Chapter 2

Cahn-Hilliard Fluid Dynamics

2.1 Introduction

The Cahn-Hilliard equation is the prototypical continuum model of phase separation. It was originally proposed by Cahn and Hilliard [31] to model binary alloys and has subsequently been adopted to model many other physical situations such as phase transitions and interface dynamics in multiphase fluids [78]. This is what we consider here. Phase separation occurs, for example, when a single phase homogeneous system composed of two fluid components, in thermal equilibrium (e.g. at a high temperature), is rapidly cooled to a temperature T below a critical temperature T_c, where the system is unstable with respect to infinitesimal concentration fluctuations. Spinodal decomposition then takes place and the system separates into spatial regions rich in one component and poor in the other. The evolution lowers the free energy and leads to an equilibrium state with coexisting phases (see the papers [18], [99]). Assuming that the fluid components are incompressible ($\nabla \cdot \mathbf{u} = 0$) with equal densities (set to one for simplicity) and that the evolution is isothermal, the Cahn-Hilliard (**C-H**) model is as follows. Let c be the phase variable (i.e. concentration), then

$$c_t(\mathbf{x},t) + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (M(c)\nabla\mu(\mathbf{x},t)) \text{ , for } (\mathbf{x},t) \in \Omega \times [0,T] \subset \mathbb{R}^n \times \mathbb{R} \tag{2.1.1}$$

and

$$\mu(\mathbf{x},t) = \phi(c(\mathbf{x},t)) - \epsilon^2 \Delta c(\mathbf{x},t), \tag{2.1.2}$$

where $\mathbf{u}$ is the mass-averaged fluid velocity (i.e. $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$, where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the velocities of the two components), M is the mobility, μ is the generalized chemical potential, $\phi(c) = F'(c)$, and $F(c)$ is the Helmholtz free energy which is non convex if $T < T_c$, to reflect the coexistence of separate phases and $\epsilon > 0$ is a measure of non-locality (gradient energy) and introduces an internal length scale

(interface thickness). See also [4, 77, 93] for further details and references. Here, for simplicity, we consider a constant mobility ($M \equiv 1$) and a quartic free energy $F(c)$, which is defined by

$$F(c) = \frac{1}{4}c^2(c-1)^2. \tag{2.1.3}$$

Thus, the coexisting phases correspond to $c = 0$ and 1. The natural boundary and initial conditions for the (**C-H**) equation are

$$\frac{\partial c}{\partial n} = \frac{\partial \mu}{\partial n} = 0, \text{ and } \quad \mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega, \qquad c(\mathbf{x}, 0) = c_0(\mathbf{x}), \quad \mathbf{u}(\mathbf{x}, t) = \mathbf{0}, \tag{2.1.4}$$

where n is the normal unit vector pointing out of Ω.

Two important features of the (**C-H**) problem in the case of zero Neumann boundary conditions are the conservation of mass $\frac{1}{|\Omega|}\int_\Omega c(x,t)d\mathbf{x}$, and the existence of a Lyapunov(Energy) functional $J(c)$

$$J(c) = \int_\Omega \left[F(c) + \frac{\epsilon^2}{2}|\nabla c|^2\right] \, d\mathbf{x} \tag{2.1.5}$$

such that

$$\frac{d}{dt}J(c) = -\int_\Omega |\nabla \mu|^2 \, d\mathbf{x},$$

in the absence of flow. The second feature plays a crucial role in the analysis of the (**C-H**) equation, including the proof of the existence of a solution to the initial boundary value problem [53], and asymptotic long time behavior [123], [114]. The energy functional also readily yields a pointwise estimate of c in the one-dimensional case. In this chapter, we develop a finite difference scheme that inherits mass conservation and energy dissipation from the continuous level. It is highly desirable to have a discrete energy functional because this implies that the numerical solution is uniformly bounded with respect to the time and space step sizes from which it follows that the scheme is stable.

The (**C-H**) equation, even without flow, is challenging to solve numerically for two reasons. First, the equation is fourth order in space which makes straightforward difference stencils very large and introduces a severe time step restriction for stability (stiffness), i.e., $\Delta t \sim \Delta x^4$ for explicit methods. Second, there is nonlinearity associated with ϕ, which can also contribute to numerical stiffness. To overcome these difficulties, we split the fourth order equation into a system of second order equations. We then use a fully implicit time discretization. A new nonlinear multigrid method is developed to solve the nonlinear discrete system to obtain the solution at the new time step.

In the presence of flow, the advection term is implemented as a forcing function in the nonlinear multigrid scheme. The functional $J(c)$ now may either increase or decrease in time and satisfies

$$\frac{d}{dt}J(c) + \int_\Omega \mu\nabla\cdot(c\mathbf{u}) \, d\mathbf{x} = -\int_\Omega |\nabla\mu|^2 \, d\mathbf{x}.$$

If the interfaces are passive, by which we mean that the concentration field c does not affect the flow field, then $\mathbf{u}$ satisfies the classical Navier-Stokes equations and may be imposed independently of the concentration field c. If the interfaces are active, on the other hand, the velocity field $\mathbf{u}$ depends on c through the introduction of extra stresses that mimic the surface tension between the two fluid components. In this case, the system energy is given by

$$E_{tot} = \int_\Omega |\mathbf{u}|^2/2 \, dx + \frac{\sigma}{\epsilon} J(c), \tag{2.1.6}$$

where the first term is the kinetic energy of the fluid system and σ is related to the surface tension [93]. Note the dependence of the second term upon ϵ. The velocity satisfies a generalized Navier-Stokes (**N-S**) system:

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p - \frac{\sigma}{\epsilon} c \nabla \mu + \nabla \cdot \left(\eta(c) \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right), \tag{2.1.7}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.1.8}$$

where the extra stress due to the concentration gradients (i.e. interfaces) is $\frac{\sigma}{\epsilon} \mu \nabla c$. In [92], it is shown using the method of matched asymptotic expansions that this term converges to the classical surface tension force as $\epsilon \to 0$. This (**NSCH**) system is known as Model H in the notation of Hohenberg & Halperin [77]. We refer the reader to the recent review paper by Anderson, McFadden & Wheeler [4] and to [93], for example, for further details on the model. The extension of this system to the more realistic case of multiphase fluids whose components have different densities is extensively discussed in [93]. The system energy, assuming no-slip ($\mathbf{u} = 0$) boundary conditions, satisfies

$$\frac{d}{dt} E_{tot} = -\frac{\sigma}{\epsilon} \int_\Omega |\nabla \mu|^2 \, d\mathbf{x} - \frac{1}{2} \int_\Omega \eta(c) \mathbf{D} : \mathbf{D} \, dx, \tag{2.1.9}$$

where $\mathbf{D} = \nabla \mathbf{u} + \nabla \mathbf{u}^T$ is the scaled deformation tensor.

In this paper, we consider both passive and active interfaces where the generalized (**N-S**) equations are solved using a second-order accurate finite difference projection method. The resulting discretization of the (**NSCH**) system (2.1.1-2.1.2) and (2.1.7- 2.1.8)

(i) preserves the mass (average of c) on the discrete level;

(ii) gives a simple treatment of boundary conditions;

(iii) has a discrete equivalent of the Lyapunov function E_{tot};

(iv) has a constraint on the time step which depends only on the physical quantities ϕ, ϵ, σ and η and not on the spatial or temporal discretizations;

(v) extends easily to the multi-component case (i.e. two or more concentration fields).

One of the main achievements of this paper is the development of an efficient nonlinear multigrid method to solve the discrete nonlinear scheme for the (**C-H**) equation. To our knowledge this is the first work in which a nonlinear multigrid method is used to solve this equation. Following the general strategy outlined in [11], we design a smoother based on treating the (**C-H**) equation as a system of two equations for c and μ as in (2.1.1-2.1.2). This smoother is of pointwise collective Newton-Gauss-Seidel type [11], in which the nonlinearity is linearized about the current grid point and a Jacobi type update is used for the remaining nonlinear terms. A Gauss-Seidel type update is used for the linear terms and c and μ are updated simultaneously. We demonstrate the excellent performance of the numerical method by simulating various regimes of the (**NSCH**) model including spinodal decomposition.

In the absence of flow, there has been much algorithm development and many simulations of the (**C-H**) equation using finite element methods (e.g. [19], [20], [21], [22], [53], [55], [56], [57], [61]), finite difference algorithms (e.g. [58], [62], [63], [118]) and spectral methods (e.g. [90]). Most of these finite difference and finite element references use conservative algorithms with discrete energy functionals. The discretization that is closest to ours is given in [53] in the context of finite element methods. In [53], the (**C-H**) is treated as a system of equations for c and μ and a Crank-Nicholson type time discretization is used where the free energy derivative $\phi = F'(c)$ is approximated so as to yield a scheme with a discrete energy functional for any value of the time step (the scheme is nonlinear at the implicit time level). Here, we choose a different approximation for ϕ on the discrete level that yields enhanced stability over the method presented in [53]. This alternative approximation also allows us to extend systematically the discrete system to the case of ternary mixtures; the scheme presented in [53] does not have such a straightforward ternary extension. In spite of recent algorithmic developments of numerical approximations to the (**C-H**) equations, the solution of the discrete equations has remained problematic due to the nonlinearity of the implicit scheme. By using the nonlinear multigrid method to obtain the numerical solution at the implicit time level, we gain improved numerical stability and efficiency over standard solution techniques based on Newton's method and over algorithms for which the nonlinear term ϕ is treated as a forcing function.

In the presence of flow, there has been much recent work on simulating multicomponent fluid flows using Cahn-Hilliard (diffuse interface) models. We again refer the reader the review [4] and to the discussion below for references. To our knowledge, the scheme we present here is the first to have an associated Lyapunov function E_{tot} on the fully discrete level for any value of the time and space steps in presence of flow. Here, we examine the effect of shear flow and interfacial tension on the decomposition. The solution of the Cahn-Hilliard equation and the resolution of the associated extra stresses in the fluid

are typically the bottlenecks of these simulations. Our improved (**C-H**) solver together with our energy-preserving discretization for the (**NSCH**) system, which has the advantageous side-effect of improving the accuracy of the extra-stress and results in improved stability and accuracy of the multicomponent flow simulations.

2.2 Numerical analysis

In this section, we present semi-discrete and fully discrete schemes for the (**NSCH**) system. In addition, we prove discrete versions of mass conservation and energy dissipation, which immediately imply the stability of the numerical scheme. Finally, proof of convergence of the numerical solution is also established.

2.A Discretization

We shall first discretize the (**C-H**) equation (2.1.1-2.1.2) in space. Let $[a, b]$ and $[c, d]$ be partitioned by

$$a = x_{\frac{1}{2}} < x_{1+\frac{1}{2}} < \cdots < x_{N_x-1+\frac{1}{2}} < x_{N_x+\frac{1}{2}} = b,$$

$$c = y_{\frac{1}{2}} < y_{1+\frac{1}{2}} < \cdots < y_{N_x-1+\frac{1}{2}} < y_{N_y+\frac{1}{2}} = d$$

so that the cells $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], 1 \le i \le N_x, 1 \le j \le N_y$ cover $\Omega = [a, b] \times [c, d]$. Let

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}.$$

For simplicity, we assume that the above partitions are uniform in both directions so that

$$\Delta x_i = \Delta y_j = h \quad \text{for } 1 \le i \le N_x, \quad 1 \le j \le N_y$$

where $h = (b - a)/N_x = (d - c)/N_y$. Therefore

$$x_{i+\frac{1}{2}} = a + ih, \qquad y_{j+\frac{1}{2}} = c + jh$$

and let $\Omega_h = \{(x_i, y_j) : 1 \le i \le N_x, \ \ 1 \le j \le N_y\}$, be the set of cell-centers where

$$x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \quad y_j = \frac{1}{2}(y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}}).$$

The set of cell-corners is $\Omega_{h,\frac{1}{2}} = \{(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) : 0 \le i \le N_x, \ \ 0 \le j \le N_y\}$.

Since the concentration c and the chemical potential μ satisfy Neumann boundary conditions, it is natural to define them at cell centers. Let c_{ij} and μ_{ij} be approximations of $c(x_i, y_j)$ and $\mu(x_i, y_j)$. We first implement the zero Neumann boundary condition (2.1.4) by requiring that

$$D_x c_{i-\frac{1}{2},j} = 0 \quad \text{for } i = 0, \qquad D_x c_{i+\frac{1}{2},j} = 0 \quad \text{for } i = N_x,$$
$$D_y c_{i,j-\frac{1}{2}} = 0 \quad \text{for } j = 0, \qquad D_y c_{i,j+\frac{1}{2}} = 0 \quad \text{for } j = N_y, \tag{2.2.10}$$

where the discrete differentiation operators are

$$D_x c_{i+\frac{1}{2},j} = \frac{1}{h}(c_{i+1,j} - c_{i,j}), \quad D_y c_{i,j+\frac{1}{2}} = \frac{1}{h}(c_{i,j+1} - c_{i,j}).$$

We then define the discrete Laplacian by

$$\Delta_d c_{ij} = \frac{1}{h}(D_x c_{i+\frac{1}{2},j} - D_x c_{i-\frac{1}{2},j}) + \frac{1}{h}(D_y c_{i,j+\frac{1}{2}} - D_y c_{i,j-\frac{1}{2}}),$$

and the discrete L^2 inner product by

$$(c_1, c_2)_h = h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} c_{1ij} c_{2ij}. \tag{2.2.11}$$

For a grid function c defined at cell centers, $D_x c$ and $D_y c$ are defined at cell-edges, and we use the following notation

$$\nabla_d^e c_{ij} = (D_x c_{i+\frac{1}{2},j}, D_y c_{i,j+\frac{1}{2}}),$$

to represent the discrete gradient of c at cell-edges. Correspondingly, the (MAC) divergence at cell-centers, using values from cell-edges, is

$$\tilde{\nabla}_d^e \cdot \mathbf{g}_{ij} = \frac{1}{h}\left(g^1_{i+\frac{1}{2},j} - g^1_{i-\frac{1}{2},j}\right) + \frac{1}{h}\left(g^2_{i,j+\frac{1}{2}} - g^2_{i,j-\frac{1}{2}}\right),$$

for a grid function $\mathbf{g} = (g^1, g^2)$ defined on cell-edges. We can define an inner product for $\nabla_d^e c$ on the staggered grid by

$$(\nabla_d^e c_1, \nabla_d^e c_2)_e = h^2 (\sum_{i=0}^{N_x} \sum_{j=1}^{N_y} D_x c_{1\,i+\frac{1}{2},j} D_x c_{2\,i+\frac{1}{2},j} + \sum_{i=1}^{N_x} \sum_{j=0}^{N_y} D_y c_{1\,i,j+\frac{1}{2}} D_y c_{2\,i,j+\frac{1}{2}}). \tag{2.2.12}$$

We also define discrete norms associated with (2.2.11) and (2.2.12) as

$$||c||^2 = (c, c)_h, \qquad |c|^2_{e,1} = (\nabla_d^e c, \nabla_d^e c)_e.$$

The time-continuous, space-discrete system that corresponds to (2.1.1-2.1.4) in the absence of flow is

$$\frac{d}{dt} c_{ij} = \Delta_d \mu_{ij}, \quad \mu_{ij} = \phi(c_{ij}) - \epsilon^2 \Delta_d c_{ij}, \tag{2.2.13}$$

where ϕ is defined in (2.1.3) and boundary conditions are implemented using (2.2.10). It is easy to see that this discretization is second order accurate in space and that mass is conserved identically. The

scheme also has an energy functional given by the discretization of (2.1.5). We discretize (2.2.13) in time by the Crank-Nicholson type algorithm:

$$\frac{c_{ij}^{n+1}-c_{ij}^{n}}{\Delta t}=\Delta_d \mu_{ij}^{n+\frac{1}{2}}, \tag{2.2.14}$$

$$\mu_{ij}^{n+\frac{1}{2}}=\hat{\phi}(c_{ij}^{n},c_{ij}^{n+1})-\frac{\epsilon^2}{2}\Delta_d(c_{ij}^{n}+c_{ij}^{n+1}), \tag{2.2.15}$$

where

$$\hat{\phi}(c_1,c_2)=\phi(c_2)-\frac{1}{2}\phi'(c_2)(c_2-c_1)+\frac{1}{3!}\phi''(c_2)(c_2-c_1)^2. \tag{2.2.16}$$

This is obtained by using the Taylor expansion of $(F(c_1)-F(c_2))/(c_1-c_2)$ and retaining terms up to the second order derivative. This is a modification of the scheme presented in [53], where $\hat{\phi}(c_1,c_2)$ is defined as follow.

$$\hat{\phi}(c_1,c_2) \quad = \quad \begin{cases} \frac{F(c_1)-F(c_2)}{c_1-c_2}, & \text{if } c_1\neq c_2 \\ \\ \phi(c_1), & \text{if } c_1=c_2 \end{cases}$$

And unlike the scheme in [53], our modified scheme can be easily extended to multi-component systems.

