf{x},t)c_2(\mathbf{x},t)c_3(\mathbf{x},t)/\epsilon^2$ is the imposed Lagrange multiplier to enforce the constraint $c_1(\mathbf{x},t) + c_2(\mathbf{x},t) + c_3(\mathbf{x},t) = 1$ [12, 13]. Therefore, Equation (2) can be rewritten as follows:

$$\frac{\partial c_k(\mathbf{x},t)}{\partial t} = -\frac{F'(c_k(\mathbf{x},t))}{\epsilon^2} + \Delta c_k(\mathbf{x},t) + \frac{c_1(\mathbf{x},t)c_2(\mathbf{x},t)c_3(\mathbf{x},t)}{\epsilon^2}, \text{ for } \mathbf{x} \in \Omega, t > 0, \tag{3}$$

The zero Neumann boundary condition is used as

$$\mathbf{n} \cdot \nabla c_k(\mathbf{x},t) = 0, \quad \text{for} \quad \mathbf{x} \in \partial\Omega, \quad t \geq 0, k = 1, 2, 3,$$

where $\mathbf{n}$ denotes the normal vector.

In practical applications, most phase transition phenomena involve more than just two phases. This is especially true in the case of many alloys that are of significant technological importance, as they are often composed of multiple chemical components and exhibit intricate multi-phase structures that evolve over time. The vector-valued AC equation was proposed by Garcke et al. [14] as a generalization of the classical scalar AC model. The formulation was developed to effectively describe the dynamics of phase separation in multi-component systems, where each component is represented as part of a vector-valued order parameter. This approach enables the model to capture complex interactions and the evolution of interfaces among multiple distinct phases. Blank et al. [15] presented a primal-dual active set method for solving both the scalar and vector-valued AC equation with mass constraints, formulated as a PDE-constrained optimization problem. Sun et al. [16] proposed a space-time operator splitting method (OSM) for the high-dimensional vector-valued AC equations. The proposed method divide the PDEs into the heat equation and nonlinear ordinary differential equations. Krnhuber and Krause [17] presented multigrid methods for the vector-valued AC equations with isotropic interfacial energy and logarithmic potential. Baharlouei et al. [18] developed a stable numerical scheme based on backward Euler time discretization and the HDG method for space, and analyzed it for multi-component reaction–diffusion systems, with stability proofs, optimal convergence results, and validation through applications to the Brusselator and glycolysis models.

Li and Li [12] investigated the maximum bound principle for the vector-valued AC equations by using linear stabilization and exponential time differencing methods. The proposed schemes were proven to satisfy the discrete maximum bound principle and original energy dissipation, with numerical experiments validating their effectiveness and convergence. Xiao and Feng [19] proposed a second-order operator splitting finite element method that preserves the maximum bound principle for multi-phase AC systems. The proposed method is applied to the AC equation with both polynomial and Flory–Huggins potentials in two- and three-dimensional spaces. Weng et al. [20] developed a second-order accurate scalar auxiliary variable (SAV) computational method for the nonlocal ternary conservative AC system with both nonlocal space-independent and local-nonlocal space-time Lagrange multipliers. Zhai et al. [1] developed energy dissipation and maximum bound principle preserving method for a nonlocal tAC model. Weng et al. [21] investigated stability and error estimates of Strang splitting technique for the nonlocal ternary conservative AC model with a spatial convolution term and Lagrange multipliers ensuring mass conservation and hyperplane link. Zhang and Yang [13] presented unconditionally energy stable method for the AC type ternary phase-field equation with precise nonlocal volume conservation. The vector-valued AC system has been extensively applied in a wide range of fields, including immiscible ternary fluid dynamics [22], multi-phase image segmentation [23, 24], multi-surface reconstruction [25], and the modeling of triblock copolymers [26].

The main purpose of this study is to develop and validate a positivity-preserving computational scheme for efficiently solving the tAC equation by using an operator splitting approach that integrates an explicit–implicit finite difference method (FDM) for the nonlinear term and a fully implicit FDM for the linear term.

The remainder of this article is structured as follows. In Section 2, we describe the detailed numerical formulation of the proposed positivity-preserving scheme. In Section 3, we present computational tests that demonstrate the accuracy, effectiveness, and positivity-preserving property of the method. In Section 4, we present the conclusion and outline possible directions for future research. Additionally, the Appendix A provides the MATLAB code implementing the proposed scheme.

2 | Numerical Solutions

Let $\Omega = (L_x,\ R_x) \times (L_y,\ R_y)$ be the numerical domain and be discretized as $\Omega_h = \{(x_i, y_j) | x_i = L_x + (i - 0.5)h,\ 1 \le i \le N_x$ and $y_j = L_y + (j - 0.5)h,\ 1 \le j \le N_y\}$, where $h = (R_x - L_x)/N_x$. Let $c^n_{k,ij} = c_k\left(x_i, y_j, t_n\right)$ for $k = 1, 2, 3$, where $t_n = n\Delta t$. To numerically solve Equation (3), we apply the OSM [27–29]. In the OSM, intermediate variables are denoted using superscript asterisks: $c^*_{k,ij}$ represents the solution after the nonlinear step, $c^{**}_{k,ij}$ after the x-direction diffusion, and $c^{***}_{k,ij}$ after the y-direction diffusion. First, we solve the nonlinear equation:

$$\frac{\partial c_k(\mathbf{x}, t)}{\partial t} = -\frac{F'(c_k(\mathbf{x}, t))}{\epsilon^2} + \frac{c_1(\mathbf{x}, t)c_2(\mathbf{x}, t)c_3(\mathbf{x}, t)}{\epsilon^2}, \text{ for } \mathbf{x} \in \Omega, t > 0. \tag{4}$$

Equation (4) is discretized for each component $k = 1, 2, 3$ at every grid point, and this process yields a system of nonlinear algebraic equations that correspond to Equations (5, 6, 7) for the respective components:

$$\frac{c^*_{1,ij} - c^n_{1,ij}}{\Delta t} = -\frac{F'(c^*_{1,ij})}{\epsilon^2} + \frac{c^n_{2,ij}c^n_{3,ij}c^*_{1,ij}}{\epsilon^2}, \tag{5}$$

$$\frac{c^*_{2,ij} - c^n_{2,ij}}{\Delta t} = -\frac{F'(c^*_{2,ij})}{\epsilon^2} + \frac{c^n_{1,ij}c^n_{3,ij}c^*_{2,ij}}{\epsilon^2}, \tag{6}$$

$$\frac{c^*_{3,ij} - c^n_{3,ij}}{\Delta t} = -\frac{F'(c^*_{3,ij})}{\epsilon^2} + \frac{c^n_{1,ij}c^n_{2,ij}c^*_{3,ij}}{\epsilon^2}, \tag{7}$$

which can be rewritten as

$$(c^*_{1,ij})^3 - \frac{3}{2}(c^*_{1,ij})^2 + \left(\frac{1}{2} - c^n_{2,ij}c^n_{3,ij} + \frac{\epsilon^2}{\Delta t}\right)(c^*_{1,ij}) - \frac{\epsilon^2 c^n_{1,ij}}{\Delta t} = 0, \tag{8}$$

$$(c^*_{2,ij})^3 - \frac{3}{2}(c^*_{2,ij})^2 + \left(\frac{1}{2} - c^n_{1,ij}c^n_{3,ij} + \frac{\epsilon^2}{\Delta t}\right)(c^*_{2,ij}) - \frac{\epsilon^2 c^n_{2,ij}}{\Delta t} = 0, \tag{9}$$

$$(c^*_{3,ij})^3 - \frac{3}{2}(c^*_{3,ij})^2 + \left(\frac{1}{2} - c^n_{1,ij}c^n_{2,ij} + \frac{\epsilon^2}{\Delta t}\right)(c^*_{3,ij}) - \frac{\epsilon^2 c^n_{3,ij}}{\Delta t} = 0. \tag{10}$$

Let us consider Equation (8) and

$$g(c) = c^3 - \frac{3}{2}c^2 + \left(\frac{1}{2} - c_{2,ij}^n c_{3,ij}^n + \frac{\epsilon^2}{\Delta t}\right)c - \frac{\epsilon^2 c_{1,ij}^n}{\Delta t}. \tag{11}$$

