J Sci Comput (2015) 63:913–937
DOI 10.1007/s10915-014-9920-3

Local Discontinuous Galerkin Methods for the Functionalized Cahn–Hilliard Equation

Ruihan Guo · Yan Xu · Zhengfu Xu

Received: 18 March 2014 / Revised: 21 July 2014 / Accepted: 8 September 2014 /
Published online: 19 September 2014

Abstract In this paper, we develop a local discontinuous Galerkin (LDG) method for the sixth order nonlinear functionalized Cahn–Hilliard (FCH) equation. We address the accuracy and stability issues from simulating high order stiff equations in phase-field modeling. Within the LDG framework, various boundary conditions associated with the background physics can be naturally implemented. We prove the energy stability of the LDG method for the general nonlinear case. A semi-implicit time marching method is applied to remove the severe time step restriction ($\Delta t \sim O(\Delta x^6)$) for explicit methods. The $h-p$ adaptive capability of the LDG method allows for capturing the interfacial layers and the complicated geometric structures of the solution with high resolution. To enhance the efficiency of the proposed approach, the multigrid (MG) method is used to solve the system of linear equations resulting from the semi-implicit temporal integration at each time step. We show numerically that the MG solver has mesh-independent convergence rates. Numerical simulation results for the FCH equation in two and three dimensions are provided to illustrate that the combination of the LDG method for spatial approximation, semi-implicit temporal integration with the MG solver provides a practical and efficient approach when solving this family of problems.

Research supported by NSFC Grant Nos. 11371342, 11031007, Fok Ying Tung Education Foundation No.131003. Research supported by NSF Grant DMS-1316662.

R. Guo · Y. Xu (✉)
School of Mathematical Sciences, University of Science and Technology of China,
Hefei 230026, Anhui, People's Republic of China
e-mail: yxu@ustc.edu.cn

R. Guo
e-mail: guoguo88@mail.ustc.edu.cn

Z. Xu
Department of Mathematical Sciences, Michigan Technological University,
Houghton, MI 49931, USA
e-mail: zhengfux@mtu.edu

Keywords Functionalized Cahn–Hilliard equation · Local discontinuous Galerkin method · Multigrid algorithm · Stability

1 Introduction

Phase-field modeling has been an alternative method to investigate morphology and microstructural evolution of systems with blending of various materials on a macroscopic level which gives the advantage of significantly less computational cost over the more accurate molecular dynamics simulation, see [5,6,12] and the references therein. In the phase-field modeling, the dynamics of the underlying physical system is generally described by a gradient flow resulting from the Euler–Lagrangian variation of a pre-defined energy form with embedded phase-field functions. However, the gradient flow, in the form of high order stiff partial differential equations (PDEs), poses a great deal of difficulty for numerical simulation. Namely, high resolution simulation is preferred in order to capture the generally sharp interfacial structures and to provide numerical solution with fidelity; The system itself experiences long time evolution therefore computational efficiency is essential to map out the whole dynamics from initial state to steady state; The physical domain under consideration is normally irregular which makes it very difficult to apply fast computational methods such as fast Fourier or pseudo-spectral method, not to mention various boundary conditions of the background physical process. In this paper, we would like to provide a practical tool to handle those issues using the PDE from a functionalized Cahn–Hilliard model as an example. The proposed approach can be considered for other similar phase-field equations as well.

To be specific, we consider numerical methods in a bounded domain $\Omega \in \mathbb{R}^d (d \leq 3)$ for the functionalized Cahn–Hilliard (FCH) equation

$$u_t = \Delta \left[(\epsilon^2 \Delta - W''(u))(\epsilon^2 \Delta u - W'(u)) + \epsilon(\eta_1 \epsilon^2 \Delta u - \eta_2 W'(u)) \right], \tag{1.1}$$

where u takes values between -1 and 1 and represents the volume fraction of polymer and solvent in the mixture. Here $W(u)$ is generally given as a potential well for the purpose of phase separation.