In the presence of flow $\mathbf{u}=(u,v)\neq 0$, we use the center difference operator to define the discrete gradient and divergence operators at cell-centers respectively by

$$\begin{aligned} \nabla_d c_{ij} &= \frac{1}{2}\left(D_x c_{i+\frac{1}{2},j}+D_x c_{i-\frac{1}{2},j}, D_y c_{i,j+\frac{1}{2}}+D_y c_{i,j-\frac{1}{2}}\right), \\ \nabla_d\cdot\mathbf{u}_{ij} &= \frac{1}{2}\left(D_x u_{i+\frac{1}{2},j}+D_x u_{i-\frac{1}{2},j}\right)+\frac{1}{2}\left(D_y v_{i,j+\frac{1}{2}}+D_y v_{i,j-\frac{1}{2}}\right). \end{aligned}$$

The non-slip boundary conditions are implemented by introducing a ring of ghost-cells surrounding the physical domain such that sum of the velocities (at cell-centers) are equal to zero on the physical boundary, i.e.

$$\mathbf{u}_{0,j}=-\mathbf{u}_{1,j},\ \ \mathbf{u}_{N_x+1,j}=-\mathbf{u}_{N_x,j},\ \ \mathbf{u}_{i,0}=-\mathbf{u}_{i,1}\ \text{ and }\ \mathbf{u}_{i,N_y+1}=-\mathbf{u}_{i,N_y}. \tag{2.2.17}$$

Accordingly, equation (2.2.14) becomes by

$$\frac{c_{ij}^{n+1}-c_{ij}^{n}}{\Delta t}+\nabla_d\cdot\left(\mathbf{u}_{ij}^{n+\frac{1}{2}}c_{ij}^{n+\frac{1}{2}}\right)=\Delta_d\mu_{ij}^{n+\frac{1}{2}}, \tag{2.2.18}$$

where $\mathbf{u}^{n+\frac{1}{2}}=\left(\mathbf{u}^{n+1}+\mathbf{u}^{n}\right)/2$, and $c^{n+\frac{1}{2}}$ is defined analogously. The chemical potential $\mu_{ij}^{n+\frac{1}{2}}$ is still obtained from (2.2.15).

For active interfaces, the velocity $\mathbf{u}$ satisfies the generalized Navier-Stokes equations (2.1.8). Here, we will use the rotation form (i.e. $\mathbf{u}\cdot\nabla\mathbf{u}=\omega\times\mathbf{u}+\frac{1}{2}\nabla|\mathbf{u}|^2$) of the equations, together with the

Bernoulli pressure $P = p + \frac{1}{2}|\mathbf{u}|^2$. To solve this system, we will use a projection method. Following [6], the velocity components are defined at cell centers where $\mathbf{u}_{ij} \approx \mathbf{u}(x_i, y_j)$ and the pressure is defined at cell-corners ($\Omega_{h,\frac{1}{2}}$) where $P_{i+\frac{1}{2},j+\frac{1}{2}} \approx P(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$. In the corresponding discrete system, we additionally define gradient and divergence operators taking values from cell-centers to cell-corners and vice-versa. These operators are:

$$\begin{aligned}
\nabla_d^c \cdot \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}\left(D_x c_{i+\frac{1}{2},j} + D_x c_{i+\frac{1}{2},j+1}\right) + \frac{1}{2}\left(D_y c_{i,j+\frac{1}{2}} + D_y c_{i+1,j+\frac{1}{2}}\right), \\
\tilde{\nabla}_d^c P_{i,j} &= \frac{1}{2h}\Big(P_{i+\frac{1}{2},j+\frac{1}{2}} + P_{i+\frac{1}{2},j-\frac{1}{2}} - P_{i-\frac{1}{2},j+\frac{1}{2}} - P_{i-\frac{1}{2},j-\frac{1}{2}}, \\
&\qquad P_{i+\frac{1}{2},j+\frac{1}{2}} - P_{i+\frac{1}{2},j-\frac{1}{2}} + P_{i-\frac{1}{2},j+\frac{1}{2}} - P_{i-\frac{1}{2},j-\frac{1}{2}}\Big),
\end{aligned}$$

We then use the following discretization:

$$\nabla_d^c \cdot \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = 0, \tag{2.2.19}$$

$$\frac{\mathbf{u}_{ij}^{n+1} - \mathbf{u}_{ij}^{n}}{\Delta t} + \omega_{ij}^{n+\frac{1}{2}} \times \mathbf{u}_{ij}^{n+\frac{1}{2}} = -\tilde{\nabla}_d^c P_{ij}^{n+\frac{1}{2}} - \frac{\sigma}{\epsilon} c_{ij}^{n+\frac{1}{2}} \nabla_d \mu_{ij}^{n+\frac{1}{2}} + \nabla_d \cdot \left(\eta(c_{ij}^{n+\frac{1}{2}}) \mathbf{D}_{ij}^{n+\frac{1}{2}}\right) \tag{2.2.20}$$

where the vorticity is $\omega_{ij}^{n+\frac{1}{2}} = \nabla_d \times \mathbf{u}_{ij}^{n+\frac{1}{2}}$, and the (scaled) rate of deformation tensor is $\mathbf{D}_{ij}^{n+\frac{1}{2}} = \nabla_d \mathbf{u}_{ij}^{n+\frac{1}{2}} + \left(\nabla_d \mathbf{u}_{ij}^{n+\frac{1}{2}}\right)^T$. If the viscosity η is constant, we instead replace the viscous term in Eq. (2.2.20) by $\eta \Delta_d \mathbf{u}_{ij}^{n+\frac{1}{2}}$. As we will demonstrate in the next section, the discrete system (2.2.18)-(2.2.20) in fact has an energy functional given by the discretization of (2.1.6).

The numerical implementation of Eqs. (2.2.14)-(2.2.15) and (2.2.19)-(2.2.20) is discussed in sections 2.3 and 2.4 where it is shown how the implicit solutions at time t_{n+1} are obtained using a nonlinear multigrid method and a projection method respectively.

2.B Analysis of Scheme

We next analyze the numerical schemes. Before we proceed, we first state without proof the following lemma which is a easy consequence of discrete summation by parts.

Lemma 2.1. *Let c_1 and c_2 be defined on Ω_h satisfying (2.2.10). Let* **u** *be defined on Ω_h and satisfy (2.2.17). Finally, let P be defined on $\Omega_{h,\frac{1}{2}}$. Then,*

$$(c_1, \Delta_d c_2)_h = (\Delta_d c_1, c_2)_h = -(\nabla_d^e c_1, \nabla_d^e c_2)_e, \tag{2.2.21}$$

$$(\mathbf{u}, \nabla_d c_1)_h = -(\nabla_d \cdot \mathbf{u}, c_1)_h, \text{ and} \tag{2.2.22}$$

$$(\mathbf{u}, \tilde{\nabla}_d^c P)_h = -(\nabla_d^c \cdot \mathbf{u}, P)_{h,\frac{1}{2}}, \tag{2.2.23}$$

where the cell-corner inner product is given by the trapezoidal rule approximation of the continuous inner product:

$$(\nabla_d^c \cdot \mathbf{u}, P)_{h,\frac{1}{2}} = \frac{h^2}{2}\sum_{i=0}^{N_x}\sum_{j=0}^{N_y} \nabla_d^c \cdot \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}} P_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{h^2}{2}\sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y-1} \nabla_d^c \cdot \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}} P_{i+\frac{1}{2},j+\frac{1}{2}}$$

The following lemma establishes the mass conservation and the existence of a discrete energy functional in the absence of flow.

Lemma 2.2. *If $\{c^n, \mu^{n+\frac{1}{2}}\}$ is the solution of (2.2.14-2.2.15) and if we define the discrete energy functional by*

$$\mathcal{F}_h(c) = (F(c), 1)_h + \frac{\epsilon^2}{2}|c|_{e,1}^2, \tag{2.2.24}$$

where F is defined in (2.1.3), then

$$\begin{aligned} (c^{n+1}, 1)_h &= (c^n, 1)_h, \\ \mathcal{F}_h(c^{n+1}) - \mathcal{F}_h(c^n) &= -\Delta t|\mu^{n+\frac{1}{2}}|_{e,1}^2 - \frac{1}{4}((c^{n+1}-c^n)^4, 1)_h. \end{aligned} \tag{2.2.25}$$

Proof. The first assertion is due to the combination of (2.2.14) and the discrete version of integration by parts in lemma 2.1. Indeed,

$$(c^{n+1}, 1)_h = (c^n, 1)_h + \Delta t(\Delta_d \mu^{n+\frac{1}{2}}, 1)_h = (c^n, 1)_h - \Delta t(\nabla_d \mu^{n+\frac{1}{2}}, \nabla_d 1)_e = (c^n, 1)_h.$$

It remains to prove the second assertion. Multiplying $\mu^{n+\frac{1}{2}}$ and $c^{n+1} - c^n$ to (2.2.14) and (2.2.15), respectively and summing by parts, we obtain the following two identities

$$(c^{n+1} - c^n, \mu^{n+\frac{1}{2}})_h + \Delta t|\mu^{n+\frac{1}{2}}|_{e,1}^2 = 0,$$

$$(c^{n+1} - c^n, \mu^{n+\frac{1}{2}})_h - \frac{\epsilon^2}{2}|c^{n+1}|_{e,1}^2 + \frac{\epsilon^2}{2}|c^n|_{e,1}^2 = (\hat{\phi}(c^n, c^{n+1}), c^{n+1} - c^n)_h.$$

Using the identities above, we obtain

$$\begin{aligned} \mathcal{F}_h(c^{n+1}) - \mathcal{F}_h(c^n) &= \tfrac{\epsilon^2}{2}|c^{n+1}|_{e,1}^2 - \tfrac{\epsilon^2}{2}|c^n|_{e,1}^2 + (F(c^{n+1}) - F(c^n), 1)_h \\ &= -\Delta t|\mu^{n+\frac{1}{2}}|_{e,1}^2 - (\hat{\phi}(c^n, c^{n+1}), c^{n+1} - c^n)_h \\ &\quad + (F(c^{n+1}) - F(c^n), 1)_h. \end{aligned}$$

From the Taylor expansion, we have

$$\begin{aligned} F(c^{n+1}) - F(c^n) &= \phi(c^{n+1})(c^{n+1} - c^n) - \tfrac{1}{2}\phi'(c^{n+1})(c^{n+1} - c^n)^2 \\ &\quad + \tfrac{1}{3!}\phi''(c^{n+1})(c^{n+1} - c^n)^3 - \tfrac{1}{4!}\phi'''(c^{n+1})(c^{n+1} - c^n)^4. \end{aligned}$$

Since $\phi'''(c^{n+1}) = 6$, we obtain

$$\begin{aligned} & \mathcal{F}_h(c^{n+1}) - \mathcal{F}_h(c^n) + \Delta t|\mu^{n+\frac{1}{2}}|^2_{e,1} \\ = \quad & -(\hat{\phi}(c^n, c^{n+1}), c^{n+1} - c^n)_h + (F(c^{n+1}) - F(c^n), 1)_h \\ = \quad & -\tfrac{6}{4!}((c^{n+1} - c^n)^4, 1)_h = -\tfrac{1}{4}((c^{n+1} - c^n)^4, 1)_h. \end{aligned}$$

This completes the proof. □

From lemma 2.2, it follows that the numerical solution $||c^n||$ is bounded. This yields stability (in l^2) of the numerical scheme. The presence of the second term on the right hand side of Eq. (2.2.25) suggests that our method is more stable than that of [53] where this term is absent. Lemma 2.2 still holds for regular solution model free energies of the form [82]:

$$F(c) := \theta\left[c\ln(c) + (1-c)\ln(1-c)\right] - 2\theta_c c(1-c),$$

(θ and θ_c are the absolute and the critical temperature, respectively) that are regularized by fourth order polynomials near the singular points, i.e.,

$$F_\delta(c) = \begin{cases} p_l(c) & \text{if} \quad c \le \delta, \\ F(c) & \text{if} \quad \delta < c < 1-\delta, \\ p_r(c) & \text{if} \quad c \ge 1-\delta, \end{cases}$$

where $p_l(c)$ and $p_r(c)$ are fourth order polynomials which match values with $F(c)$ up to fourth order derivatives at $c = \delta$ and $c = 1 - \delta$, respectively and δ is a small positive parameter.

Next, we demonstrate the existence of an energy functional in the presence of flow.