Then, we want to find the solution to $g(c_{1,ij}^*) = 0$ for the given values of $c_{1,ij}^n$, $c_{2,ij}^n$, and $c_{3,ij}^n$. The solution is given by

$$c_{1,ij}^* = \frac{1}{2} + \sqrt[3]{-\frac{1}{2}q_{1,ij} + \sqrt{\frac{q_{1,ij}^2}{4} + \frac{p_{1,ij}^3}{27}}} + \sqrt[3]{-\frac{1}{2}q_{1,ij} - \sqrt{\frac{q_{1,ij}^2}{4} + \frac{p_{1,ij}^3}{27}}}, \tag{12}$$

$$c_{2,ij}^* = \frac{1}{2} + \sqrt[3]{-\frac{1}{2}q_{2,ij} + \sqrt{\frac{q_{2,ij}^2}{4} + \frac{p_{2,ij}^3}{27}}} + \sqrt[3]{-\frac{1}{2}q_{2,ij} - \sqrt{\frac{q_{2,ij}^2}{4} + \frac{p_{2,ij}^3}{27}}}, \tag{13}$$

$$c_{3,ij}^* = \frac{1}{2} + \sqrt[3]{-\frac{1}{2}q_{3,ij} + \sqrt{\frac{q_{3,ij}^2}{4} + \frac{p_{3,ij}^3}{27}}} + \sqrt[3]{-\frac{1}{2}q_{3,ij} - \sqrt{\frac{q_{3,ij}^2}{4} + \frac{p_{3,ij}^3}{27}}}, \tag{14}$$

where $p_{1,ij} = -0.25 - c_{2,ij}^n c_{3,ij}^n + \epsilon^2/\Delta t$, $p_{2,ij} = -0.25 - c_{3,ij}^n c_{1,ij}^n + \epsilon^2/\Delta t$, $p_{3,ij} = -0.25 - c_{1,ij}^n c_{2,ij}^n + \epsilon^2/\Delta t$, $q_{1,ij} = -0.5c_{2,ij}^n c_{3,ij}^n + 0.5\epsilon^2(1 - 2c_{1,ij}^n)/\Delta t$, $q_{2,ij} = -0.5c_{3,ij}^n c_{1,ij}^n + 0.5\epsilon^2(1 - 2c_{2,ij}^n)/\Delta t$, and $q_{3,ij} = -0.5c_{1,ij}^n c_{2,ij}^n + 0.5\epsilon^2(1 - 2c_{3,ij}^n)/\Delta t$. Let us take a derivative $g(c)$ with respect to c, then we obtain

$$\begin{aligned} g'(c) &= c^2 - 3c + \frac{1}{2} - c_{2,ij}^n c_{3,ij}^n + \frac{\epsilon^2}{\Delta t} \\ &= 3\left(c - \frac{1}{2}\right)^2 - \frac{1}{4} - c_{2,ij}^n c_{3,ij}^n + \frac{\epsilon^2}{\Delta t}. \end{aligned} \tag{15}$$

If $g(c)$ is an increasing function, then $g'(c) \geq 0$, which leads to

$$\Delta t \leq \frac{\epsilon^2}{0.25 + c_{2,ij}^n c_{3,ij}^n}, \tag{16}$$

which can be written, by using the condition $c_{1,ij}^n + c_{2,ij}^n + c_{3,ij}^n = 1$, so that the inequality $c_{2,ij}^n c_{3,ij}^n \leq 0.25$ can be directly applied as

$$\Delta t \leq 2\epsilon^2. \tag{17}$$

Under the condition (17), the equation $g(c) = 0$ has a unique positive solution $c_{1,ij}^* > 0$ if $c_{1,ij}^n > 0$, since $g(0) = -\epsilon^2 c_{1,ij}^n/\Delta t$ is negative. Next, we prove that $c_{1,ij}^* < 1$ under the assumption that $0 < c_{1,ij}^n,\ c_{2,ij}^n,\ c_{3,ij}^n < 1$. Because $c_{2,ij}^n c_{3,ij}^n \leq 0.25$ and the condition (17) gives $\epsilon^2/\Delta t \geq 1/2$, we obtain $g'(c) \geq 0$ for all c. Hence, g is strictly increasing. Since $g(0) = -\epsilon^2 c_{1,ij}^n/\Delta t < 0$,

$$g(1) = -c_{2,ij}^n c_{3,ij}^n + \frac{\epsilon^2(1 - c_{1,ij}^n)}{\Delta t} = -c_{2,ij}^n c_{3,ij}^n + \frac{\epsilon^2(c_{2,ij}^n + c_{3,ij}^n)}{\Delta t}. \tag{18}$$

From $\epsilon^2/\Delta t \geq 0.5$, it follow that

$$g(1) \geq -c_{2,ij}^n c_{3,ij}^n + \frac{(c_{2,ij}^n + c_{3,ij}^n)}{2}. \tag{19}$$

For $c_{2,ij}^n,\ c_{3,ij}^n \in (0, 1)$, the arithmetic–geometric mean inequality gives

$$\frac{(c_{2,ij}^n + c_{3,ij}^n)}{2} \geq \sqrt{c_{2,ij}^n c_{3,ij}^n} \geq c_{2,ij}^n c_{3,ij}^n, \tag{20}$$

which leads to $(c_{2,ij}^n + c_{3,ij}^n)/2 - c_{2,ij}^n c_{3,ij}^n > 0$ and thus $g(1) > 0$. Because $g(0) < 0 < g(1)$ and g is strictly increasing, the unique root satisfies

$$0 < c_{1,ij}^* < 1. \tag{21}$$

The same argument applies to $c^*_{2,ij}$ and $c^*_{3,ij}$ by cyclic permutation of the indices. Therefore, under the assumptions $0 < c^n_{1,ij}, c^n_{1,ij}, c^n_{1,ij} < 1$ and $\Delta t \le 2\epsilon^2$, the nonlinear update satisfies $0 < c^*_{k,ij} < 1$ for $k = 1, 2, 3$. In the specific cases, for example, when $c^n_{1,ij} = 1$, we have $c^n_{2,ij} = c_{3,ij} = 0$, and the Equation (11) reduces to

$$g(c) = c^3 - \frac{3}{2}c^2 + \left(\frac{1}{2} + \frac{\epsilon^2}{\Delta t}\right)c - \frac{\epsilon^2}{\Delta t}$$
$$= (c-1)\left(c^2 - \frac{1}{2}c + \frac{\epsilon^2}{\Delta t}\right).$$

Since $\epsilon^2/\Delta t \ge 0.5$, the only real root satisfying $g(c^*_{1,ij}) = 0$ in this case is $c^*_{1,ij} = 1$. Similarly, in the specific case $c^n_{1,ij} = 0$, we obtain

$$g(c) = c^3 - \frac{3}{2}c^2 + \left(\frac{1}{2} - c^n_{2,ij}c^n_{3,ij} + \frac{\epsilon^2}{\Delta t}\right)c$$
$$= c\left(c^2 - \frac{3}{2}c + \left(\frac{1}{2} - c^n_{2,ij}c^n_{3,ij} + \frac{\epsilon^2}{\Delta t}\right)\right).$$

The conditions $\epsilon^2/\Delta t \ge 0.5$ and $c^n_{2,ij}c^n_{3,ij} \le 0.25$ imply the existence of a real solution of $g(c^*_{1,ij}) = 0$ in $[0, 1]$, and this solution is $c^*_{1,ij} = 0$. Therefore, under the assumptions

$$0 \le c^n_{k,ij} \le 1, \quad c^n_{1,ij} + c^n_{1,ij} + c^n_{1,ij} = 1, \quad \Delta t \le 2\epsilon^2,$$

the existence of a real value $c^*_{k,ij}$ in $[0, 1]$ follows for every component. In the two-phase case, which corresponds to the condition $c^n_{3,ij} = 0$ for all grid points, the same argument implies $c^*_{k,ij} \in [0, 1]$. Thus, the proposed scheme preserves positivity in both the ternary and two-phase settings.

Finally, note that $g(1) \ge 0$ holds whenever $\epsilon^2/\Delta t \ge c^n_{2,ij}c^n_{3,ij}/(c^n_{2,ij} + c^n_{3,ij})$, and monotonicity $g'(c) \ge 0$ requires $\epsilon^2/\Delta t \ge 0.25 + c^n_{2,ij}c^n_{3,ij}$. The time-step constraint (17) guarantees $\epsilon^2/\Delta t \ge 0.5$, which satisfies both conditions simultaneously. The time step constraint given in Equation (17) allows for a time step size approximately ten times larger than that of a fully explicit method [30].