The FCH equation was proposed by Gompper and Schick [18] as a model for interfacial energy in phase separated mixtures with an amphiphilic structure. Then, Promislow [24] developed the model to describe nanoscale morphology changes in functionalized polymer chains. Gavish et al. introduced the FCH energy and extended the scope of the continuum variational approach to incorporate the influence of the solvation entropy of ions on the network morphology of phase separated materials in [16,17].

In this paper, we develop a local discontinuous Galerkin (LDG) method for the FCH equation. The $h - p$ adaptive capability of the LDG method allows for resolving the interfacial layers and the complicated geometric structures of the solution with high accuracy. The nonlinear stability of the LDG scheme provides a layer of robustness of the numerical simulation. To add another layer of stability for robust long-time simulation, we prove the energy stability of the LDG method for the general nonlinear case. Also, the LDG method smoothly incorporate the background physical boundary conditions into the numerical computation.

The LDG spatial discretization for the FCH equation typically results in a stiff ordinary differential equation (ODE). Explicit time marching methods require extremely small time step size ($\Delta t \sim O(\Delta x^6)$) to maintain the stability of the methods. This requirement turns explicit methods useless, considering that a long time simulation is generally expected to

obtain a steady state solution. In this case, implicit time integration is often applied to permit adaptive time stepping strategy for both accuracy and efficiency. Obviously, it requires to solve a system of linear equations at each time step. Traditional iterative solution methods such as Gauss–Seidel method suffers from slow convergence rates. Therefore, a faster iterative solver for the system of equations is essential. To enhance the efficiency of the proposed approach, the multigrid (MG) method is used to solve the system of linear equations resulting from the semi-implicit temporal integration at each time step. Numerical simulation results demonstrate that the MG method is an efficient method and the number of iterations is independent of the problem size.

In order to predict the MG behavior, a two-level local mode analysis is applied to study the convergence of the MG method. Although we restrict ourselves to two-dimensional problems, with considerably extra complexity, a similar analysis can be made for three-dimensional problems by the tensor product principle.

There are only few numerical methods for the simulation of the solution. In [4], Chen and Shen rewrote the sixth order nonlinear FCH equation to a system of three coupled second-order equations and applied the spectral-Galerkin methods. Jones et al. introduced an implicit-explicit scheme and the spectral method for the FCH equation in [20]. The FCH model shows high similarities to the Cahn–Hilliard (CH) model, and due to this, we can borrow the numerical methods for the CH equation to treat the FCH equation. Xia et al. developed the LDG method and proved the energy stability for the CH equation in [26], which used the discontinuous, piecewise polynomials as the solution and test functions. Guo and Xu studied the multigrid solver coupled with the LDG method for the CH equation in [19]. Elliott et al. [13–15] developed finite element methods for solving the CH equation. Kay and Welford [21] introduced a nonlinear MG method coupled with a finite element approximation of the CH equation and the total work required is $O(N)$. Kim et al. presented the conservative multigrid methods for CH fluids and ternary CH systems in [22,23].

The discontinuous Galerkin (DG) method we discuss in this paper is a class of finite element methods using completely discontinuous piecewise polynomial space for the solution and test functions. It was first designed as a method for solving hyperbolic conservation laws containing only first order spatial derivatives, e.g. Reed and Hill [25] for solving linear equations and Cockburn et al. [7–10] for solving nonlinear hyperbolic conservation laws.

For partial differential equations (PDEs) containing higher order spatial derivatives, it is difficult to apply the DG method directly, so the LDG method was introduced. The LDG method was firstly introduced by Cockburn and Shu in [11] as an extension of the Runge-Kutta DG (RKDG) method to general convection-diffusion problems. Xia et al. designed the LDG method for the CH equation and the Allen–Cahn/Cahn–Hilliard system in [26,27]. The idea of the LDG method is to rewrite the equations with higher order derivatives into a first order system, then apply the DG method on the system. The design of the numerical fluxes is the key ingredient to ensure stability. More details about the LDG methods for high-order time dependent PDEs can be found in the review paper [28].