Lemma 2.3. *Let $\{c^n, \mu^{n-\frac{1}{2}}, \mathbf{u}^n\}$ be the solutions of (2.2.14)-(2.2.15), and (2.2.19)-(2.2.20) and let the discrete total energy functional be*

$$\mathcal{E}_h(c, \mathbf{u}) = \frac{1}{2}(\mathbf{u}, \mathbf{u})_h + \frac{\sigma}{\epsilon}\left((F(c), 1)_h + \frac{\epsilon^2}{2}|c|^2_1\right), \tag{2.2.26}$$

then

$$\begin{aligned} \frac{\mathcal{E}_h(c^{n+1}, \mathbf{u}^{n+1}) - \mathcal{E}_h(c^n, \mathbf{u}^n)}{\Delta t} \quad = \quad & -\eta\left(|u^{n+\frac{1}{2}}|^2_{e,1} + |v^{n+\frac{1}{2}}|^2_{e,1}\right) - \frac{\sigma}{\epsilon}|\mu^{n+\frac{1}{2}}|^2_{e,1} \\ & -\frac{\sigma}{4\epsilon\Delta t}((c^{n+1} - c^n)^4, 1)_h. \end{aligned} \tag{2.2.27}$$

Proof. Multiply Eq. (2.2.20) by $\mathbf{u}^{n+\frac{1}{2}}$ and sum over cell-centers to get

$$\begin{aligned}\frac{(\mathbf{u}^{n+1},\mathbf{u}^{n+1})_h-(\mathbf{u}^n,\mathbf{u}^n)_h}{2\Delta t} &= -(\mathbf{u}^{n+\frac{1}{2}},\tilde{\nabla}_d^c P^{n+\frac{1}{2}})_h-\frac{\sigma}{\epsilon}(\mathbf{u}^{n+\frac{1}{2}}c^{n+\frac{1}{2}},\nabla_d\mu^{n+\frac{1}{2}})_h \\ &\quad +\eta(\mathbf{u}^{n+\frac{1}{2}},\Delta_d\mathbf{u}^{n+\frac{1}{2}})_h \qquad (2.2.28)\\ &= (\nabla_d^c\cdot\mathbf{u}^{n+\frac{1}{2}},P^{n+\frac{1}{2}})_{h,\frac{1}{2}}+\frac{\sigma}{\epsilon}(\nabla_d\cdot\left(\mathbf{u}^{n+\frac{1}{2}}c^{n+\frac{1}{2}}\right),\mu^{n+\frac{1}{2}})_h \\ &\quad -\eta\left(|u^{n+\frac{1}{2}}|_{e,1}^2+|v^{n+\frac{1}{2}})|_{e,1}^2\right)\end{aligned}$$

where we have used lemma 2.1 to sum by parts. Next, multiply Eq. (2.2.14) by $\mu^{n+\frac{1}{2}}$ and sum to get

$$\begin{aligned}(c^{n+1}-c^n,\mu^{n+\frac{1}{2}})_h &= -\Delta t(\nabla_d\cdot(c^{n+\frac{1}{2}}\mathbf{u}^{n+\frac{1}{2}}),\mu^{n+\frac{1}{2}})_h+\Delta t(\Delta_d\mu^{n+\frac{1}{2}},\mu^{n+\frac{1}{2}})_h \\ &= -\Delta t(\nabla_d\cdot(c^{n+\frac{1}{2}}\mathbf{u}^{n+\frac{1}{2}}),\mu^{n+\frac{1}{2}})_h-\Delta t|\mu^{n+\frac{1}{2}}|_{e,1}^2. \qquad (2.2.29)\end{aligned}$$

Combining Eqs. (2.2.28) and (2.2.29) and using the argument in the proof of lemma 2.2, we obtain

$$\begin{aligned}\mathcal{E}_h(c^{n+1},\mathbf{u}^{n+1})-\mathcal{E}_h(c^n,\mathbf{u}^n) &= -\eta\Delta t\left(|u^{n+\frac{1}{2}}|_{e,1}^2+|v^{n+\frac{1}{2}}|_{e,1}^2\right)-\frac{\sigma\Delta t}{\epsilon}|\mu^{n+\frac{1}{2}}|_{e,1}^2 \qquad (2.2.30)\\ &\quad -\frac{\sigma}{4\epsilon}((c^{n+1}-c^n)^4,1)_h+\Delta t(\nabla_d^c\cdot\mathbf{u}^{n+\frac{1}{2}},P^{n+\frac{1}{2}})_{h,\frac{1}{2}},\end{aligned}$$

where we have used that the terms that couple flow with concentration in the equations cancel one another exactly. Since we have assumed that the velocity is discretely divergence-free from Eq. (2.2.19), this completes the proof of the lemma.

□

From lemma 2.3, it follows that both $||c^n||$ and $||\mathbf{u}^n||$ are bounded uniformly in n. This yields stability (in l^2) of the numerical scheme in the presence of flow.

Using the discretization (2.2.20), the lemma also holds if the viscosity η is not constant.

Next, we demonstrate the convergence of the scheme at a fixed time. Let C^n and c^n be the continuous and discrete solutions at time $t=t_n$, respectively and let $e^n=C^n-c^n$ be the error. Then we have the following error estimate.

Theorem 2.1. *Suppose C is smooth. Then, for any $T>0$, there exists a constant K, Δt_0, and h_0 depending on $T,\phi,\hat{\phi},\epsilon$, and smoothness of C such that the following error estimate holds:*

$$||e^n||\le K(h^2+\Delta t^2)$$

for $n\Delta t\le T$ if $h\le h_0$ and $\Delta t\le\Delta t_0$.

Proof. Using the numerical scheme, we obtain

$$\begin{aligned}\partial_t e^m + \epsilon^2 \Delta_d^2 e^{m+\frac{1}{2}} &= \partial_t C^m + \epsilon^2 \Delta_d^2 C^{m+\frac{1}{2}} - \Delta_d \hat{\phi}(c^m, c^{m+1}) \\ &= C_t(t_{m+\frac{1}{2}}) + \epsilon^2 \Delta^2 C(t_{m+\frac{1}{2}}) - \Delta_d \hat{\phi}(c^m, c^{m+1}) + O(h^2 + \Delta t^2) \\ &= \Delta_d \phi(C^{m+\frac{1}{2}}) - \Delta_d \phi(c^m, c^{m+1}) + O(h^2 + \Delta t^2). \end{aligned} \tag{2.2.31}$$

From now on, the capital letter K will be used to denote the generic constant, the value of which may change from line to line. Adding and subtracting $\Delta_d \phi(c^{m+\frac{1}{2}})$ in (2.2.31), we obtain

$$\partial_t e^m + \epsilon^2 \Delta_d^2 e^{m+\frac{1}{2}} = A + B + O(h^2 + \Delta t^2),$$

where A and B are defined as follows:

$$A \equiv \phi(C^{m+\frac{1}{2}}) - \phi(c^{m+\frac{1}{2}}), \qquad B \equiv \phi(c^{m+\frac{1}{2}}) - \hat{\phi}(c^m, c^{m+1}).$$

Multiplying $e^{m+\frac{1}{2}}$ and using summation by parts, we have

$$\begin{aligned}\tfrac{1}{2}\partial_t ||e^m||^2 + \epsilon^2 ||\Delta_d e^{m+\frac{1}{2}}||^2 &\leq (A, \Delta_d e^{m+\frac{1}{2}})_h + (B, \Delta_d e^{m+\frac{1}{2}})_h \\ &\quad + K(h^4 + \Delta t^4) + ||e^{m+\frac{1}{2}}||^2. \end{aligned} \tag{2.2.32}$$

We first consider the first term of the right side of (2.2.32). Since $||u^n||$ and $||c^n||$ are bounded, we obtain

$$(A, \Delta_d e^{m+\frac{1}{2}})_h \leq K(|e^{m+\frac{1}{2}}|, |\Delta_d e^{m+\frac{1}{2}}|)_h \leq K||e^{m+\frac{1}{2}}||^2 + \frac{\epsilon^2}{4}||\Delta_d e^{m+\frac{1}{2}}||^2.$$

It remains to estimate the second term. A simple computation shows that B is factored as follows:

$$B = \frac{1}{8}(c^{m+1} - c^m)^2(3c^{m+1} - c^m - 1).$$

With the aid of the factorization and Young's inequality we obtain

$$(B, \Delta_d e^{m+\frac{1}{2}})_h \leq K||B||^2 + \frac{\epsilon^2}{4}||\Delta_d e^{m+\frac{1}{2}}||^2 \leq K||(c^{m+1} - c^m)^2||^2 + \frac{\epsilon^2}{4}||\Delta_d e^{m+\frac{1}{2}}||^2,$$

where we used the fact that numerical solution is bounded. The next step is to estimate $||(c^{m+1} - c^m)^2||^2$. Adding and subtracting the analytic solution, we have

$$\begin{aligned}||(c^{m+1} - c^m)^2||^2 &\leq 2(||(c^{m+1} - c^m)^2 - (C^{m+1} - C^m)^2||^2 + ||(C^{m+1} - C^m)^2||^2) \\ &\leq K(||e^{m+1} - e^m||^2 + ||(C^{m+1} - C^m)^2||^2),\end{aligned}$$

where again the boundedness of the numerical and analytical solutions is used. Since analytic solution u is smooth, the second term is estimated as follows:

$$||(C^{m+1} - C^m)^2||^2 \leq K\Delta t^4 ||C_t||_\infty^4.$$

Putting everything together, we get

$$(B, \Delta_d e^{m+\frac{1}{2}})_h \leq K\Delta t^4 + K||e^{m+1} - e^m||^2 + \frac{\epsilon^2}{4}||\Delta_d e^{m+\frac{1}{2}}||^2.$$

Subtracting $\frac{\epsilon^2}{2}||\Delta_d e^{m+\frac{1}{2}}||_2^2$ and multiplying 2 to both sides in (2.2.32), we obtain

$$\partial_t||e^m||^2 + \epsilon^2||\Delta_d e^{m+\frac{1}{2}}||^2 \leq K(h^4 + \Delta t^4) + K||e^{m+\frac{1}{2}}||^2 + K||e^{m+1} - e^m||^2. \tag{2.2.33}$$

Dropping $\epsilon^2||\Delta_d e^{m+\frac{1}{2}}||^2$ in (2.2.33) and summing in time, we have

$$\begin{aligned} ||e^n||^2\Delta t^{-1} &\leq \sum_{m=0}^{n-1}[K(h^4 + \Delta t^4) + K||e^{m+\frac{1}{2}}||^2 + K||e^{m+1} - e^m||^2] \\ &\leq \sum_{m=0}^{n-1}[K(h^4 + \Delta t^4) + K||e^{m+1}||^2 + K||e^m||^2] \\ &= Kn(h^4 + \Delta t^4) + K\sum_{m=0}^{n-1}||e^m||^2 + K||e^n||. \end{aligned}$$

Multiplying Δt on both sides, we obtain

$$\begin{aligned} (1 - K\Delta t)||e^n||^2 &\leq K(\Delta t n)(h^4 + \Delta t^4) + K\Delta t\sum_{m=0}^{n-1}||e^m||^2 \\ &\leq KT(h^4 + \Delta t^4) + K\Delta t\sum_{m=0}^{n-1}||e^m||^2 \end{aligned}$$

where we used the fact that $n\Delta t \leq T$. Since Δt can be chosen such that $1 - K\Delta t > 0$, the discrete version of Gronwall's inequality enables us to obtain $||e^n|| \leq K(h^2 + \Delta t^2)$. This completes the proof.

□

2.3 Solution of the system in absence of flow

In this section, we develop a nonlinear Full Approximation Storage (**FAS**) multigrid method to solve the nonlinear discrete system at the implicit time level. The fundamental idea of nonlinear multigrid is analogous to the linear case. First, the errors to the solution have to be smoothed so that they can be approximated on a coarser grid. An analog of the linear defect equation is transformed to the coarse grid. The coarse grid corrections are interpolated back to the fine grid, where the errors are again smoothed. However, because the system is nonlinear formally we do not work with the errors, but rather with full approximations to the discrete solution on the coarse grid. The nonlinearity is treated using one step of Newton's iteration and a pointwise Gauss-Seidel relaxation scheme is used as the smoother in the

multigrid method. This corresponds to a local rather than global linearization of the nonlinear scheme and as such is more efficient than standard Newton-Gauss-Seidel global linearization schemes. See the reference text [125] for additional details and background.

Let us rewrite equations (2.2.14), (2.2.15) as follows.

$$\mathbf{NSO}(c^n, c^{n+1}, \mu^{n+\frac{1}{2}}) = (f^n, g^n),$$

where

$$\mathbf{NSO}(c^n, c^{n+1}, \mu^{n+\frac{1}{2}}) = (\frac{c_{ij}^{n+1}}{\Delta t} - \Delta_d \mu_{ij}^{n+\frac{1}{2}}, \mu_{ij}^{n+\frac{1}{2}} - \hat{\phi}(c_{ij}^n, c_{ij}^{n+1}) + \frac{\epsilon^2}{2}\Delta_d c_{ij}^{n+1}),$$

and the source term is

$$(f^n, g^n) = (\frac{c_{ij}^n}{\Delta t}, -\frac{\epsilon^2}{2}\Delta_d c_{ij}^n).$$

In the following description of one **FAS** cycle, we assume a sequence of grids Ω_k (Ω_{k-1} is coarser than Ω_k by factor 2). Given the number ν of pre- and post- smoothing relaxation sweeps, an iteraton step for the nonlinear multigrid method using the V-cycle is formally written as follows:

FAS multigrid cycle

$$\{c_k^{m+1}, \mu_k^{m+1/2}\} = FAScycle(k, c_k^n, c_k^m, \mu_k^{m-1/2}, \mathbf{NSO}_k, f_k^n, g_k^n, \nu).$$

That is, $\{c_k^m, \mu_k^{m-1/2}\}$ and $\{c_k^{m+1}, \mu_k^{m+1/2}\}$ are the approximations of $c_k(x_i, y_j)$ and $\mu_k(x_i, y_j)$ before and after an FAScycle. Now, define the FAScycle.

(1) Presmoothing

Compute $\{\bar{c}_k^m,\ \bar{\mu}_k^{m-1/2}\}$ by applying ν smoothing steps to $\{c_k^m,\ \mu_k^{m-1/2}\}$

$$\{\bar{c}_k^m,\ \bar{\mu}_k^{m-1/2}\} = SMOOTH^\nu(c_k^n, c_k^m, \mu_k^{m-1/2}, \mathbf{NSO}_k, f_k^n, g_k^n),$$

which means performing ν smoothing steps with initial approximation $c_k^m, \mu_k^{m-1/2}$, c_k^n, source terms f_k^n, g_k^n, and $SMOOTH$ relaxation operator (see Appendix 8.A for its derivation) to get the approximation $\bar{c}_k^m,\ \bar{\mu}_k^{m-1/2}$.

One $SMOOTH$ relaxation operator step consists of solving the system (2.3.34) and (2.3.35) given below by 2×2 matrix inversion for each i and j.

$$\frac{\bar{c}_{ij}^m}{\Delta t} + (\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2})\bar{\mu}_{ij}^{m-\frac{1}{2}} = \frac{\mu_{i+1,j}^{m-\frac{1}{2}} + \bar{\mu}_{i-1,j}^{m-\frac{1}{2}}}{\Delta x^2} + \frac{\mu_{i,j+1}^{m-\frac{1}{2}} + \bar{\mu}_{i,j-1}^{m-\frac{1}{2}}}{\Delta y^2} + f_{ij}^n \tag{2.3.34}$$

and

$$\begin{aligned} -[\frac{\epsilon^2}{2}(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}) \quad &+ \quad \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial v}]\bar{c}_{ij}^m + \bar{\mu}_{ij}^{m-\frac{1}{2}} \\ &= \quad g_{ij}^n + \hat{\phi}(c_{ij}^n, c_{ij}^m) - \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial v} c_{ij}^m \\ &\quad -\frac{\epsilon^2}{2\Delta x^2}(c_{i+1,j}^m + \bar{c}_{i-1,j}^m) - \frac{\epsilon^2}{2\Delta y^2}(c_{i,j+1}^m + \bar{c}_{i,j-1}^m). \end{aligned} \tag{2.3.35}$$

(2) Compute the defect

$$(\bar{d_1}_k^m, \bar{d_2}_k^m) = (f_k^n, g_k^n) - \mathbf{NSO}_k(\bar{c}_k^n, \bar{c}_k^m, \bar{\mu}_k^{m-1/2}).$$

(3) Restrict the defect and $\{\bar{c}_k^m,\ \bar{\mu}_k^{m-1/2}\}$

$$\begin{aligned} (\bar{d_1}_{k-1}^m, \bar{d_2}_{k-1}^m) &= I_k^{k-1}(\bar{d_1}_k^m, \bar{d_2}_k^m), \\ (\bar{c}_{k-1}^m, \bar{\mu}_{k-1}^{m-1/2}) &= I_k^{k-1}(\bar{c}_k^m, \bar{\mu}_k^{m-1/2}). \end{aligned}$$

The restriction operator I_k^{k-1} maps k-level functions to $(k-1)$-level functions.

$$\begin{aligned} d_{k-1}(x_i, y_j) = I_k^{k-1} d_k(x_i, y_j) = \frac{1}{4} \big[& d_k(x_i - h/2, y_j - h/2) + d_k(x_i - h/2, y_j + h/2) \\ & + d_k(x_i + h/2, y_j - h/2) + d_k(x_i + h/2, y_j + h/2) \big]. \end{aligned}$$

That is, coarse grid values are obtained by averaging the four nearby fine grid values.