Second, we solve the diffusion equation:

$$\frac{\partial c_k(\mathbf{x}, t)}{\partial t} = \Delta c_k(\mathbf{x}, t), \text{ for } \mathbf{x} \in \Omega, \ t > 0, \tag{22}$$

for each component $k = 1, 2, 3$. Equation (22) can be written as

$$\frac{\partial c_k(\mathbf{x}, t)}{\partial t} = \frac{\partial^2 c_k(\mathbf{x}, t)}{\partial x^2} + \frac{\partial^2 c_k(\mathbf{x}, t)}{\partial y^2}. \tag{23}$$

To numerically solve Equation (23), we apply the OSM, which decomposes the equation into two one-dimensional parts as follows:

$$\frac{c^{**}_{k,ij} - c^*_{k,ij}}{\Delta t} = \frac{c^{**}_{k,i+1,j} - 2c^{**}_{k,ij} + c^{**}_{k,i-1,j}}{h^2}, \tag{24}$$
$$\frac{c^{***}_{k,ij} - c^{**}_{k,ij}}{\Delta t} = \frac{c^{***}_{k,i,j+1} - 2c^{***}_{k,ij} + c^{***}_{k,i,j-1}}{h^2}. \tag{25}$$

Equations (24) and (25) are solved using the Thomas algorithm with homogeneous Neumann boundary conditions [31], implemented as follows:

$$c^{**}_{k,0j} = c^{**}_{k,1j}, \ c^{**}_{k,N_x+1,j} = c^{**}_{k,N_x j}, \ c^{***}_{k,i0} = c^{***}_{k,i1}, \text{ and } c^{***}_{k,i,N_y+1} = c^{***}_{k,iN_y}.$$

The diffusion substep (Equations (24) and (25)) solves tridiagonal linear systems. For a fixed j, Equation (24) in the x-direction case can be written as $A\mathbf{c}^{**}_{k,j} = \mathbf{c}^{*}_{k,j}$, where

$$A = \begin{pmatrix} 1+\frac{\Delta t}{h^2} & -\frac{\Delta t}{h^2} & 0 & \cdots & 0 \\ -\frac{\Delta t}{h^2} & 1+2\frac{\Delta t}{h^2} & -\frac{\Delta t}{h^2} & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & -\frac{\Delta t}{h^2} & 1+2\frac{\Delta t}{h^2} & -\frac{\Delta t}{h^2} \\ 0 & \cdots & 0 & -\frac{\Delta t}{h^2} & 1+\frac{\Delta t}{h^2} \end{pmatrix}, \quad \mathbf{c}^{**}_{k,j} = \begin{pmatrix} c^{**}_{k,1j} \\ c^{**}_{k,2j} \\ \vdots \\ c^{**}_{k,N_x-1,j} \\ c^{**}_{k,N_xj} \end{pmatrix}, \quad \text{and} \quad \mathbf{c}^{*}_{k,j} = \begin{pmatrix} c^{*}_{k,1j} \\ c^{*}_{k,2j} \\ \vdots \\ c^{*}_{k,N_x-1,j} \\ c^{*}_{k,N_xj} \end{pmatrix}.$$

The matrix A is an M-matrix that is strictly diagonally dominant under homogeneous Neumann boundary conditions. Therefore, A is invertible and its inverse A^{-1} is nonnegative.

Assume that $\mathbf{c}^{*}_{k,j}$ satisfies $0 \le c^{*}_{k,ij} \le 1$ for all grid indices i and j.Since $A\mathbf{1} = \mathbf{1}$, it follows that $A^{-1}\mathbf{1} = \mathbf{1}$, where $\mathbf{1} = (1, 1, \dots, 1)^T$. Given that $\mathbf{0} \le \mathbf{c}^{*}_{k,j} \le \mathbf{1}$, $A^{-1} \ge 0$, and $A^{-1}\mathbf{1} = \mathbf{1}$, we obtain

$$\mathbf{0} \le A^{-1}\mathbf{c}^{*}_{k,j} \le \mathbf{1}. \tag{26}$$

Therefore,

$$0 \le c^{**}_{k,ij} \le 1, \tag{27}$$

for all i and j. A similar argument can be applied to the diffusion step in the y-direction, which guarantees the same bound $0 \le c^{***}_{k,ij} \le 1$.

These results confirm that the proposed scheme is a positivity-preserving scheme. In general, the solutions of Equations (8, 9, 10) do not satisfy the constraint $c^{n+1}_{1,ij} + c^{n+1}_{2,ij} + c^{n+1}_{3,ij} = 1$. Therefore, a projection step is applied to enforce this condition and redefine the concentrations as follows:

$$c^{n+1}_{1,ij} = \frac{c^{***}_{1,ij}}{c^{***}_{1,ij} + c^{***}_{2,ij} + c^{***}_{3,ij}}, \quad c^{n+1}_{2,ij} = \frac{c^{***}_{2,ij}}{c^{***}_{1,ij} + c^{***}_{2,ij} + c^{***}_{3,ij}}, \text{ and } c^{n+1}_{3,ij} = \frac{c^{***}_{3,ij}}{c^{***}_{1,ij} + c^{***}_{2,ij} + c^{***}_{3,ij}}. \tag{28}$$

To study the stability of the computational scheme, we define the discrete total energy as

$$\mathcal{E}_d(\mathbf{c}^n) = \frac{1}{N_xN_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{3}\left(\frac{F(c^n_{k,ij})}{\epsilon^2} + \frac{1}{2}\left[\left(\frac{c^n_{k,i+1,j} - c^n_{k,i-1,j}}{2h}\right)^2 + \left(\frac{c^n_{k,i,j+1} - c^n_{k,i,j-1}}{2h}\right)^2\right]\right).$$

3 | Computational Tests

Before the overall numerical tests, a convergence test of the proposed method is first performed to confirm its temporal and spatial accuracy. The test is conducted on the domain $\Omega = (-1, 1) \times (-1, 1)$. The initial condition is set as

$$c_1(x, y, 0) = \frac{1}{3} + 0.1\cos(\pi x)\cos(\pi y), \; c_2(x, y, 0) = \frac{1}{3} + 0.1\sin(\pi x)\sin(\pi y), \text{ and } c_3(x, y, 0) = 1 - c_1(x, y, 0) - c_2(x, y, 0),$$

which guarantees that $c_1 + c_2 + c_3 = 1$ at all grid points. To confirm the time-accuracy, the grid size is fixed at $h = 0.01$ and $\epsilon = 0.02$, and time step is successively reduced as $\Delta t = \{h^2, 0.5h^2, 0.25h^2, 0.125h^2\}$. A highly resolved solution computed with the smallest time step, $\Delta t_{\text{ref}} = h^2/2^{10}$, is regarded as the reference solution. The discrete l^2-norm of the error for each component is defined as

$$\|e_k\|_2 = \sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}|c_{k,ij} - c^{\text{ref}}_{k,ij}|^2}, \quad k = 1, 2, 3,$$

TABLE 1 | Temporal convergence of proposed scheme with fixed grid step size $h = 0.01$.

Δt		4e−4		2e−4		1e−4		5e−5
c_1	Error	4.51812e−2		2.17747e−2		1.06373e−2		5.23784e−3
	Rate		1.05		1.03		1.02	
c_2	Error	4.51812e−2		2.17747e−2		1.06373e−2		5.23784e−3
	Rate		1.05		1.03		1.02	
c_3	Error	4.25851e−2		2.08992e−2		1.03143e−2		5.10614e−3
	Rate		1.03		1.02		1.01	

TABLE 2 | Spatial convergence of the proposed scheme with time step $\Delta t = 0.1 \times 2^{-10}$.

h		0.05		0.025		0.0125		0.00625
c_1	Error	9.91703e−3		2.12251e−3		5.15380e−4		1.31703e−4
	Rate		2.22		2.04		1.97	
c_2	Error	9.91703e−3		2.12251e−3		5.15380e−4		1.31703e−4
	Rate		2.22		2.04		1.97	
c_3	Error	1.33917e−2		2.81811e−3		6.73677e−4		1.60953e−4
	Rate		2.25		2.06		2.07	

where $c_{k,ij}^{\text{ref}}$ denotes the reference solution. The convergence errors and rates are summarized in Table 1. As shown, the errors decrease approximately by a factor of two when ever the time step is halved, which confirms first-order accuracy in time. The first-order temporal convergence observed in Table 1 is consistent with the OSM used in this study. The proposed scheme applies a sequential (Lie-type) operator splitting, where the nonlinear reaction step and the diffusion step are solved one after the other within each time step. Even though each subproblem is solved exactly or by an unconditionally stable implicit method, the overall temporal accuracy is limited by the splitting error, which is first order in time. As a result, the observed convergence rates reflect the basic accuracy of the OSM, rather than limitations of the spatial discretization or the nonlinear solver.