The outline of this paper is as follows. In Sect. 2, we review the properties of the FCH equation. In Sect. 3, we present the LDG method for the FCH equation and give a proof of the energy stability for the semi-discrete LDG scheme. A semi-implicit time marching method is presented in Sect. 4. In Sect. 5, we give the two-level local mode analysis of the MG solver. Section 6 contains numerical results for the nonlinear FCH equation in two and three dimensions, which demonstrate the accuracy and capability of the methods. Finally we give concluding remarks in Sect. 7. A detailed description of the bi-grid algorithm and the two-level local mode analysis are presented in Appendices 1, 2, respectively.

2 The Functionalized Cahn–Hilliard Equation

In the development of modern materials for energy generation and storage, what becomes a primary difficulty is effective control of morphology on the nanoscale level. The knowledge of what governs the formation and evolution of the nanoscale morphology is lacking in many applications. There is an additional difficulty that the morphology of the active (hydrated) state cannot be probed directly in the case of soft polymer materials. To better understand the nanostructure of Nafion and other similar materials, the FCH model was developed by Promislow et al. [24] to describe the time evolution of the nanoscale morphology generated by hydrating polymer chains with a polar solvent. With a small set of parameters, the FCH model characterizes bilayer, pore-like, and micelle network structures.

To introduce the FCH energy, we first give the Cahn–Hilliard (CH) energy. The CH energy was introduced by Cahn and Hilliard in [3] to characterize a binary mixture by a phase field function u that maps $\Omega \subset \mathbb{R}^n$ into mixture values $[-1, 1]$, with $u = \pm 1$ in water and polymer domains respectively. The CH (or Ginzburg–Landau) interfacial free energy takes the form

$$\mathcal{E}(u) = \int_\Omega \left(\frac{\epsilon^2}{2} |\nabla u|^2 + W(u) \right) d\Omega, \tag{2.1}$$

where ϵ is the thickness of the interface between the phases.

The FCH energy, a continuum characterization of interfacial energy whose minimizers describe the network morphology of solvated functionalized polymer membranes. It is derived through the square of the variational derivative of the CH energy $\mathcal{E}(u)$, balanced against unfolding parameters η_1 and η_2 as explained in [24]. Thus it takes the form

$$\mathcal{F}(u) = \int_\Omega \frac{1}{2} \left(\epsilon^2 \Delta u - W'(u) \right)^2 - \epsilon \left(\frac{\epsilon^2 \eta_1}{2} |\nabla u|^2 + \eta_2 W(u) \right) d\Omega. \tag{2.2}$$

The first term in the integrand accounts for elastic bending energy of the polymer, and the second term for the surface area which tends to increase due to reduction of electrostatic energy when the charged polymer side-chains are solvated. The parameters η_1 and η_2 are positive constants, which govern the nature of the energetic interactions. η_1 and η_2 can be chosen as the same value or not. $W(u)$ is a function that describes potential energy of mixing and a typical form is

$$W(u) = \frac{1}{4}(u^2 - 1)^2, \tag{2.3}$$

which is the symmetric Ginzburg–Landau double-well potential with equilibria at $u = 1$ and $u = -1$.

The Euler–Lagrangian variation of the energy (2.2) gives chemical potential

$$\mu = \frac{\delta F}{\delta u} = (\epsilon^2 \Delta - W''(u))(\epsilon^2 \Delta u - W'(u)) + \epsilon(\eta_1 \epsilon^2 \Delta u - \eta_2 W'(u)). \tag{2.4}$$

Thus we have the FCH equation

$$u_t = \Delta \mu = \Delta \left[(\epsilon^2 \Delta - W''(u))(\epsilon^2 \Delta u - W'(u)) + \epsilon(\eta_1 \epsilon^2 \Delta u - \eta_2 W'(u)) \right]. \tag{2.5}$$