(4) Compute the right-hand side

$$(f_{k-1}^n, g_{k-1}^n) = (\bar{d_1}_{k-1}^m, \bar{d_2}_{k-1}^m) + \mathbf{NSO}_{k-1}(\bar{c}_{k-1}^n, \bar{c}_{k-1}^m, \bar{\mu}_{k-1}^{m-1/2}).$$

(5) Compute an approximate solution $\{\hat{c}_{k-1}^m,\ \hat{\mu}_{k-1}^{m-1/2}\}$ **of the coarse grid equation on** Ω_{k-1}**, i.e.**

$$\mathbf{NSO}_{k-1}(c_{k-1}^n, c_{k-1}^m, \mu_{k-1}^{m-1/2}) = (f_{k-1}^n, g_{k-1}^n). \tag{2.3.36}$$

If $k = 1$, we explicitly invert a 2×2 matrix to obtain the solution. If $k > 1$, we solve (2.3.36) by performing a **FAS** k-grid cycle using $\{\bar{c}_{k-1}^m,\ \bar{\mu}_{k-1}^{m-1/2}\}$ as an initial approximation:

$$\{\hat{c}_{k-1}^m, \hat{\mu}_{k-1}^{m-1/2}\} = \text{FAScycle}(k-1, c_{k-1}^n, \bar{c}_{k-1}^m, \bar{\mu}_{k-1}^{m-1/2}, \mathbf{NSO}_{k-1}, f_{k-1}^n, g_{k-1}^n, \nu).$$

(6) Compute the coarse grid correction (CGC):

$$\begin{aligned} \hat{v_1}_{k-1}^{\,m} &= \hat{c}_{k-1}^m - \bar{c}_{k-1}^m. \\ \hat{v_2}_{k-1}^{\,m-1/2} &= \hat{\mu}_{k-1}^{m-1/2} - \bar{\mu}_{k-1}^{m-1/2}. \end{aligned}$$

(7) Interpolate the correction

$$\begin{aligned} \hat{v_1}_{k}^{\,m} &= I_{k-1}^k \hat{v_1}_{k-1}^{\,m}. \\ \hat{v_2}_{k}^{\,m-1/2} &= I_{k-1}^k \hat{v_2}_{k-1}^{\,m-1/2}. \end{aligned}$$

The interpolation operator I_{k-1}^k maps (k-1)-level functions to k-level functions. Here, the coarse values are simply transferred to the four nearby fine grid points, i.e. $v_k(x_i, y_j) = I_{k-1}^k v_{k-1}(x_i, y_j) = v_{k-1}(x_i + h/2, y_j + h/2)$ for i and j odd-numbered integers. The values at the other node points are given by

$$v_k(x_i + h, y_j) = v_k(x_i, y_j + h) = v_k(x_i + h, y_j + h) = v_{k-1}(x_i + h/2, y_j + h/2),$$

where i and j are odd.

(8) Compute the corrected approximation on Ω_k

$$\begin{aligned} c_k^{m,\text{ after } CGC} &= \bar{c}_k^m + \hat{v_1}_k^{\,m}. \\ \mu_k^{m-1/2,\text{ after } CGC} &= \bar{\mu}_k^{m-1/2} + \hat{v_2}_k^{\,m-1/2}. \end{aligned}$$

(9) Postsmoothing

Compute $\{c_k^{m+1}, \mu_k^{m+1/2}\}$ by applying ν smoothing steps to $c_k^{m,\text{ after } CGC}$, $\mu_k^{m-1/2,\text{ after } CGC}$

$$\{c_k^{m+1}, \mu_k^{m+1/2}\} = SMOOTH^{\nu}(c_k^n, c_k^{m,\text{ after } CGC}, \mu_k^{m-1/2,\text{ after } CGC}, \mathbf{NSO}_k, f_k^n, g_k^n).$$

This completes the description of a nonlinear FAScycle. We next turn to the solution of the discrete system in the presence of flow.

2.4 Solution of system in presence of flow

Here, we present an iterative projection method for solving the coupled discrete NSCH system (2.2.15), (2.2.18) and (2.2.19)- (2.2.20). For simplicity, we focus on the case in which η is constant. Let $u^{*,k+1}$ be an intermediate velocity field and satisfy the following system for $k \geq 0$:

$$\begin{aligned}\frac{\mathbf{u}^{*,k+1}-\mathbf{u}^{n}}{\Delta t}+\frac{1}{2}\omega^{n+\frac{1}{2},k}\times(\mathbf{u}^{n+1,k}+\mathbf{u}^{n}) &= -\tilde{\nabla}_{d}^{c}P^{n-\frac{1}{2}}+\frac{\nu}{2}\Delta_{d}(\mathbf{u}^{*,k+1}+\mathbf{u}^{n}) \\ &\quad -\frac{\sigma}{2\epsilon}\left(c^{n+1,k}+c^{n}\right)\nabla_{d}\mu^{n+\frac{1}{2},k}, \qquad (2.4.37)\end{aligned}$$

where

$$\begin{aligned}&\frac{1}{2}\left(\omega^{n+\frac{1}{2},k}\times\left(\mathbf{u}^{n+\frac{1}{2},k}+\mathbf{u}^{n}\right)\right)_{ij}= \\ &\qquad\left(-\frac{\bar{v}_{i+1,j}-\bar{v}_{i-1,j}-\bar{u}_{i,j+1}+\bar{u}_{i,j-1}}{2h}\bar{v}_{ij},\frac{\bar{v}_{i+1,j}-\bar{v}_{i-1,j}-\bar{u}_{i,j+1}+\bar{u}_{i,j-1}}{2h}\bar{u}_{ij}\right),\end{aligned}$$

and $(\bar{u},\bar{v})=(\mathbf{u}^{n+\frac{1}{2},k}+\mathbf{u}^{n})/2$. The initial step is taken to be $\mathbf{u}^{n+1,0}=\mathbf{u}^{n}$. Note that the advection, pressure gradient and surface force terms are treated as forcing functions. Following [125], Eq. (2.4.37) is solved for $\mathbf{u}^{*,k+1}$ using a linear multigrid method. For completeness, this scheme is given in Appendix 8.B.

The velocity field $\mathbf{u}^{*,k+1}$ is not, in general, divergence-free. The projection step of the algorithm decomposes the intermediate velocity into a discrete gradient of a scalar potential and a divergence-free (or approximately divergence-free) vector field. They correspond to the pressure gradient and to the velocity updates, respectively. In particular, if $\mathbf{P}$ represents the projection operator, then

$$\frac{\mathbf{u}^{n+1,k+1}-\mathbf{u}^{n}}{\Delta t}=\mathbf{P}\left(\frac{\mathbf{u}^{*,k+1}-\mathbf{u}^{n}}{\Delta t}\right)\equiv\frac{\mathbf{u}^{*,k+1}-\mathbf{u}^{n}}{\Delta t}-\tilde{\nabla}_{d}^{c}\psi \qquad (2.4.38)$$

where ψ is obtained either by

$$\nabla_{d}^{c}\cdot\tilde{\nabla}_{d}^{c}\psi=\nabla_{d}^{c}\cdot\left(\frac{\mathbf{u}^{*,k+1}-\mathbf{u}^{n}}{\Delta t}\right) \qquad (2.4.39)$$

for exact projection, i.e. $\nabla_{d}^{c}\cdot\mathbf{u}^{n+1,k+1}=0$, or by

$$\Delta_{d}^{c}\psi=\nabla_{d}^{c}\cdot\left(\frac{\mathbf{u}^{*,k+1}-\mathbf{u}^{n}}{\Delta t}\right) \qquad (2.4.40)$$

for an approximate projection, i.e.

$$\nabla_{d}^{c}\cdot\left(\frac{\mathbf{u}^{n+1,k+1}-\mathbf{u}^{n}}{\Delta t}\right)=\left(\Delta_{d}^{c}-\nabla_{d}^{c}\cdot\tilde{\nabla}_{d}^{c}\right)\psi. \qquad (2.4.41)$$

The operators Δ_{d}^{c} and $\nabla_{d}^{c}\cdot\tilde{\nabla}_{d}^{c}$ are defined by

$$\Delta_{d}^{c}\psi_{i+\frac{1}{2},j+\frac{1}{2}}=\left(\psi_{i-\frac{1}{2},j+\frac{1}{2}}+\psi_{i+\frac{1}{2},j+\frac{3}{2}}-4\psi_{i+\frac{1}{2},j+\frac{1}{2}}+\psi_{i+\frac{1}{2},j-\frac{1}{2}}+\psi_{i+\frac{3}{2},j+\frac{1}{2}}\right)/h^{2},$$

and

$$\nabla_d^c \cdot \tilde{\nabla}_d^c \psi_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\psi_{i-\frac{1}{2},j-\frac{1}{2}} + \psi_{i+\frac{3}{2},j+\frac{1}{2}} - 4\psi_{i+\frac{1}{2},j+\frac{1}{2}} + \psi_{i-\frac{1}{2},j-\frac{3}{2}} + \psi_{i+\frac{3}{2},j+\frac{3}{2}}\right)/2h^2.$$

In the case of exact projection, a conjugate gradient method is used to solve the discrete equations. This works well for low resolutions. However, the stencil is algebraically decoupled (e.g. see [96]) and the system is ill-conditioned so that the number of iterations necessary to solve the equation grows dramatically with increasing numbers of grid points. The approximate projection, however, is a more robust alternative [96] and here we use a linear multigrid method to solve the discrete equation. Similiar multigrid methods have been used in 3D by [1] and by [6] in 2D in a finite element context. The multigrid algorithm we use is analogous to that presented in Appendix 8.B for the intermediate velocity. In the case of approximate projection, we use the Gauss-Seidel smoothing operator on grid level k, i.e. $\bar{\psi}_k^m = SMOOTH^\nu(\psi_k^m, L_k, f_k)$ with L_k given by the Laplacian operator Δ_d^c on the k grid level, is

$$\bar{\psi}^m_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left(\bar{\psi}^m_{i-\frac{1}{2},j+\frac{1}{2}} + \psi^m_{i+\frac{1}{2},j+\frac{3}{2}} + \bar{\psi}^m_{i+\frac{1}{2},j-\frac{1}{2}} + \psi^m_{i+\frac{3}{2},j+\frac{1}{2}}\right) - \frac{h^2}{4} f_{i+\frac{1}{2},j+\frac{1}{2}}$$

where the forcing term is

$$f_{i+\frac{1}{2},j+\frac{1}{2}} = \nabla_d^c \cdot \left(\frac{\mathbf{u}_{ij}^{*,k+1} - \mathbf{u}_{ij}^n}{\Delta t}\right).$$

In section 2.5, we analyze both projection algorithms and demonstrate the ill-conditioning of the exact projection method. In the case of approximate projection, the analysis shows that the multigrid method convergence rate is insensitive to the grid size.

We next update the concentration field by

$$\frac{c^{n+1,k+1} - c^n}{\Delta t} + \frac{1}{4}\nabla_d \cdot \left((\mathbf{u}^{n+1,k+1} + \mathbf{u}^n)(c^{n+1,k} + c^n)\right) = \Delta_d \mu^{n+\frac{1}{2},k+1}, \tag{2.4.42}$$

where

$$\mu^{n+\frac{1}{2},k+1} = \hat{\phi}(c^n, c^{n+1,k+1}) - \frac{\epsilon^2}{2}\Delta_d(c^n + c^{n+1,k+1}). \tag{2.4.43}$$

This system is solved using the non-linear multigrid method presented in section 2.3 where the source term f^n is modified to account for the advection.

The above system (2.4.37)-(2.4.38) and (2.4.42)-(2.4.43) is iterated in k until the difference between successive iterates is less than an error tolerance. Finally, the pressure is then updated using the second-order accurate scheme [97]:

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \psi - \frac{\nu\Delta t}{2}\Delta_d^c \psi.$$

2.5 Local Fourier Analysis

5.A Absence of flow

To analyze the behavior of the multigrid method, we linearize the nonlinear scheme and perform a local Fourier analysis (e.g. see [125]). In particular, we analyze the smoother since the performance of the multigrid method depends strongly on the smoother.

Let $c^{n+1},\ \mu^{n+1/2}$ be the solution of

$$\frac{c_{ij}^{n+1}-c_{ij}^{n}}{\Delta t}=\Delta_d \mu_{ij}^{n+\frac{1}{2}}, \tag{2.5.44}$$

$$\mu_{ij}^{n+\frac{1}{2}}=\hat{\phi}(c_{ij}^{n},c_{ij}^{n+1})-\frac{\epsilon^2}{2}\Delta_d(c_{ij}^{n+1}+c_{ij}^{n}). \tag{2.5.45}$$

After linearizing the nonlinear term $\hat{\phi}(c_{ij}^{n},c_{ij}^{n+1})=\frac{\alpha}{2}(c_{ij}^{n}+c_{ij}^{n+1})+\beta$, where $\alpha=\phi'(c_m),\ c_m$ is an average concentration, and β is a constant and substituting $\mu_{ij}^{n+\frac{1}{2}}$ into (2.5.44), the scheme becomes

$$L_h c_h^{n+1}=f_h^{n},$$

where

$$\begin{aligned}
L_h c_h^{n+1} \quad := \quad & \frac{c_{ij}^{n+1}}{\Delta t}-\frac{\alpha}{2h^2}(c_{i-1,j}^{n+1}+c_{i,j-1}^{n+1}-4c_{ij}^{n+1}) \\
& +\frac{\epsilon^2}{2h^4}[c_{i-2,j}^{n+1}+c_{i,j-2}^{n+1}+2(c_{i-1,j+1}^{n+1}+c_{i-1,j-1}^{n+1}) \\
& -8(c_{i-1,j}^{n+1}+c_{i,j-1}^{n+1})+20c_{i,j}^{n+1}]-\frac{\alpha}{2h^2}(c_{i+1,j}^{n+1}+c_{i,j+1}^{n+1}) \\
& +\frac{\epsilon^2}{2h^4}[c_{i+2,j}^{n+1}+c_{i,j+2}^{n+1}+2(c_{i+1,j+1}^{n+1}+c_{i+1,j-1}^{n+1})-8(c_{i+1,j}^{n+1}+c_{i,j+1}^{n+1})]
\end{aligned}$$

and

$$f_h^{n}=\frac{\alpha}{2}\Delta_d c_{ij}^{n}-\frac{\epsilon^2}{2}\Delta_d^2 c_{ij}^{n}+\frac{c_{ij}^{n}}{\Delta t}.$$

For Gauss-Seidel iteration with a lexicographic ordering of the grid points applied to the above equation, we have the following operator decomposition.

$$\begin{aligned}
L_h^{+} c_h^{n+1} \quad := \quad & \frac{c_{ij}^{n+1}}{\Delta t}-\frac{\alpha}{2h^2}(c_{i-1,j}^{n+1}+c_{i,j-1}^{n+1}-4c_{ij}^{n+1})+\frac{\epsilon^2}{2h^4}[c_{i-2,j}^{n+1}+c_{i,j-2}^{n+1} \\
& +2(c_{i-1,j+1}^{n+1}+c_{i-1,j-1}^{n+1})-8(c_{i-1,j}^{n+1}+c_{i,j-1}^{n+1})+20c_{i,j}^{n+1}], \\
L_h^{-} c_h^{n+1} \quad := \quad & -\frac{\alpha}{2h^2}(c_{i+1,j}^{n+1}+c_{i,j+1}^{n+1})+\frac{\epsilon^2}{2h^4}[c_{i+2,j}^{n+1}+c_{i,j+2}^{n+1}+2(c_{i+1,j+1}^{n+1}+c_{i+1,j-1}^{n+1}) \\
& -8(c_{i+1,j}^{n+1}+c_{i,j+1}^{n+1})].
\end{aligned}$$