For the spatial convergence test, the time step is fixed $\Delta t = 0.1 \times 2^{-10}$ and $\epsilon = 0.02$, and the grid step size is refined as $h = \{0.05, 0.025, 0.0125, 0.00625\}$. The numerical solution obtained on the finest grid, $h = 0.003125$, is taken as the reference solution. The results summarized in Table 2 show that the error decreases approximately by a factor of four as h is halved, which verifies second-order accuracy in space.

To verify the capability of the tAC model in capturing the dynamics of multiple phase interfaces and triple junctions, we consider an initial configuration composed of three well-separated circular regions representing phases c_1, c_2, and c_3. The initial conditions are defined as follows:

$$c_1(x, y, 0) = \frac{1}{3}\left(1 + u\left(0, \frac{2}{3}\right) - \frac{1}{2}u\left(\frac{1}{\sqrt{3}}, -\frac{1}{3}\right) - \frac{1}{2}u\left(-\frac{1}{\sqrt{3}}, -\frac{1}{3}\right)\right),$$

$$c_2(x, y, 0) = \frac{1}{3}\left(1 + u\left(\frac{1}{\sqrt{3}}, -\frac{1}{3}\right) - \frac{1}{2}u\left(0, \frac{2}{3}\right) - \frac{1}{2}u\left(-\frac{1}{\sqrt{3}}, -\frac{1}{3}\right)\right),$$

$$c_3(x, y, 0) = 1 - \sum_{k=1}^{2} c_k(x, y, 0),$$

where $u(p, q) = \tanh((0.2 - \sqrt{(x-p)^2 + (y-q)^2})/(2\sqrt{2}\epsilon))$. We use the following parameters in the simulation: The computational domain is set to $\Omega = (-1, 1) \times (-1, 1)$ and discretized with 100×100 uniform grid points. The interface thickness is chosen as $\epsilon = h$, where h is the grid size. The time step is taken as $\Delta t = h^2$, and the simulation is carried out up to final time $T = 0.024$. Figure 1 shows the evolution of the phase domains at four representative times. Initially, three

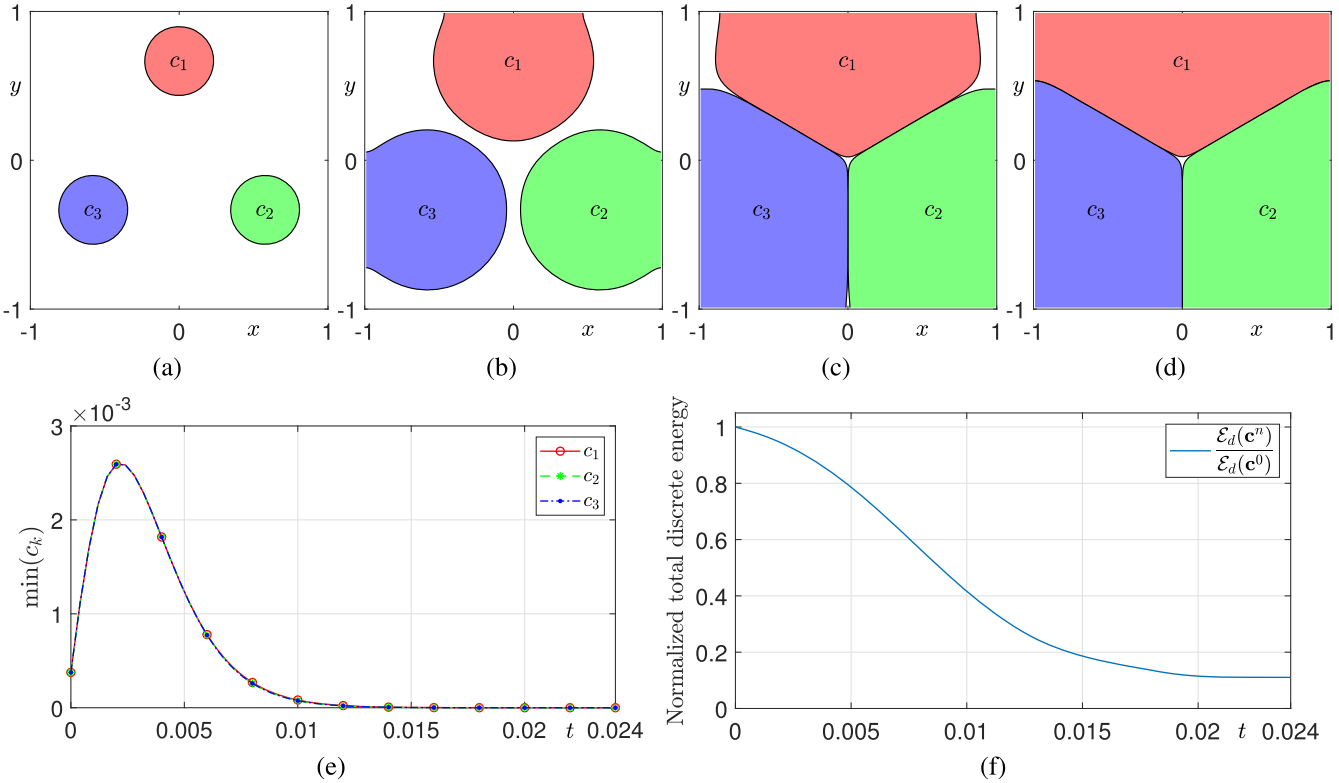

FIGURE 1 | Snapshots of the tAC evolution at (a) $t = 0$, (b) $t = 20\Delta t$, (c) $t = 40\Delta t$, and (d) $t = 60\Delta t$. Temporal evolutions of (e) the minimum values of c_k and (f) the normalized discrete total energy.

nearly circular domains are well separated, as shown in Figure 1a. As time progresses, the interfaces expand and interact, eventually forming a triple junction where all three phases meet symmetrically (Figure 1c and d). The simulation confirms the capture curvature-driven interface motion and stable triple junction formation.

To validate the proposed computational method, the temporal evolution of the triple junction is investigated on the domain $\Omega = (0, 1) \times (0, 1)$, based on the following initial condition. The initial condition consists of three phases, c_1, c_2, and c_3 arranged in periodic square blocks of width 0.1. The phases are assigned according to a modular pattern in the x-direction with a row-dependent phase shift, which generates a diagonally staggered triplet tiling pattern. The mesh size used is 200×200 with $\epsilon = 0.8h$ and $\Delta t = h^2$. Figure 2 illustrates the temporal evolution of the triple junction structure. Over time, the contact angle approaches 120°, which is consistent with the theoretical equilibrium configuration under equal interfacial tensions [32]. This result provides evidence that the proposed numerical method can accurately capture the dynamics of the ternary AC equation.

We consider a circular droplet as a liquid lens that is surrounded by two other phases. The initial condition is defined on the $\Omega = (-1, 1) \times (-1, 1)$ as follows:

$$c_1(x, y, 0) = \frac{1}{2}\left(1 + \tanh\left(\frac{0.5 - \sqrt{x^2 + y^2}}{2\sqrt{2}\epsilon}\right)\right),$$

$$c_2(x, y, 0) = \frac{1}{2}\left(1 + \tanh\left(\frac{y}{2\sqrt{2}\epsilon}\right)\right)\big(1 - c_1(x, y, 0)\big),$$

$$c_3(x, y, 0) = 1 - \sum_{k=1}^{2} c_k(x, y, 0).$$

The other parameters are used as follows: $h = 0.01$, $\epsilon = h$, and $\Delta t = h^2$. As shown in Figure 3, the initially circular droplet of phase c_1 is deformed through time evolution, and the contact angle converges to 120°.

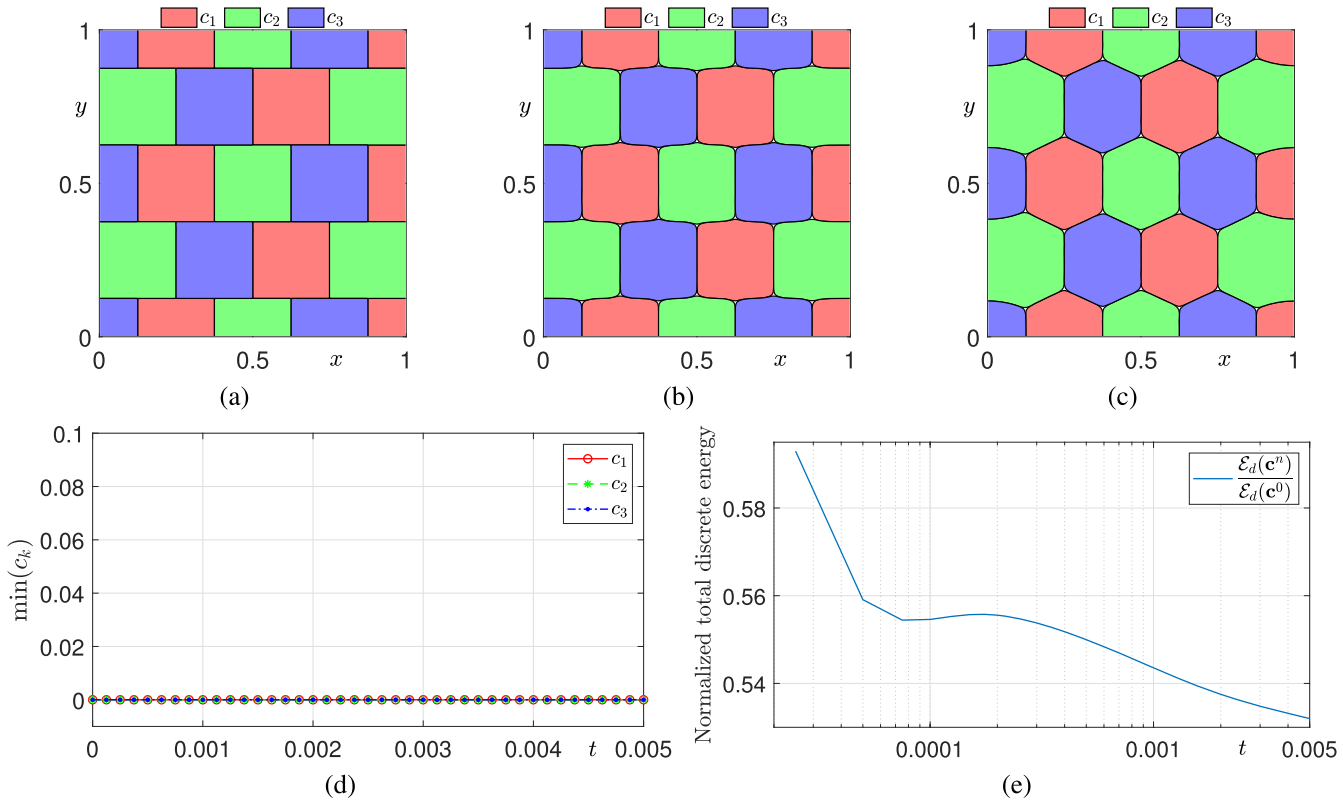

FIGURE 2 | Temporal evolution of the triple junction structure. (a) Initial conditions, (b) $t = 20\Delta t$, and (c) $t = 200\Delta t$. (d) Temporal evolution of the minimum values of c_k. (e) Temporal evolution of the normalized discrete total energy.

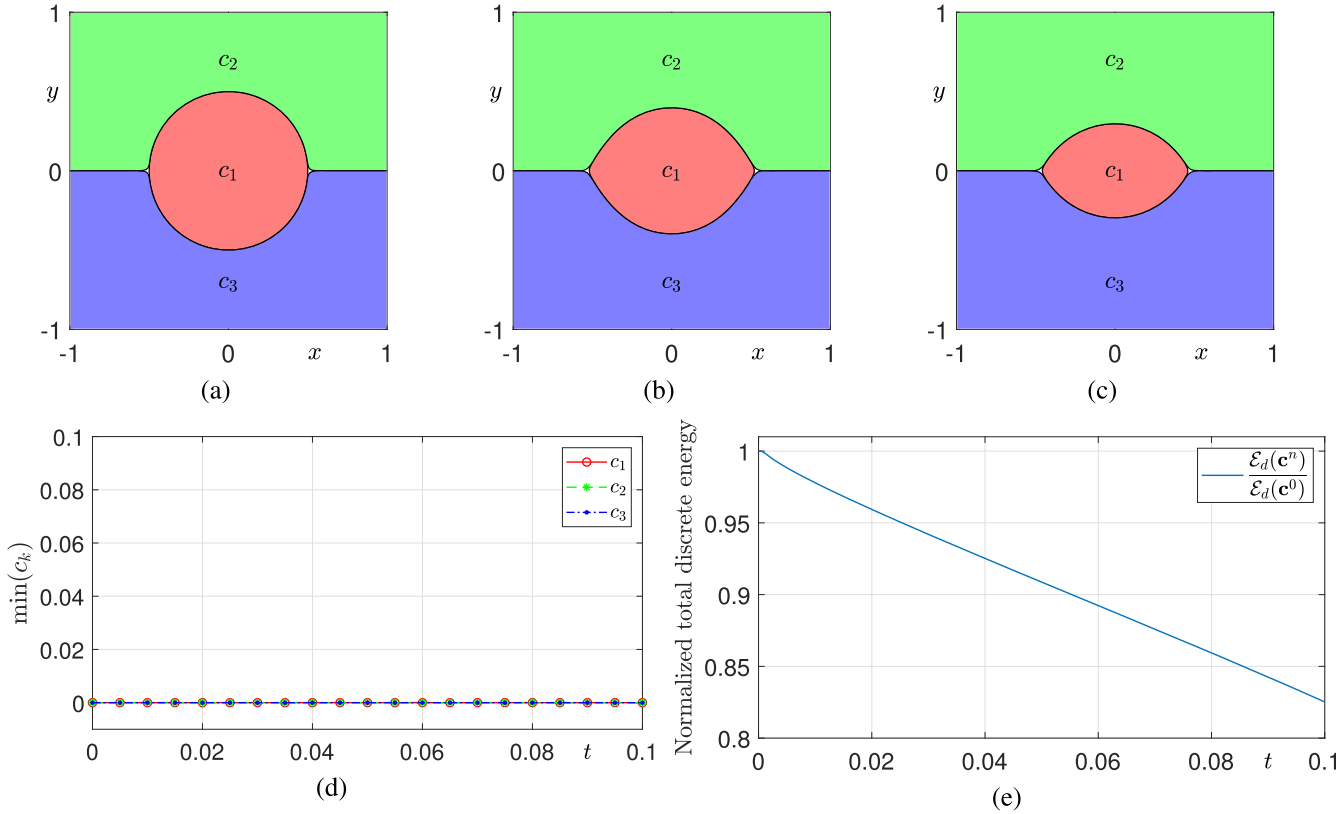

FIGURE 3 | Evolution of a circular droplet of phase c_1 in contact with phases c_2 and c_3. (a) Initial conditions, (b) $t = 500\Delta t$, and (c) $t = 1000\Delta t$. Temporal evolutions of (d) the minimum values of c_k and (e) the normalized discrete total energy.