Therefore, this relaxation method (which is the linear analogue of the nonlinear multigrid smoother) can be written *locally* as

$$L_h^{+}\tilde{z}_h+L_h^{-}z_h=f_h^{n}, \tag{2.5.46}$$

where z_h corresponds to the old approximation of c_h (approximation before the relaxation step) and $\tilde{z}_h$ to the new approximation (after the step). Subtracting (2.5.46) from the discrete equation $L_h c_h = f_h$ and letting $\tilde{v}_h = c_h - \tilde{z}_h$ and $v_h = c_h - z_h$, we obtain the equation

$$L_h^+ \tilde{v}_h + L_h^- v_h = 0,$$

or, equivalently,

$$\tilde{v}_h = S_h v_h,$$

where $S_h = -(L_h^+)^{-1} L_h^-$ is the resulting smoothing operator. Applying L_h^+ and L_h^- to the formal eigenfunctions $e^{i\theta_1 x/h} e^{i\theta_2 y/h}$, we obtain

$$L_h^+ e^{i\theta_1 x/h} e^{i\theta_2 y/h} = \hat{L}_h^+ e^{i\theta_1 x/h} e^{i\theta_2 y/h},$$

$$L_h^- e^{i\theta_1 x/h} e^{i\theta_2 y/h} = \hat{L}_h^- e^{i\theta_1 x/h} e^{i\theta_2 y/h},$$

where $\hat{L}_h^+$ and $\hat{L}_h^-$ are the symbols of the operators L_h^+ and L_h^-, respectively:

$$\begin{aligned}
\hat{L}_h^+(\theta_1,\theta_2) = \quad & \frac{1}{\Delta t} - \frac{\alpha}{2h^2}(e^{-i\theta_1} + e^{-i\theta_2} - 4) + \frac{\epsilon^2}{2h^4}[e^{-2i\theta_1} + e^{-2i\theta_2} \\
& + 2(e^{-i(\theta_1-\theta_2)} + e^{-i(\theta_1+\theta_2)}) - 8(e^{-i\theta_1} + e^{-i\theta_2}) + 20], \\
\hat{L}_h^-(\theta_1,\theta_2) = \quad & -\frac{\alpha}{2h^2}(e^{i\theta_1} + e^{i\theta_2}) + \frac{\epsilon^2}{2h^4}[e^{2i\theta_1} + e^{2i\theta_2} \\
& + 2(e^{i(\theta_1+\theta_2)} + e^{i(\theta_1-\theta_2)}) - 8(e^{i\theta_1} + e^{i\theta_2})].
\end{aligned}$$

The amplification factor of the relaxation scheme is

$$\hat{S}_h(\theta_1,\theta_2) := -\frac{\hat{L}_h^-(\theta_1,\theta_2)}{\hat{L}_h^+(\theta_1,\theta_2)}.$$

Define the high frequency smoothing (HFS) factor:

$$\mu_{loc}(S_h) := \sup\{|\tilde{S}_h(\theta_1,\theta_2)| : \ \frac{\pi}{2} \le |\theta_1|, \ |\theta_2| \le \pi\}.$$

Here, as is typically done [125], we assume that the coarse grid operations are ideal and annihilate the low frequency error components while leaving the high frequency components unchanged. Therefore, we only consider $\frac{\pi}{2} \le |\theta_1|, \ |\theta_2| \le \pi$. We define a convergence factor as an average of the quantity $||d_h^m||/||d_h^{m-1}||$, where d_h^m $(m = 1, 2, \cdots)$ are the defects.

The convergence factor is estimated numerically using our nonlinear code with the parameters $\epsilon = 0.01$, and the mesh-dependent time step $\Delta t = 0.1h$ and initial conditions

$$c_0(x,y) = \begin{cases} 0.0 + 0.01\cos(0.5\pi x/h)\cos(0.5\pi y/h), & \alpha = 1/4 \\ 0.5 + 0.01\cos(0.5\pi x/h)\cos(0.5\pi y/h), & \alpha = -1/4 \end{cases}.$$

We measure the $V(m,n)$- HFS factors, where m and n are the numbers of pre-smoothing and post-smoothing, with different mesh sizes. We focus on $m = 1$ and $n = 0$ and 1 as these yield the most efficient algorithms. In addition, we consider $\alpha = \pm 1/4$ where the positive (negative) values correspond to linearization in the stable (unstable, i.e. spinodal region) ranges of the evolution.

Table 2.1 shows HFS factors and measured V(m,n)-cycle convergence factors with different mesh sizes and $\alpha = 0.25$. Note $\sqrt{V(1,1)}$-cycle means the square root of V(1,1)-cycle convergence factor. Observe that with $m = 1$ and $n = 0$, the HFS factor tends to 1 as the mesh is refined and thus suggests that the number of V-cycles required to solve the full system increases with increasing resolution. In fact this result is comparable to simply using Gauss-Siedel iteration without multigrid. However, the HFS factor corresponding to $m = 1$ and $n = 1$ remains uniformly bounded below 1 with increasing resolution and apparently converges to 0.4196 as $h \to 0$. This is significantly below the theoretical estimate 0.6694. Thus the number of $V(1,1)$-cycles required to solve the full problem is insensitive to the resolution.

The HFS factors for $\alpha = -0.25$ are given in Table 2.2. Their behavior is analogous to that observed for the $\alpha = 0.25$. Note that the HFS factor for $V(1,0)$-cycles is actually greater than 1 at the finest resolution.

Table 2.1: HFS factors for different mesh sizes. $\alpha = 0.25$, $\Delta t = 0.1h$ and $h = 1/N$

Case	16 x 16	32 x 32	64 x 64	128 x 128
μ_{loc}	0.0363	0.2839	0.5460	0.6376
V(1,0)-cycle	0.1643	0.2108	0.2865	0.3489
$\sqrt{V(1,1)}$-cycle	0.2108	0.2462	0.2840	0.3434
Case	256 x 256	512 x 512	1024 x 1024	2048 x 2048
μ_{loc}	0.6616	0.6676	0.6690	0.6694
V(1,0)-cycle	0.4636	0.6563	0.8198	0.9810
$\sqrt{V(1,1)}$-cycle	0.4021	0.4171	0.4196	0.4196

Table 2.2: HFS factors for different mesh sizes. $\alpha = -0.25$, $\Delta t = 0.1h$ and $h = 1/N$

Case	16 x 16	32 x 32	64 x 64	128 x 128
μ_{loc}	1.3773	1.1563	0.7874	0.6992
V(1,0)-cycle	0.5183	0.5837	0.4245	0.4829
$\sqrt{V(1,1)}$-cycle	0.5501	0.6061	0.3706	0.4162
Case	256 x 256	512 x 512	1024 x 1024	2048 x 2048
μ_{loc}	0.6771	0.6714	0.6700	0.6697
V(1,0)-cycle	0.6260	0.7904	0.8719	N/A
$\sqrt{V(1,1)}$-cycle	0.4141	0.4209	0.4299	0.4382

Together, these results for $\alpha = \pm 0.25$ suggest that the multigrid method using a $V(1,1)$-cycle with time step $\Delta t \sim h$ converges uniformly with respect to increasing resolution. Correspondingly, this would impose a first order time step constraint on our discrete scheme to solve the (**C-H**) equation.

Next, let us consider results obtained using a *fixed* $\Delta t = 0.005$ independent of the mesh size h. All other parameters are as before. This value of Δt roughly corresponds to that used in Tables 2.1 and 2.2 with $h = 1/16$. Table 2.3 shows the $V(1,1)$-HFS factors with different mesh sizes and $\alpha = 0.25$. Again, the results are qualitatively similar to the case with variable $\Delta t \sim h$. When $\alpha = -0.25$, however, the results are very different. This is seen in Table 2.4. The numerical HFS factor now tends to 1 as the mesh is refined in spite of the fact that theoretical HFS factors are comparable for all choices of Δt and α. This behavior at fine grids occurs because the coarse grid correction steps internal to the V-cycle use much larger time steps when Δt is fixed than when $\Delta t \sim h$ where h is the fine grid size. This can be seen in figs. 2.1(left,right) where diagonal slices of the amplification factors $|\hat{S}_h(\theta,\theta)|$, for $\theta \in [-\pi, \pi]$, are shown for $h = 1/16$, $1/32$ and $1/2048$ where $\Delta t = 0.005$ (left) and $\Delta t = 0.1/2048$ (right). Observe that for the latter case, the amplification factors for the coarse meshes are nearly equal to zero in contrast to the case with the larger time step.

These results suggest that when a fixed Δt is used, it's value should be such that the coarse grid HFS factors are less than 1. This is confirmed in Table 2.5 where the HFS factors are shown with $\Delta t = 0.1/64 = 1.5625 \times 10^{-3}$. This suggests that the multigrid method using a $V(1,1)$-cycle with fixed time step Δt converges uniformly with respect to increasing resolution if Δt is small enough. This would impose no grid-dependent time step constraints on our discrete scheme to solve the (**C-H**) equation.

Table 2.3: HFS factors for different mesh sizes. $\alpha = 0.25$ and $\Delta t = 5 \times 10^{-3}$

Case	16 x 16	32 x 32	64 x 64	128 x 128
μ_{loc}	0.0329	0.2984	0.5542	0.6392
$\sqrt{V(1,1)}$-cycle	0.1930	0.2706	0.3428	0.3920
Case	256 x 256	512 x 512	1024 x 1024	2048 x 2048
μ_{loc}	0.6618	0.6676	0.6691	0.6694
$\sqrt{V(1,1)}$-cycle	0.4127	0.4199	0.4294	0.4452

Table 2.4: HFS factors for different mesh sizes. $\alpha = -0.25$ and $\Delta t = 5 \times 10^{-3}$

Case	16 x 16	32 x 32	64 x 64	128 x 128
μ_{loc}	0.9594	1.2796	0.8014	0.7009
$\sqrt{V(1,1)}$-cycle	0.3925	0.8503	0.9098	0.9177
Case	256 x 256	512 x 512	1024 x 1024	2048 x 2048
μ_{loc}	0.6773	0.6715	0.6700	0.6697
$\sqrt{V(1,1)}$-cycle	0.9209	0.9220	0.9208	0.9860

Table 2.5: HFS factors for different mesh sizes. $\alpha = -0.25$ and $\Delta t = .1/64 = 1.5625 \times 10^{-3}$

Case	16 x 16	32 x 32	64 x 64	128 x 128
μ_{loc}	0.2324	0.9016	0.7874	0.7002
$\sqrt{V(1,1)}$-cycle	0.1305	0.5117	0.3522	0.3907
Case	256 x 256	512 x 512	1024 x 1024	2048 x 2048
μ_{loc}	0.6772	0.6715	0.6700	0.6697
$\sqrt{V(1,1)}$-cycle	0.4228	0.4206	0.4290	0.4495

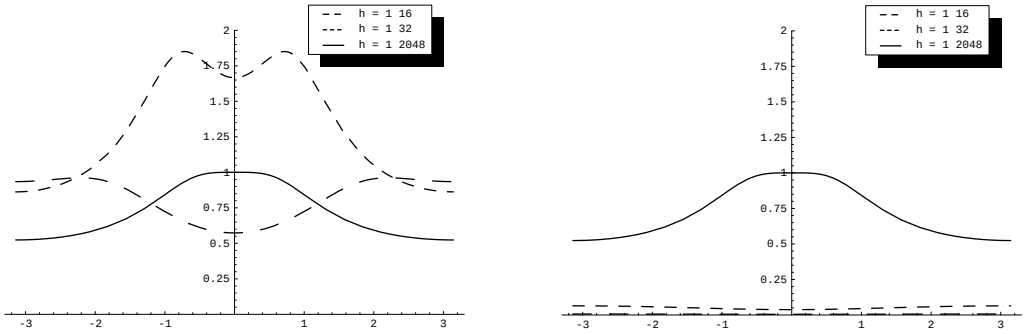

Figure 2.1: Diagonal slice of amplification factor $|\hat{S}_h(\theta,\theta)|$, over the interval $[-\pi,\ \pi]$. In the left figure, $\Delta t = 0.005$. In the right figure, $\Delta t = 0.1/2048$.

5.B Presence of flow

Here, we analyze projection algorithms given in section 2.4. In the approximate algorithm, the smoothing operator $L_h^{approx} = L_h^{a,+} + L_h^{a,-}$ where

$$\begin{aligned} L_h^{a,+}\psi &:= \left(\psi_{i+\frac{1}{2},j+\frac{3}{2}} - 4\psi_{i+\frac{1}{2},j+\frac{1}{2}} + \psi_{i+\frac{3}{2},j+\frac{1}{2}}\right)/h^2, \\ L_h^{a,-}\psi &:= \left(\psi_{i-\frac{1}{2},j+\frac{1}{2}} + \psi_{i+\frac{1}{2},j-\frac{1}{2}}\right)/h^2. \end{aligned}$$

The resulting amplification factor is

$$\hat{S}_h^a(\theta_1,\theta_2) := -\frac{\hat{L}_h^{a,-}(\theta_1,\theta_2)}{\hat{L}_h^{a,+}(\theta_1,\theta_2)},$$

where the symbols are calculated as in section 5.A. In Fig. 2.2 (left), the diagonal slice $|\hat{S}_h^a(\theta,\theta)|$ is shown. Observe that the HFS factor $\mu_{loc} \approx 0.45$ which suggests that the linear multigrid converges uniformly with respect to increasing resolution. In Fig. 2.2 (right), the corresponding amplification factor $|\hat{S}_h^e(\theta,\theta)|$ is shown for the exact projection method. In this case, however, the HFS factor $\mu_{loc} \approx 1.0$ and suggests that the smoothing operator does not remove high frequencies efficiently which reflects the ill-conditioning of the system. The corresponding multigrid method requires a prohibitive number of iterations to converge when the mesh size h is small. Therefore, in the numerical results we present in this paper, we only use the approximate projection algorithm.

2.6 Numerical experiments

In this section, we validate our scheme by verifying the second-order convergence and comparing the numerical results with the prediction of a linear stability analysis. We then perform simulation of spinodal decomposition and examine the effect of boundary conditions, flow and interfacial tension.

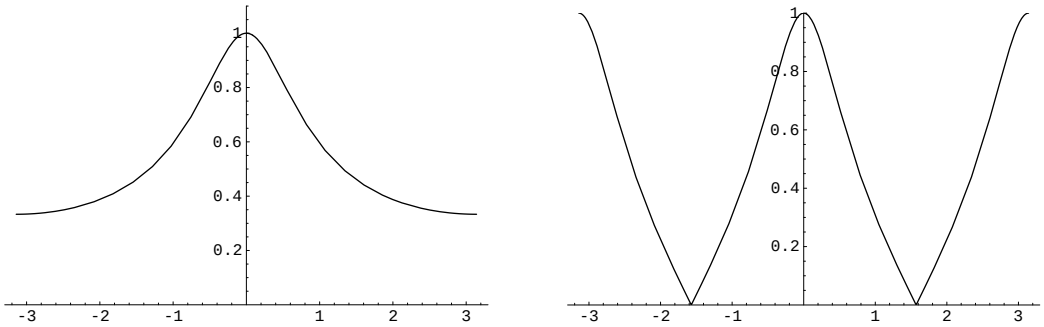

Figure 2.2: Diagonal slices of amplification factor. Approximate projection (left), exact projection (right).