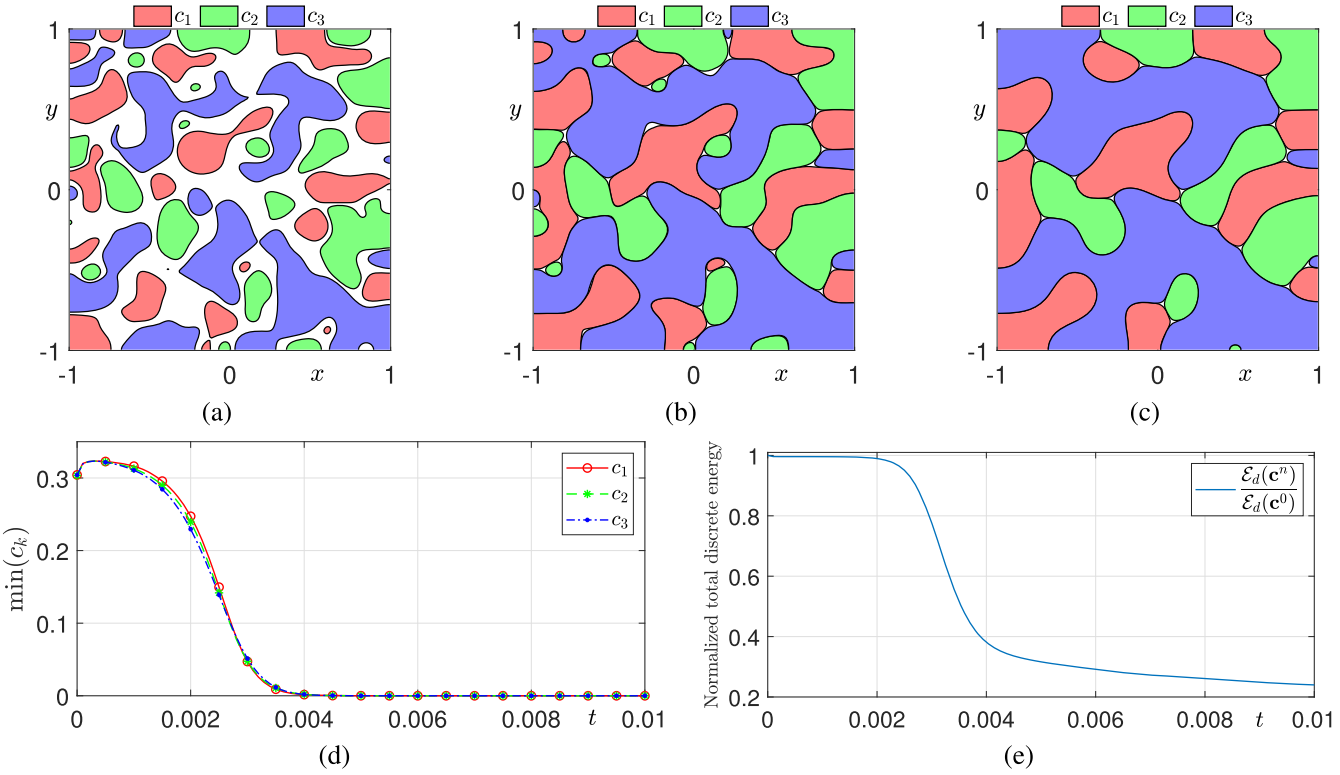

FIGURE 4 | Phase separation dynamics with random initial condition. (a) $t = 30\Delta t$, (b) $t = 50\Delta t$, and (c) $t = 100\Delta t$. (d) Time evolution of the minimum values of c_k for $k = 1,\ 2$, and 3. (e) The normalized discrete total energy.

To validate the positivity-preserving property of the proposed numerical scheme in practical simulations, a numerical experiment is conducted using a random initial condition. The initial condition is defined as follows:

$$c_1(x, y, 0) = \frac{1}{3} + 0.05\text{rand}(x, y),\ c_2(x, y, 0) = \frac{1}{3} + 0.05\text{rand}(x, y),\ \text{and } c_3(x, y, 0) = 1 - \sum_{k=1}^{2} c_k(x, y, 0),$$

where $(x, y) \in \Omega = (-1, 1) \times (-1, 1)$, and $\text{rand}(x, y)$ denotes a random value between 1 and -1 drawn from a truncated normal distribution. The other parameters are set to be the same as those used in the previous computational experiment. Figure 4 demonstrates the phase separation dynamics starting from a random initial condition to examine the positivity-preserving property of the proposed numerical algorithm. In Figure 4a–c, the three phases c_1, c_2, and c_3 gradually separate and coarsen. In Figure 4d presents the time evolution of the minimum values of c_k for $k = 1, 2$, and 3. The results show that all minimum values remain non-negative throughout the simulation, which strongly supports the claim that the proposed scheme preserves positivity in practice.

Furthermore, we investigate the effect of using time steps lager than the theoretical constraint. When Δt exceeds the theoretical bound $\Delta t \le 2\epsilon^2$, the cubic discriminant in the nonlinear substep,

$$D_{k,ij} = \frac{1}{4} q_{k,ij}^2 + \frac{1}{27} p_{k,ij}^3,$$

becomes negative at certain grid points. In particular, when $\Delta t = 3\epsilon^2$ is used, we observed that $D_{k,ij} < 0$ appears at several grid points already in the first iteration. In this regime, the analytic solution formulas (12) and (13) no longer provide real-valued updates, and the positivity-preserving scheme becomes ill-defined. In our numerical implementation, this results in an immediate breakdown of the solver due to complex-valued updates. These observations clearly demonstrate that the proposed time-step restriction is practically sharp and essential for maintaining the correctness of the algorithm.

Finally, we consider the following initial condition on the domain $\Omega = (-1, 1) \times (-1, 1)$:

$$c_1(x, y, 0) = 0.5 + 0.5 \tanh\left(\frac{\sqrt{x^2 + y^2} - 0.5}{2\sqrt{2}\epsilon} \right),$$

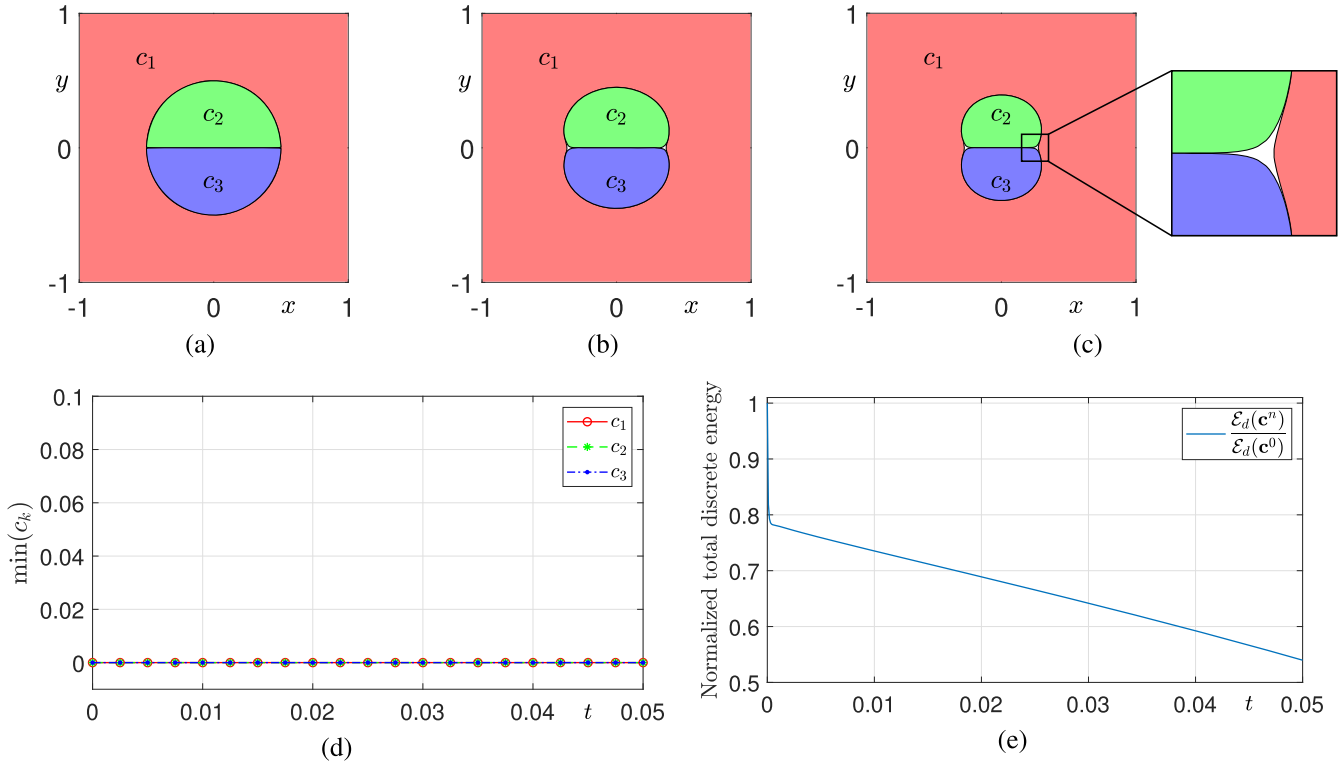

FIGURE 5 | The phase evolution of c_1, c_2, and c_3 at (a) $t = 0$, (b) $t = 250\Delta t$, and (c) $t = 500\Delta t$. Temporal evolutions of (d) the minimum values of c_k and (e) the normalized discrete total energy.

$$c_2(x, y, 0) = \begin{cases} 1 - c_1(x, y, 0) & \text{if } y \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

$$c_3(x, y, 0) = \begin{cases} 1 - c_1(x, y, 0) & \text{if } y < 0 \\ 0 & \text{otherwise} \end{cases}.$$

The other parameters are set as $h = 0.01$, $\Delta t = h^2$, and $\epsilon = h$. Figure 5 presents snapshots of the temporal phase evolution. As time progresses, the evolution at the triple junction shows that each phase satisfies the properties of the AC equation, and the contact angle converges to 120°.

The positivity-preserving property of the proposed numerical algorithm was confirmed through several numerical experiments.

4 | Conclusions

In this study, we developed a positivity-preserving computational scheme for the tAC equation. The proposed method decomposes the tAC equation into nonlinear and diffusion terms using OSM. The resulting nonlinear algebraic system admits a unique real solution, which is obtained analytically using the cubic formula under an appropriate time step restriction. We proved that the method satisfies a sufficient condition for positivity preservation in the discrete setting and implemented a normalization step to ensure that the constraint $c_1 + c_2 + c_3 = 1$ is maintained at each time step. The various numerical experiments were conducted to validate the performance of the proposed scheme. The numerical results demonstrate that the method captures the evolution of interfaces and the formation of triple junctions at the expected 120° contact angle, even under complex initial conditions such as random noise. Moreover, the scheme consistently preserved the non-negativity of phase variables, and this result confirmed its positivity-preserving property. The proposed method provides a computationally efficient and reliable approach for simulating multi-phase dynamics described by the tAC equation. As a natural extension, we plan to apply the developed scheme to the conservative ternary AC system in order to investigate mass-conserving multi-phase dynamics [33].

Author Contributions

Soobin Kwak: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review and editing. **Seokjun Ham:** formal analysis, investigation, validation, writing – original draft. **Junseok Kim:** conceptualization, formal analysis, project administration, supervision, validation, writing – original draft, writing – review and editing.

Acknowledgments

The authors would like to express their sincere gratitude to the reviewers for their valuable comments, which greatly improved the quality of this paper.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (Grant No. 2022R1A2C1003844).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

No data was used for the research described in the article.

References

1. S. Zhai, Z. Weng, Y. Mo, and X. Feng, "Energy Dissipation and Maximum Bound Principle Preserving Scheme for Solving a Nonlocal Ternary Allen–Cahn Model," *Computers & Mathematics with Applications* 155 (2024): 150–164, https://doi.org/10.1016/j.camwa.2023.12.006.

2. S. M. Allen and J. W. Cahn, "A Microscopic Theory for Antiphase Boundary Motion and Its Application to Antiphase Domain Coarsening," *Acta Metallurgica* 27, no. 6 (1979): 1085–1095, https://doi.org/10.1016/0001-6160(79)90196-2.

3. H. G. Lee, Y. Li, J. Yang, et al., "A Review of the Numerical Methods for Solving the Binary Allen–Cahn Equation," *Physica A* 643 (2025): 130625, https://doi.org/10.1016/j.physa.2025.130625.

4. S. Kang, Y. Hwang, and J. Kim, "Unconditionally Stable Method for the High-Order Allen–Cahn Equation," *Journal of Computational Science* 76 (2025): 102636, https://doi.org/10.1016/j.jocs.2025.102636.

5. S. Lai, J. Feng, Z. Lv, J. Kim, and Y. Li, "A Dual-Energy Physics-Informed Multi-Material Topology Optimization Method Within the Phase-Field Framework," *Computer Methods in Applied Mechanics and Engineering* 447 (2025): 118338, https://doi.org/10.1016/j.cma.2025.118338.

6. Y. Chen, H. Tang, and Y. Zhang, "Wave Propagation for the Isentropic Compressible Navier–Stokes/Allen–Cahn System," *Mathematical Methods in the Applied Sciences* 48 (2025): 2726–2742, https://doi.org/10.1002/mma.10458.

7. B. Xia, S. Lai, Q. Xia, X. Liu, Y. Li, and J. Kim, "Phase Field Modeling of Fiber-Based Thermal Diffusion and Phase Transitions in the Fused Deposition Modeling Process," *Communications in Nonlinear Science and Numerical Simulation* 151 (2025): 109071, https://doi.org/10.1016/j.cnsns.2024.109071.

8. S. Lai, Q. Xia, J. Kim, and Y. Li, "Numerical Investigation of Thermo-Mechanical-Phase Interaction in Laser Melting Additive Manufacturing Process Incorporating Phase-Field Framework," *Physics of Fluids* 37 (2025): 087164, https://doi.org/10.1063/5.0282033.

9. J. Jiang, K. Garikipati, and S. Rudraraju, "A Diffuse Interface Framework for Modeling the Evolution of Multi-Cell Aggregates as a Soft Packing Problem Driven by the Growth and Division of Cells," *Bulletin of Mathematical Biology* 81 (2019): 3282–3300, https://doi.org/10.1007/s11538-019-00577-1.

10. S. Aihara, N. Takada, and T. Takaki, "Highly Conservative Allen–Cahn-Type Multi-Phase-Field Model and Evaluation of Its Accuracy," *Theoretical and Computational Fluid Dynamics* 37, no. 5 (2023): 639–659, https://doi.org/10.1007/s00162-023-00655-0.

11. L. Liu, S. Huang, X. Xiao, and X. Feng, "Mathematical Modeling and Numerical Simulation of the N-Component Cahn–Hilliard Model on Evolving Surfaces," *Journal of Computational Physics* 513 (2024): 113189, https://doi.org/10.1016/j.jcp.2024.113189.

12. J. Li and J. Li, "Unconditional MBP Preservation and Energy Stability of the Stabilized Exponential Time Differencing Schemes for the Vector-Valued Allen–Cahn Equations," *Communications in Nonlinear Science and Numerical Simulation* 139 (2024): 108271, https://doi.org/10.1016/j.cnsns.2024.108271.

13. J. Zhang and X. Yang, "Unconditionally Energy Stable Large Time Stepping Method for the L2-Gradient Flow Based Ternary Phase-Field Model With Precise Nonlocal Volume Conservation," *Computer Methods in Applied Mechanics and Engineering* 361 (2020): 112743, https://doi.org/10.1016/j.cma.2019.112743.

14. H. Garcke, B. Nestler, and B. Stoth, "On Anisotropic Order Parameter Models for Multi-Phase Systems and Their Sharp Interface Limits," *Physica D* 115, no. 1–2 (1998): 87–108, https://doi.org/10.1016/S0167-2789(97)00227-3.

15. L. Blank, L. Sarbu, and M. Stoll, "Preconditioning for Allen–Cahn Variational Inequalities With Non-Local Constraints," *Journal of Computational Physics* 231, no. 16 (2012): 5406–5420, https://doi.org/10.1016/j.jcp.2012.04.035.

16. Y. Sun, X. Xiao, Z. Gao, and X. Feng, "An Efficient Space-Time Operator-Splitting Method for High-Dimensional Vector-Valued Allen–Cahn Equations," *International Journal of Numerical Methods for Heat & Fluid Flow* 29, no. 9 (2019): 3437–3453, https://doi.org/10.1108/HFF-01-2019-0076.

17. R. Kornhuber and R. Krause, "Robust Multigrid Methods for Vector-Valued Allen–Cahn Equations With Logarithmic Free Energy," *Computing and Visualization in Science* 9, no. 2 (2006): 103–116, https://doi.org/10.1007/s00791-006-0020-2.

18. S. Baharlouei, R. Mokhtari, and N. Chegini, "A Stable Hybridized Discontinuous Galerkin Method for Solving Some Nonlinear m-Component Reaction-Diffusion Systems," *Mathematical Methods in the Applied Sciences* 48 (2025): 12405–12420, https://doi.org/10.1002/mma.11034.

19. X. Xiao and X. Feng, "A Second-Order Maximum Bound Principle Preserving Operator Splitting Method for the Allen–Cahn Equation With Applications in Multi-Phase Systems," *Mathematics and Computers in Simulation* 202 (2022): 36–58, https://doi.org/10.1016/j.matcom.2022.05.024.

20. Z. Weng, X. Yue, and S. Zhai, "A Second Order Accurate SAV Numerical Method for the Nonlocal Ternary Conservative Allen–Cahn Model," *Applied Mathematics Letters* 142 (2023): 108633, https://doi.org/10.1016/j.aml.2023.108633.

21. Z. Weng, S. Zhai, W. Dai, Y. Yang, and Y. Mo, "Stability and Error Estimates of Strang Splitting Method for the Nonlocal Ternary Conservative Allen–Cahn Model," *Journal of Computational and Applied Mathematics* 441 (2024): 115668, https://doi.org/10.1016/j.cam.2023.115668.

22. C. Zhou and Y. Q. Zu, "A Lattice Boltzmann Model for the Interface Tracking of Immiscible Ternary Fluids Based on the Conservative Allen-Cahn Equation," *Computers and Fluids* 267 (2023): 106093, https://doi.org/10.1016/j.compfluid.2023.106093.

23. D. A. Kay and A. Tomasi, "Color Image Segmentation by the Vector-Valued Allen–Cahn Phase-Field Model: A Multigrid Solution," *IEEE Transactions on Image Processing* 18, no. 10 (2009): 2330–2339, https://doi.org/10.1109/TIP.2009.2026678.