6.A Convergence Test

To obtain an estimate of the rate of convergence, we perform a number of simulations for a sample initial problem on a set of increasingly finer grids. The initial data is

$$c_0(x,y) \quad = \quad 0.5 + 0.12\cos(2\pi x)\cos(2\pi y) + 0.2\cos(\pi x)cos(3\pi y) \tag{2.6.47}$$

on a square domain, $[0,1]\times[0,1]$. The numerical solutions are computed on the uniform grids, $\Delta x = \Delta y = h = 1/2^n$ for $n = 4, 5, 6, 7, 8, 9, 10,$ and 11. For each case, the calculations are run to time t = 0.2, the uniform time steps, $\Delta t = 0.1h$ and $\epsilon = 0.01$, are used to establish the convergence rates.

In our formulation of the method for (**C-H**) equation, since a cell centered grid is used, we define the error to be the discrete L_2-norm of the difference between that grid and the average of the next finer grid cells covering it:

$$e_{h/\frac{h}{2}\,ij} \stackrel{def}{=} c_{hij} - \left(c_{\frac{h}{2}\,2i,2j} + c_{\frac{h}{2}\,2i-1,2j} + c_{\frac{h}{2}\,2i,2j-1} + c_{\frac{h}{2}\,2i-1,2j-1}\right)/4.$$

The rate of convergence is defined as the ratio of successive errors: $\log_2(||e_{h/\frac{h}{2}}||/||e_{\frac{h}{2}/\frac{h}{4}}||)$.

The errors and rates of convergence are given in table 2.6. The results suggest that the scheme is indeed second order accurate. The deterioration of the rates from 2 at higher resolutions is believed to be due to accumulation of errors from coarse grid correction steps internal to the nonlinear multigrid method. In figure 2.3, the time evolution of the energy is shown accompanied with grey-scale contour images of the numerical solution u (black: $u = 0$, white: $u = 1$). As expected from lemma 2.2.25, the energy is non-increasing and tends to a constant value. This is in fact a local equilibrium for Neumann boundary conditions. The global equilibrium consists of a single interface.

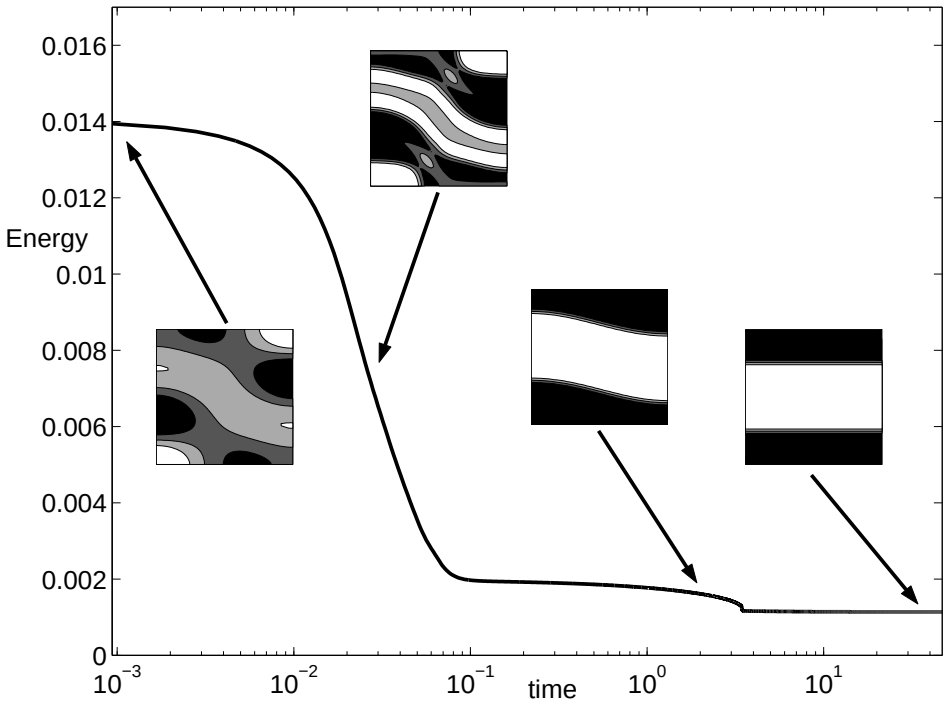

Figure 2.3: The time dependent energy of the numerical solutions with the initial data (2.6.47).

Table 2.6: l_2**-norm of the Errors and Convergence rates for concentration** c**.**

Case	16-32	Rate	32-64	Rate	64-128
l_2	1.721e-01	2.095e+00	4.027e-02	3.290e+00	4.118e-03

Case	Rate	128-256	Rate	256-512
l_2	2.028e+00	1.010e-03	1.958e+00	2.598e-04

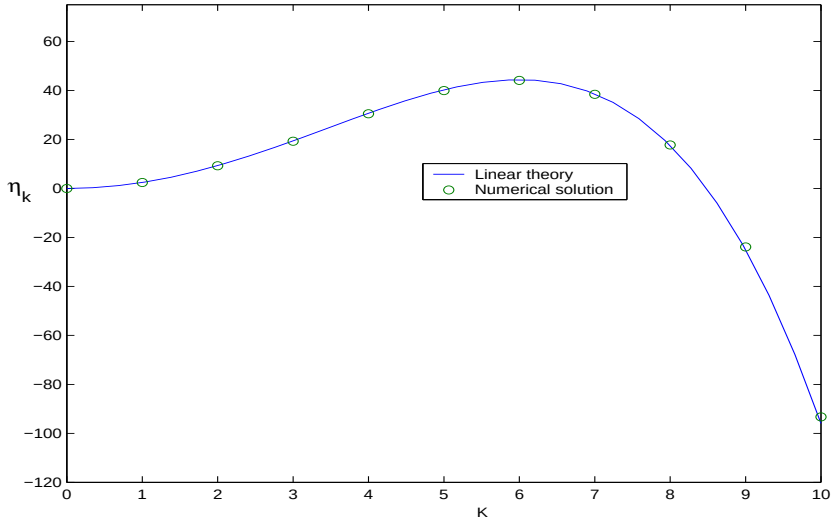

Figure 2.4: Growth rate for the different wave numbers k.

6.B Comparison with linear stability theory

Let c_m be the mean concentration. We look for a solution of equation (2.1.1) of the form

$$c(x,t) = c_m + \sum_{k=1}^{\infty} c^k(t)\cos(k\pi x)$$

where $|c^k(t)| \ll 1$. Neglecting quadratic terms in $c^k(t)$, we find that $c^k(t)$ must solve the ordinary differential equation,

$$\frac{dc^k}{dt} = -(k\pi)^2[\phi'(c_m) + \epsilon^2(k\pi)^2]c^k. \tag{2.6.48}$$

The solution is $c^k(t) = c^k(0)e^{\eta_k t},$ where $\eta_k = -(k\pi)^2[\phi'(c_m)+\epsilon^2(k\pi)^2]$ is the growth rate. Taking c_m in the spinodal region (i.e. $c_m \in (\frac{3-\sqrt{3}}{6},\ \frac{3+\sqrt{3}}{6}),\ \phi'(c_m) < 0$), Eq. (2.6.48) shows that the amplitude of a finite number of long wavelength perturbations will grow exponentially in time for sufficiently small ϵ.

In particular the fastest growing mode is

$$k_{max} = \sqrt{-\phi'(c_m)/(2\epsilon^2\pi^2)},$$

and the growth rate of this mode is $\eta_{k_{max}} = -(k_{max}\pi)^2\phi'(c_m)/2$.

In Fig. 2.4, the theoretical growth rate η_k is compared to that obtained from the nonlinear scheme. The numerical growth rate is defined by

$$\tilde{\eta}_k = \log\left(\frac{\max_{i,j}|c^n_{i,j} - 0.5|}{\max_{i,j}|c^0_{i,j} - 0.5|}\right)/t_n.$$

Here, we used $c_m = 0.5$, initial data $c_0(x) = 0.5 + 0.01\cos(k\pi x)$ and $\epsilon = 0.018757$, $\Delta t = 10^{-4}$, $h = 1/128$ and $t_n = 0.01$. The graph shows that the linear analysis (solid line) and numerical solution (circle) are in good agreement.

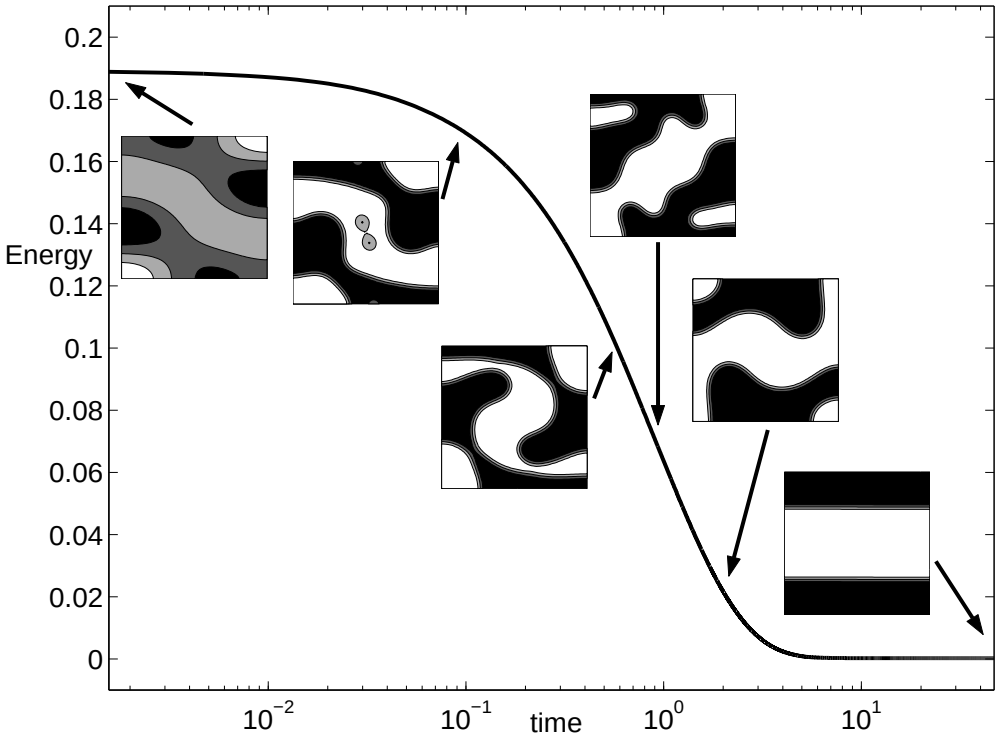

Figure 2.5: The time dependent total energy of the numerical solutions with the initial data (2.6.47).

Initial data are

$$\eta_1/\eta_2 = 1.0,\ \rho_1/\rho_2 = 1.0, \mathbf{Re} = 100.0,\ \mathbf{We} = 848.5$$
$$u(x,y) = -\sin^2(\pi x)\sin(2\pi y),\ \ v(x,y) = \sin^2(\pi y)\sin(2\pi x),$$

where the non-dimensional numbers $\mathbf{Re}$, $\mathbf{We}$ are defined at chapter 4.

Table 2.7: l_2**-norm of the Errors and Convergence rates for concentration** c**.**

Case	16-32	Rate	32-64	Rate	64-128
l_2	2.889e-01	2.261	6.030e-02	3.589	5.000e-03

Case	Rate	128-256	Rate	256-512
l_2	2.062	1.200e-03	2.014	2.970e-04

Table 2.8: l_2**-norm of the Errors and Convergence rates for velocity** u**.**

Case	16-32	Rate	32-64	Rate	64-128
l_2	3.900e-03	1.797	1.100e-03	1.787	3.230e-04

Case	Rate	128-256	Rate	256-512
l_2	1.990	8.132e-05	1.997	2.037e-05

Table 2.9: l_2**-norm of the Errors and Convergence rates for velocity** v**.**

Case	16-32	Rate	32-64	Rate	64-128
l_2	3.400e-03	1.582	1.200e-03	1.827	3.242e-04

Case	Rate	128-256	Rate	256-512
l_2	1.990	8.164e-05	1.997	2.045e-05

Table 2.10: l_2**-norm of the Errors and Convergence rates for pressure** p**.**

Case	17-33	Rate	33-65	Rate	65-129
l_2	8.200e-03	1.784	2.400e-03	2.921	3.129e-04

Case	Rate	129-257	Rate	257-513
l_2	1.774	9.149e-05	1.548	3.128e-05

Table 2.11: $(u^{n+\frac{1}{2}}, \nabla p^{n+\frac{1}{2}})_h / (u^{n+\frac{1}{2}}, u^{n+\frac{1}{2}})_h$

Case	16	Rate	32	Rate	64	Rate
α	-4.827e-03	1.5636	-1.633e-03	1.8942	-4.393e-04	1.9808

Case	128	Rate	256	Rate	512
α	-1.113e-04	1.9925	-2.797e-05	1.9913	-7.035e-06

6.C Spinodal decomposition

In order to demonstrate the applicability of the new numerical scheme we consider the effect of flow on spinodal decomposition. We consider an initial concentration field $c^0(\mathbf{x}) = c_m + \xi(\mathbf{x})$, where $c_m = 0.5$ is the spinodal point and $\xi(\mathbf{x})$ is a random perturbation with magnitude $|\xi(\mathbf{x})| \leq 0.01$. A 64×64 mesh was used on the square $\Omega = [0,1] \times [0,1]$ with periodic boundary conditions at $x = 0$ and 1. On $y = 0$ and 1 we used either no-slip velocity boundary conditions or an imposed shear. Neumann boundary conditions are used for c and μ. Further, we took $\epsilon = 0.01,\ h = 1/64,$ and $\Delta t = 0.1h$. In Fig. 2.6 the evolution is shown with no flow (left column) and in the presence of surface tension driven flow (right column) with $\mathbf{We} = 1$. That is, the flow arises only due to surface stresses between the components and we have taken $\mathbf{Re} = 1$ and $\eta_1/\eta_2 = \rho_1/\rho_2 = 1$. In Fig. 2.7, the evolution is shown in the absence and presence of surface stress (left and right columns respectively) and with an imposed shear flow, i.e. $(u(x,1), v(x,1)) = \left(\frac{1}{2}, 0\right)$ and $(u(x,1), v(x,1)) = \left(-\frac{1}{2}, 0\right)$. In the absence of surface stress, the velocity is the linear field $(u(x,y), v(x,y)) = (y - 1/2, 0)$ and satisfies the Navier-Stokes equation (2.1.8) with $\mathbf{We} = \infty$. In the presence of surface stress, the velocity field is non-linear and we have again taken the non-dimensional constants as described in Fig. (2.6) above with flow.

In Figs. 2.6 and 2.7, three spatial periods are shown and the time is constant across a row and increases down the column as indicated in the caption. The times in the corresponding rows for Figs. 2.6 and 2.7 are the same. At early times, there is classical spinodal decomposition as the unstable mixture phase separates and regions coalesce. There is little effect of surface stress and flow.

Let us focus first on the effect of surface tension in the absence of shear. In Fig. 2.6, at later times, we observe that surface tension acts to decrease the deformation of the interfaces and to reduce their overall length. This causes the fluid fingers to retract and become more vertical. This leads to the coalescence of the fingers with semi-circular drops at the bottom of the domain. The resulting vertical bands of fluid are a local equilibrium. In the absence of surface tension, the fingers do not coalesce with the drops and classical coarsening occurs as the mass transfers from the drops to the fingers. The fingers then coalesce to form a horizontal band. This is a global energy minimum.