24. C. Liu, Z. Qiao, and Q. Zhang, "Multi-Phase Image Segmentation by the Allen–Cahn Chan–Vese Model," *Computers & Mathematics with Applications* 141 (2023): 207–220, https://doi.org/10.1016/j.camwa.2022.12.020.

25. J. Wang and Z. Shi, "Multi-Reconstruction From Points Cloud by Using a Modified Vector-Valued Allen–Cahn Equation," *Mathematics* 9, no. 12 (2021): 1326, https://doi.org/10.3390/math9121326.

26. Z. Wang, J. Zhang, and X. Yang, "Fully Discrete Spectral-Galerkin Scheme for a Ternary Allen–Cahn Type Mass-Conserved Nakazawa–Ohta Phase-Field Model for Triblock Copolymers," *Journal of Computational and Applied Mathematics* 419 (2023): 114699, https://doi.org/10.1016/j.cam.2022.114699.

27. M. Emamjomeh, M. Nabati, and A. Dinmohammadi, "Numerical Study of Two Operator Splitting Localized Radial Basis Function Method for Allen–Cahn Problem," *Engineering Analysis with Boundary Elements* 163 (2024): 126–137, https://doi.org/10.1016/j.enganabound.2024.02.016.

28. H. Kim, S. Kang, G. Lee, S. Yoon, and J. Kim, "Shape Transformation on Curved Surfaces Using a Phase-Field Model," *Communications in Nonlinear Science and Numerical Simulation* 133 (2024): 107956, https://doi.org/10.1016/j.cnsns.2024.107956.

29. Z. Cao, Z. Weng, and S. Zhai, "An Effective Operator Splitting Scheme for General Motion by Mean Curvature Using a Modified Allen–Cahn Equation," *Applied Mathematics Letters* 163 (2025): 109472, https://doi.org/10.1016/j.aml.2025.109472.

30. J. Choi, S. Ham, S. Kwak, Y. Hwang, and J. Kim, "Stability Analysis of an Explicit Numerical Scheme for the Allen–Cahn Equation With High-Order Polynomial Potentials," *AIMS Mathematics* 9, no. 7 (2024): 19332–19344, https://doi.org/10.3934/math.2024941.

31. S. Kwak, "Practical Implementation of Boundary Conditions in the Thomas Algorithm," *Journal of the Korean Society for Industrial and Applied Mathematics* 29, no. 2 (2025): 171–183, https://doi.org/10.12941/jksiam.2025.29.171.

32. J. Yang, Y. Li, and J. Kim, "Modified Multi-Phase Diffuse-Interface Model for Compound Droplets in Contact With Solid," *Journal of Computational Physics* 491 (2023): 112345, https://doi.org/10.1016/j.jcp.2023.112345.

33. Y. Hwang, Jyoti, S. Kwak, H. Kim, and J. Kim, "An Explicit Numerical Method for the Conservative Allen–Cahn Equation on a Cubic Surface," *AIMS Mathematics* 9, no. 12 (2024): 34447–34465, https://doi.org/10.3934/math.20241641.

Appendix A

The following Listing 1 is MATLAB code for the tAC equation.

LISTING 1. MATLAB code for the tAC equation

```
clear all;
Lx = -1; Rx = 1; Ly = -1; Ry = 1; Nx = 100; Ny = 100; h = (Rx-Lx)/Nx;
x = linspace(Lx+0.5*h,Rx-0.5*h,Nx); y = linspace(Ly+0.5*h,Ry-0.5*h,Ny);
[X,Y] = meshgrid(x,y);

ep=h; dt = h^2; T = 0.024; Nt = round(T/dt); ns = round(Nt/3);
tol = 1e-7; coef = ep^2/dt;

alp = -dt*ones(1,Nx)/h^2; gam = -dt*ones(1,Nx)/h^2;
bet = 1+2*dt*ones(1,Nx)/h^2; bet([1,Nx]) = 1+dt/h^2;

c0 = zeros(Nx,Ny,3);
c0(:,:,1)=(1+(1+tanh((0.2-sqrt(X.^2+(Y-2/3).^2))/(2*sqrt(2)*ep)))...
    -0.5*(1+tanh((0.2-sqrt((X-1/sqrt(3)).^2+(Y+1/3).^2))/(2*sqrt(2)*ep)))...
    -0.5*(1+tanh((0.2-sqrt((X+1/sqrt(3)).^2+(Y+1/3).^2))/(2*sqrt(2)*ep))))/3;
c0(:,:,2)=(1-0.5*(1+tanh((0.2-sqrt(X.^2+(Y-2/3).^2))/(2*sqrt(2)*ep)))...
    +(1+tanh((0.2-sqrt((X-1/sqrt(3)).^2+(Y+1/3).^2))/(2*sqrt(2)*ep)))...
    -0.5*(1+tanh((0.2-sqrt((X+1/sqrt(3)).^2+(Y+1/3).^2))/(2*sqrt(2)*ep))))/3;
c0(:,:,3)=1-sum(c0(:,:,1:2),3);

figure(1); clf; hold on; box on
contourf(x,y,c0(:,:,1),[0.5,0.5],'FaceColor',"r",'FaceAlpha',0.5)
contourf(x,y,c0(:,:,2),[0.5,0.5],'FaceColor',"g",'FaceAlpha',0.5)
contourf(x,y,c0(:,:,3),[0.5,0.5],'FaceColor',"b",'FaceAlpha',0.5)
axis image; axis([Lx,Rx,Ly,Ry])
text(0,2/3,"c_1","HorizontalAlignment",'center')
text(1/sqrt(3),-1/3,"c_2","HorizontalAlignment",'center')
text(-1/sqrt(3),-1/3,"c_3","HorizontalAlignment",'center')

% Preallocate
c = c0; nc = zeros(Nx,Ny,3); ki = mod((1:3),3)+1; kj = mod((1:3)+1,3)+1;
for it=1:Nt
    oc = c;
    %%% solving the nonlinear term
    for k=1:3
        p = -0.25 - oc(:,:,ki(k)).*oc(:,:,kj(k)) + coef;
        q = -0.5*oc(:,:,ki(k)).*oc(:,:,kj(k)) + 0.5*coef*(1-2*oc(:,:,k));
        c(:,:,k) = 0.5 + nthroot(-0.5*q+sqrt(0.25*q.^2+p.^3/27),3) ...
            + nthroot(-0.5*q-sqrt(0.25*q.^2+p.^3/27),3);
        %%% solving the diffusion term
        for i=1:Nx
            nc(i,:,k)=thomas(alp, bet, gam, c(i,:,k));
        end
        for j=1:Ny
            c(:,j,k)=thomas(alp, bet, gam, nc(:,j,k)')';
        end
    end
    %%% normalization
    c = c./sum(c,3);
    %%% for drawing figure %%%
    if mod(it,ns)==0
        figure(1); clf; hold on; box on
        contourf(x,y,c(:,:,1),[0.5,0.5],'FaceColor',"r",'FaceAlpha',0.5)
        contourf(x,y,c(:,:,2),[0.5,0.5],'FaceColor',"g",'FaceAlpha',0.5)
        contourf(x,y,c(:,:,3),[0.5,0.5],'FaceColor',"b",'FaceAlpha',0.5)
        axis image; axis([Lx,Rx,Ly,Ry])
        yticks([-1,0,1])
```

```
        text(0,2/3,"c_1","HorizontalAlignment",'center')
        text(1/sqrt(3),-1/3,"c_2","HorizontalAlignment",'center')
        text(-1/sqrt(3),-1/3,"c_3","HorizontalAlignment",'center')
    end
end
figure(1); clf; hold on; box on
contourf(x,y,c(:,:,1),[0.5,0.5],'FaceColor',"r",'FaceAlpha',0.5)
contourf(x,y,c(:,:,2),[0.5,0.5],'FaceColor',"g",'FaceAlpha',0.5)
contourf(x,y,c(:,:,3),[0.5,0.5],'FaceColor',"b",'FaceAlpha',0.5)
axis image; axis([Lx,Rx,Ly,Ry])
text(0,2/3,"c_1","HorizontalAlignment",'center')
text(1/sqrt(3),-1/3,"c_2","HorizontalAlignment",'center')
text(-1/sqrt(3),-1/3,"c_3","HorizontalAlignment",'center')

function xx = thomas(aa, bb, gg, ff)
m = length(ff);
for k = 2:m
    mult = aa(k)/bb(k-1);
    bb(k) = bb(k) - mult*gg(k-1);
    ff(k) = ff(k) - mult*ff(k-1);
end
xx(m) = ff(m)/bb(m);
for k = m-1:-1:1
    xx(k) = (ff(k) - gg(k)*xx(k+1))/bb(k);
end
end
```