In the presence of shear, we see from Fig. 2.7, that surface tension has a similar effect and the deformation of interfaces is reduced. Here, however, the morphologies are much more elongated due to the shear. Further, pinchoff and reconnection events occur and the morphology actually repeats itself in time. The stretched bands at $t = 5.63$ pinchoff and form drops (see $t = 6.56$). The drops then reconnect with the fingers (approximately at $t = 7.0$, not shown) and re-form the stretched bands. When $\mathbf{We} = \infty$, this sequence occurs twice while when $\mathbf{We} = 1$ this sequence occurs three times since the fingers retract more due to surface tension which thus enhances their capability to coalesce. This temporal periodicity

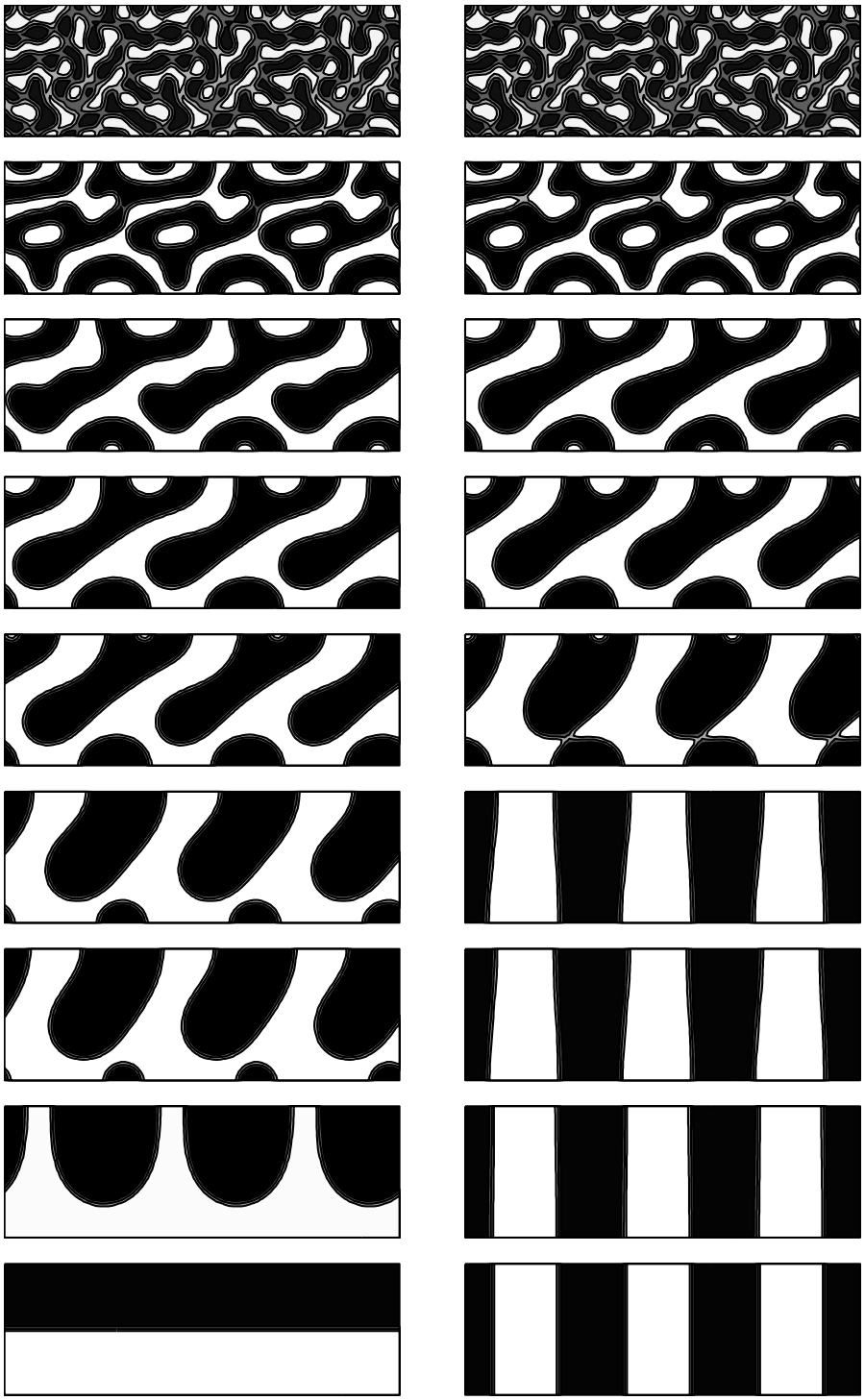

Figure 2.6: Evolution with different boundary and flow conditions (i.e., no surface tension and no shear, surface tension and no shear, respectively) at time t = 0.03, 0.16, 0.56, 0.94, 1.88, 5.63, 6.56, 17.81, and 45.94

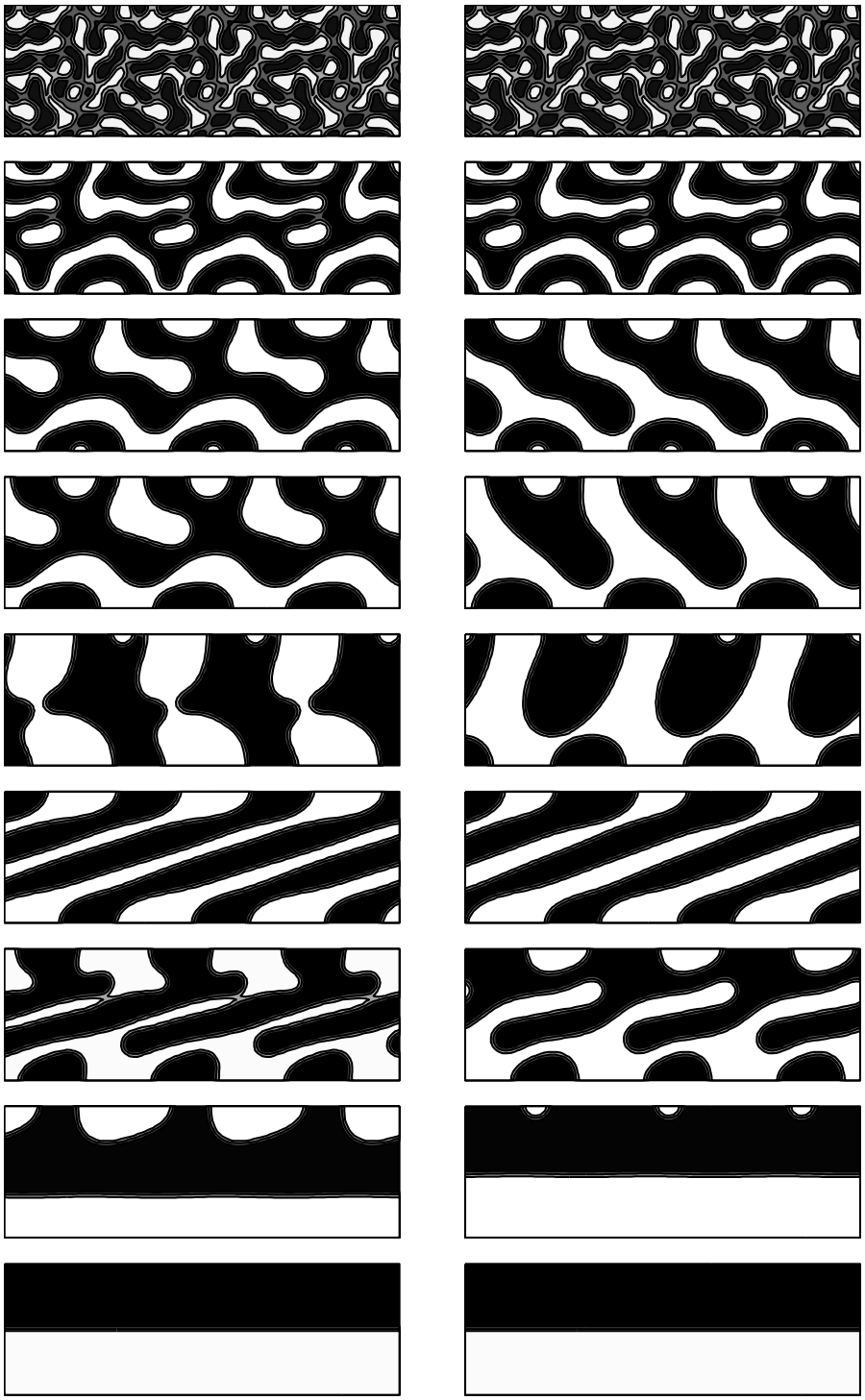

Figure 2.7: Evolution with different boundary and flow conditions (i.e., no surface tension and shear, and surface tension and shear, respectively) at time t = 0.03, 0.16, 0.56, 0.94, 1.88, 5.63, 6.56, 17.81, and 45.94

is discussed further below. Eventually, the periodicity is broken since the drops that are formed grow smaller with each cycle. After the last cycle, there is no reconnection. The fingers then merge to form a single horizontal band which is the global energy minimum.

In Fig. 2.8 the total energy (2.2.26) evolution for the different simulations is plotted versus time. In the absence of shear (dot-dashed: $\mathbf{We} = 1$ and dashed: $\mathbf{We} = \infty$), the energy decreases monotonically as predicted by lemma 2.3. In the presence of shear (large dots: $\mathbf{We} = 1$ and small dots: $\mathbf{We} = \infty$), there are energy oscillations. Observe that there are three oscillations when $\mathbf{We} = 1$ and two when $\mathbf{We} = \infty$. These correspond to the pinchoff and reconnection morphologies described above. In Fig. 2.9, we present the scaled kinetic (dashed) and surface energy (solid) evolutions through first oscillation ending approximately at $t = 9.0$. In addition, we indicate the concentration morphology at the indicated times. At early times, energy is transferred from the surface to the fluid while at later times this process is reversed. The peaks seen in the kinetic energy correspond to topology transitions of the concentration field in the flow. For example, lower ends of the stretched bands pinchoff to form drops at approximately $t = 6.4$. The drops then recoalesce at approximately $t = 7.0$. Each of these events is seen to be associated with a local peak in the kinetic energy. Energy is transmitted to the fluid through the large surface stress that develops during the transition and subsequent retraction of the fingers.

To understand further the oscillation of the energy in the presence of shear, we compare in Fig. 2.10 the difference quotient

$$\frac{\mathcal{E}_h(c^{n+1}, \mathbf{u}^{n+1}) - \mathcal{E}_h(c^n, \mathbf{u}^n)}{\Delta t}$$

shown as circles and to the corresponding spatial discretiation of the analytical value

$$\frac{d}{dt}\mathcal{E}(c^{n+\frac{1}{2}}, \mathbf{u}^{n+\frac{1}{2}}) = \frac{1}{\mathbf{Re}} \int \mathbf{u}^{n+\frac{1}{2}} \Delta \mathbf{u}^{n+\frac{1}{2}} - \frac{1}{\epsilon\, \mathbf{We}} \int |\nabla \mu^{n+\frac{1}{2}}|^2$$

shown as the solid line. Observe that there is excellent agreement between these two independently calculated quantities. This shows that the oscillations are due to the shear boundary condition and provides a check on the accuracy of our numerical results.

Next, in Fig. 2.11, we examine the growth of the fluid domains. We define the domain mass to be

$$h^2 \sum_{c_{ij} > 0.7} c_{ij},$$

which measures the mass of the fluid component in regions where there is little mixing of the components. At early times, the domain mass evolution is independent of surface stress and flow. The mass grows from zero at $t = 0$ to approximately 0.47 at $t \approx 45$. In the absence of shear, the growth is monotonic and the domain mass for the cases with surface tension seems to grow faster than when $\mathbf{We} = \infty$. The most distinguishing features of the results, however, are the oscillations due to interface stretching in

the presence of shear. This occurs because mass is distributed throughout the diffuse-interface and thus when interface area increases, so does the amount of mass trapped in the interfacial zone. Therefore, as interfaces are stretched, our measure of domain mass decreases when in fact the total amount of mass in any one region is nearly constant. We are currently working to obtain a measure insensitive to this phenomenon.

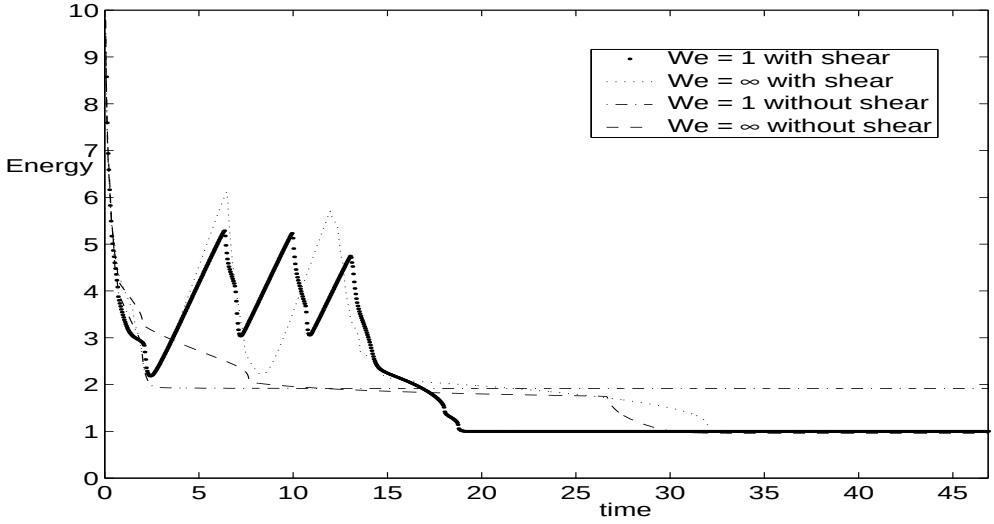

Figure 2.8: Energy evolution with four different boundary and flow conditions.

2.7 Conclusions

In this paper, we have developed a conservative, 2^{nd} order accurate fully implicit discretization of the NSCH system that has an associated discrete energy functional. In addition, the scheme has a straightforward extension to multi-component systems. This will be exploited in a future work where we examine fluid flows with three constituent components.

To efficiently solve the discrete system at the implicit time-level, we have developed a nonlinear multigrid method to solve the CH equation which is then coupled to a projection method that is used to solve the NS equation. We have analyzed and proved convergence of the scheme in the absence of flow. The convergence in the presence of flow is currently under study. We demonstrated convergence of our scheme numerically in both the presence and absence of flow and performed simulations of phase separation via spinodal decomposition. We examined the separate effects of surface tension and external flow on the decomposition. We found surface tension driven flow alone increases coalescence rates through the retraction of interfaces. When there is an external shear flow, the topology of the flow can

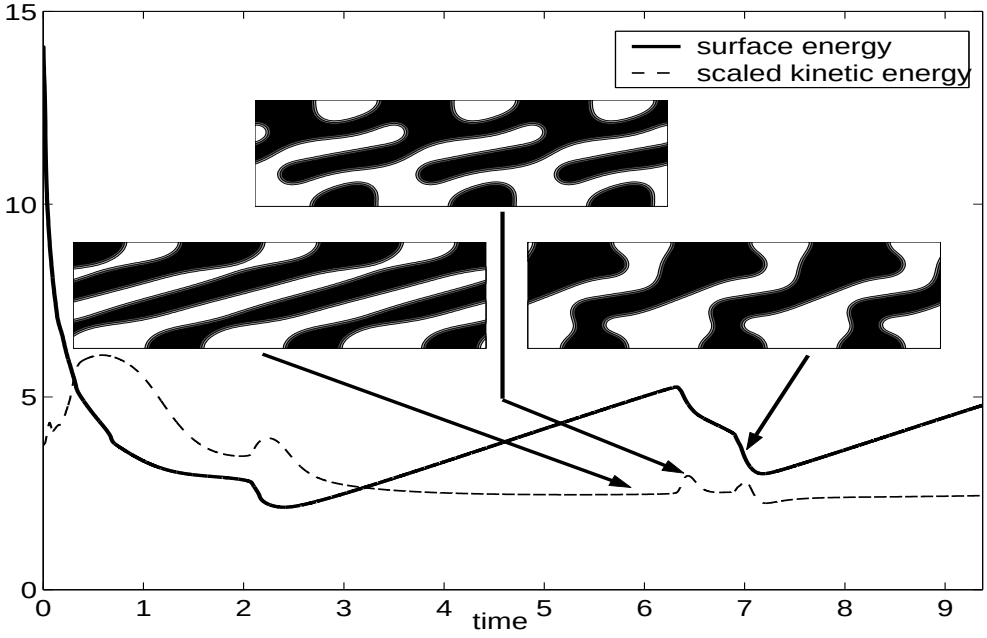

Figure 2.9: Surface energy and scaled kinetic energy.

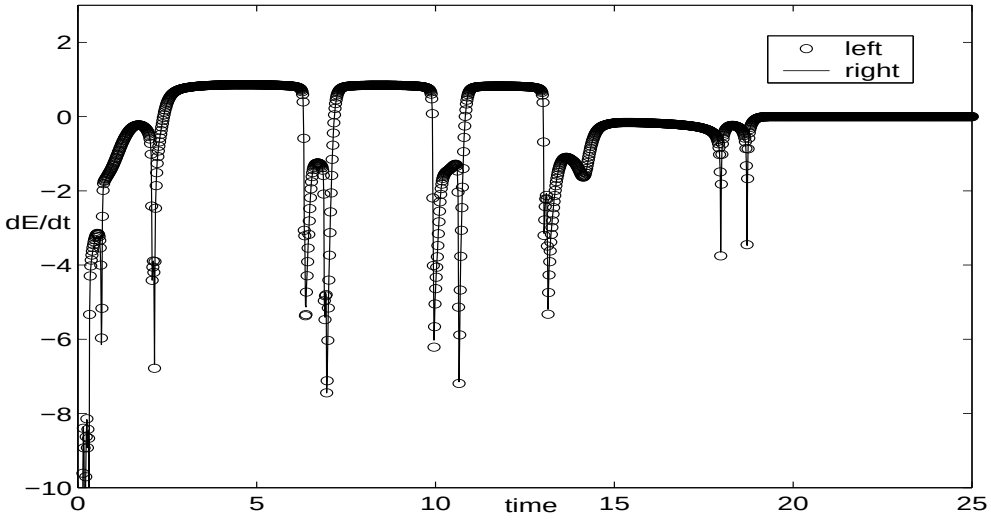

Figure 2.10: Comparison with left and right hand sides of Eq. 2.2.27

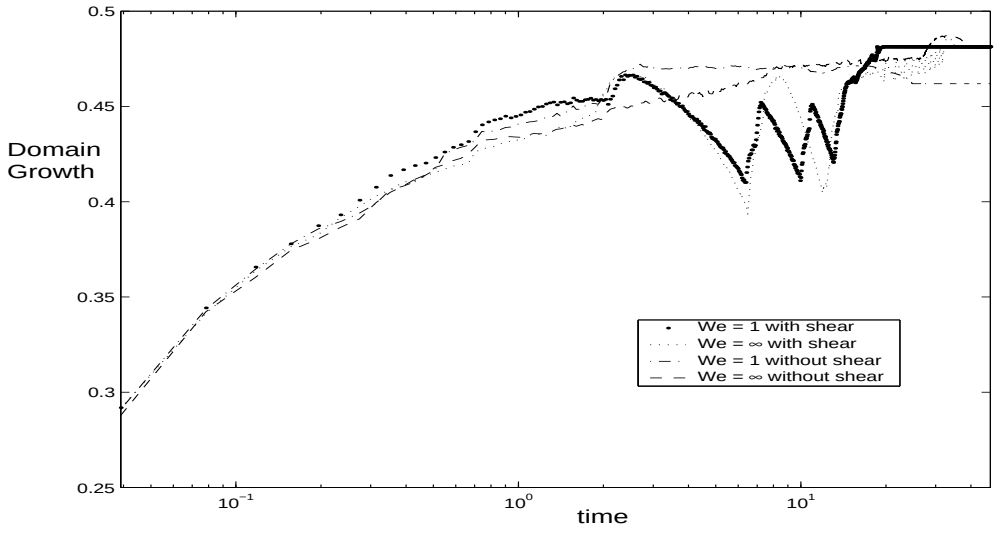

Figure 2.11: Domain growth with four different boundary and flow conditions.

become periodic in time as multiple pinchoff and reconnection events occur. Eventually, the periodic motion ceases and the system relaxes to a global equilibrium. The equilibria we observe appears has a similar structure in all cases although the dynamics of the evolution is quite different.

2.8 Appendix

8.A Smoothing operator

Use Gauss-Seidel update for the linear part and the Jacobi update for the nonlinear part. The equations for updates are written as two equations for $\{\bar{c}_k^m,\ \bar{\mu}_k^{m-\frac{1}{2}}\}$ and they are updated simultaneously by solving the two equations. We use (2.8.51), (2.8.52), and (2.8.53) to perform the $SMOOTH$ relaxation operator. Here is the derivation of the $SMOOTH$ relaxation operator. Rewriting (2.5.44), we get

$$\frac{c_{ij}^{n+1}}{\Delta t} + (\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2})\mu_{ij}^{n+\frac{1}{2}} \quad = \quad (\frac{\mu_{i+1,j}^{n+\frac{1}{2}} + \mu_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{\mu_{i,j+1}^{n+\frac{1}{2}} + \mu_{i,j-1}^{n+\frac{1}{2}}}{\Delta y^2}) + \frac{c_{ij}^n}{\Delta t}.$$

By Eq. (2.5.45),

$$\begin{aligned} -\frac{\epsilon^2}{2}(\frac{2}{\Delta x^2} \quad + \quad \frac{2}{\Delta y^2})c_{ij}^{n+1} + \mu_{ij}^{n+\frac{1}{2}} &= \frac{1}{2}\epsilon^2\Delta_d c_{ij}^n + \hat{\phi}(c_{ij}^n, c_{ij}^{n+1}) \\ -\frac{\epsilon^2}{2\Delta x^2}(c_{i+1,j}^{n+1} + c_{i-1,j}^{n+1}) &- \frac{\epsilon^2}{2\Delta y^2}(c_{i,j+1}^{n+1} + c_{i,j-1}^{n+1}). \end{aligned} \tag{2.8.49}$$

We will solve these two equations (2.8.49), (2.8.49) by using Nonlinear Multigrid Method. Since

$\hat{\phi}(c_{ij}^n, c_{ij}^{n+1})$ is nonlinear with respect to c_{ij}^{n+1}, we linearize $\hat{\phi}(c_{ij}^n, c_{ij}^{n+1})$ at c_{ij}^m

$$\hat{\phi}(c_{ij}^n, c_{ij}^{n+1}) = \hat{\phi}(c_{ij}^n, c_{ij}^m) + \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial c_2}(c_{ij}^{n+1} - c_{ij}^m).$$

After substitution of this into (2.8.49), we get

$$\begin{aligned} -[\frac{\epsilon^2}{2}(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}) &+ \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial c_2}]c_{ij}^{n+1} + \mu_{ij}^{n+\frac{1}{2}} \\ &= \frac{\epsilon^2}{2}\Delta c_{ij}^n + \hat{\phi}(c_{ij}^n, c_{ij}^m) - \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial c_2} c_{ij}^m \\ &\quad -\frac{\epsilon^2}{2\Delta x^2}(c_{i+1,j}^{n+1} + c_{i-1,j}^{n+1}) - \frac{\epsilon^2}{2\Delta y^2}(c_{i,j+1}^{n+1} + c_{i,j-1}^{n+1}). \end{aligned} \tag{2.8.50}$$

We use the following iterative technique: For a given $\{c^0, \mu^{-\frac{1}{2}}\}$, an initial guess to $\{c^{n+1}, \mu^{n+\frac{1}{2}}\}$, update $\{c^{m+1}, \mu^{m+\frac{1}{2}}\}$ until it converges to $\{c^{n+1}, \mu^{n+\frac{1}{2}}\}$, where $\{c^m, \mu^{m-\frac{1}{2}}\}$ and $\{c^{m+1}, \mu^{m+\frac{1}{2}}\}$ denote the old and the new approximation of the iteration, respectively. Usually, we take $c^0 = c^n, \mu^{-\frac{1}{2}} = \phi(c^n) - \epsilon^2 \Delta_d c^n$. Thus, the iteration scheme is

$$\frac{c_{ij}^{m+1}}{\Delta t} + (\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2})\mu_{ij}^{m+\frac{1}{2}} = \frac{\mu_{i+1,j}^{m-\frac{1}{2}} + \mu_{i-1,j}^{m+\frac{1}{2}}}{\Delta x^2} + \frac{\mu_{i,j+1}^{m-\frac{1}{2}} + \mu_{i,j-1}^{m+\frac{1}{2}}}{\Delta y^2} + \frac{c_{ij}^n}{\Delta t} \tag{2.8.51}$$

and

$$\begin{aligned} -[\frac{\epsilon^2}{2}(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}) &+ \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial c_2}]c_{ij}^{m+1} + \mu_{ij}^{m+\frac{1}{2}} \\ &= -\frac{\epsilon^2}{2}\Delta c_{ij}^n + \hat{\phi}(c_{ij}^n, c_{ij}^m) - \frac{\partial \hat{\phi}(c_{ij}^n, c_{ij}^m)}{\partial c_2} c_{ij}^m \\ &\quad -\frac{\epsilon^2}{2\Delta x^2}(c_{i+1,j}^m + c_{i-1,j}^{m+1}) - \frac{\epsilon^2}{2\Delta y^2}(c_{i,j+1}^m + c_{i,j-1}^{m+1}), \end{aligned} \tag{2.8.52}$$

with homogeneous Neumann boundary conditions

$$\begin{aligned} &c_{0,j} = c_{1,j}, \quad c_{N_x+1,j} = c_{N_x,j}, \quad c_{i,0} = c_{i,1}, \quad c_{i,N_y+1} = c_{i,N_y}, \\ &\mu_{0,j} = \mu_{1,j}, \quad \mu_{N_x+1,j} = \mu_{N_x,j}, \quad \mu_{i,0} = \mu_{i,1}, \quad \mu_{i,N_y+1} = \mu_{i,N_y}. \end{aligned} \tag{2.8.53}$$

8.B multigrid method for the intermediate velocity field $\mathbf{u}^*$

For simplicity, let $\mathbf{u} = \mathbf{u}^{*,k+1}$, then we have

$$L\mathbf{u} = (f, g), \tag{2.8.54}$$

where

$$L = \frac{1}{\Delta t} - \frac{1}{2\mathbf{Re}}\Delta_d$$

and

$$(f,g) = -\frac{1}{2}\omega^{n+\frac{1}{2},k} \times (\mathbf{u}^{n+1,k} + \mathbf{u}^n) - \tilde{\nabla}_d^c P^{n-\frac{1}{2}} + \frac{1}{2\mathbf{Re}}\Delta_d \mathbf{u}^n + \frac{1}{2\epsilon\mathbf{We}}\left(c^{n+1,k} + c^n\right)\nabla_d \mu^{n+\frac{1}{2},k}$$

Since operator L is a linear operator, we use a linear multigrid method to solve Eq. (2.8.54).

Multigrid cycle

$$\mathbf{u}_k^{m+1} = MGCYC(k, \mathbf{u}_k^m, L_k, f_k^n, g_k^n, \nu).$$

Now, define the MGCYC.

(1) Presmoothing

Compute $\bar{\mathbf{u}}_k^m$ by applying ν smoothing steps to $\mathbf{u}_k^m$

$$\bar{\mathbf{u}}_k^m = SMOOTH^\nu(\mathbf{u}_k^m, L_k, f_k^n, g_k^n),$$

One $SMOOTH$ operator step consists of relaxing the system (2.8.54) given below for each i and j.

$$\bar{u}_{ij}^m = \left(f + \frac{u_{i+1,j}^m + \bar{u}_{i-1,j}^m + u_{i,j+1}^m + \bar{u}_{i,j-1}^m}{2h^2\mathbf{Re}}\right) / \left(\frac{1}{\Delta t} + \frac{2}{h^2\mathbf{Re}}\right),$$

$$\bar{v}_{ij}^m = \left(g + \frac{v_{i+1,j}^m + \bar{v}_{i-1,j}^m + v_{i,j+1}^m + \bar{v}_{i,j-1}^m}{2h^2\mathbf{Re}}\right) / \left(\frac{1}{\Delta t} + \frac{2}{h^2\mathbf{Re}}\right).$$

(2) Compute the defect

$$(\bar{d_1}_k^m, \bar{d_2}_k^m) = (f_k^n, g_k^n) - L_k(\bar{u}_k^m, \bar{v}_k^m).$$

(3) Restrict the defect

$$(\bar{d_1}_{k-1}^m, \bar{d_2}_{k-1}^m) = I_k^{k-1}(\bar{d_1}_k^m, \bar{d_2}_k^m),$$

(4) Compute an approximate solution $\{\hat{u}_{k-1}^m,\ \hat{v}_{k-1}^m\}$ of the coarse grid equation on Ω_{k-1}, i.e.

$$L_{k-1}(u_{k-1}^m, v_{k-1}^m) = (\bar{d_1}_{k-1}^m, \bar{d_2}_{k-1}^m). \tag{2.8.55}$$

If $k = 1$, we use a direct solver. If $k > 1$, we solve (2.8.55) by performing a k-grid cycle using the zero grid function as an initial approximation:

$$\hat{\mathbf{v}}_{k-1}^{m} = MGCYC(k-1, \mathbf{0}, L_{k-1}, \bar{d_1}_{k-1}^{m}, \bar{d_2}_{k-1}^{m}, \nu).$$

(5) Interpolate the correction

$$\begin{aligned} \hat{v_1}_k^m &= I_{k-1}^k \hat{v_1}_{k-1}^m. \\ \hat{v_2}_k^m &= I_{k-1}^k \hat{v_2}_{k-1}^m. \end{aligned}$$

(6) Compute the corrected approximation on Ω_k

$$\begin{aligned} u_k^{m,\text{ after } CGC} &= \bar{u}_k^m + \hat{v_1}_k^m. \\ v_k^{m,\text{ after } CGC} &= \bar{v}_k^m + \hat{v_2}_k^m. \end{aligned}$$

(7) Postsmoothing

Compute $\mathbf{u}_k^{m+1}$ by applying ν smoothing steps to $\mathbf{u}_k^{m,\text{ after } CGC}$,

$$\mathbf{u}_k^{m+1} = SMOOTH^{\nu}(\mathbf{u}_k^{m,\text{ after } CGC}, L_k, f_k^n, g_k^n).$$

This completes the description of a multigird cycle.

8.C Crank-Nicholson scheme

We present another Crank-Nicholson scheme which has time restriction ($\Delta t < Ch^2$) for the energy decrease, but we can prove energy decrease and the second order convergence with a slightly modification to the proofs which we gave to the proposed scheme and the nonlinear multigrid implementation is very similar, also has straightforward implementation for multi-component cases. And at the linear level, it is same to proposed scheme.

$$\frac{c_{ij}^{n+1} - c_{ij}^n}{\Delta t} = \Delta_d \mu_{ij}^{n+\frac{1}{2}},$$

$$\mu_{ij}^{n+\frac{1}{2}} = \frac{1}{2}(\phi(c_{ij}^{n+1}) + \phi(c_{ij}^n)) - \frac{\epsilon^2}{2}\Delta_d(c_{ij}^{n+1} + c_{ij}^n).$$