Neumann boundary conditions are prescribed for the physical domain, namely

$$\frac{\partial u}{\partial \nu} = \frac{\partial w}{\partial \nu} = \frac{\partial \mu}{\partial \nu} = 0, \quad \text{on} \quad \partial\Omega, \tag{2.6}$$

where $w = \epsilon^2 \Delta u - W'(u)$ and v is the normal vector to $\partial\Omega$. The H^{-1} gradient flow is mass preserving and the total energy decays with respect to time t, i.e.

$$\begin{aligned}\frac{d}{dt}\mathcal{F}(u) &= \int_\Omega (\epsilon^2\Delta u - W'(u))(\epsilon^2\Delta u_t - W''(u)u_t) - \epsilon(\epsilon^2\eta_1\nabla u\cdot\nabla u_t + \eta_2 W'(u)u_t)d\Omega \\ &= \int_\Omega (\epsilon^2\Delta - W''(u))(\epsilon^2\Delta u - W'(u))u_t + \epsilon(\eta_1\epsilon^2\Delta u - \eta_2 W'(u))u_t d\Omega \\ &= \int_\Omega \mu\Delta\mu d\Omega = -\int_\Omega \nabla\mu\cdot\nabla\mu d\Omega \le 0.\end{aligned}$$

The L^2 gradient flow associated with the free energy (2.2) is

$$u_t = -\mu = -[(\epsilon^2\Delta - W''(u))(\epsilon^2\Delta u - W'(u)) + \epsilon(\eta_1\epsilon^2\Delta u - \eta_2 W'(u))]. \tag{2.7}$$

To conserve total mass for the equation, we introduce a zero-mass projection Π. Then the zero-mass projection gradient flow of the free energy (2.2) is

$$u_t = \Pi\{-[(\epsilon^2\Delta - W''(u))(\epsilon^2\Delta u - W'(u)) + \epsilon(\eta_1\epsilon^2\Delta u - \eta_2 W'(u))]\}. \tag{2.8}$$

The dynamics described by (2.5) and (2.8) are different. In this paper, we just focus on the H^{-1} gradient flow, and design an LDG method and a suitable time marching strategy such that relatively large time steps can be used for the simulation.

3 The LDG Method for the FCH Equation

In this section, we consider the LDG method for the FCH equation (2.5) in $\Omega \in \mathbb{R}^d$ with $d \le 3$. The LDG method and the energy stability results of this paper are valid for all $d \le 3$ and the general form of $W(u)$.

3.1 Notations

Let $\mathcal{T}_h$ denote a tessellation of Ω with shape-regular element K. Let Γ denote the union of the boundary faces of elements $K \in \mathcal{T}_h$, i.e. $\Gamma = \bigcup_{K\in\mathcal{T}_h} \partial K$, and $\Gamma_0 = \Gamma\backslash\partial\Omega$.

In order to describe the flux functions, we need to introduce some notations. Let e be a face shared by the "left" and "right" elements K_L and K_R (we refer to [28] for more details of the definition). Define the normal vectors $\boldsymbol{\nu}_L$ and $\boldsymbol{\nu}_R$ on e pointing exterior to K_L and K_R, respectively. If ψ is a function on K_L and K_R, but possibly discontinuous across e, let ψ_L denote $(\psi|_{K_L})|_e$ and ψ_R denote $(\psi|_{K_R})|_e$, the left and right trace, respectively.

Let $\mathcal{P}^k(K)$ be the space of polynomials of degree at most $k \ge 0$ on K. The finite element spaces associated with the mesh are of the form

$$\begin{aligned}V_h^k &= \{v \in L^2(\Omega) : v|_K \in \mathcal{P}^k(K), \forall K \in \mathcal{T}_h\}, \\ \Sigma_h^k &= \{\boldsymbol{w} = (w_1,\ldots,w_d)^T \in L^2(\Omega)^d : w_l|_K \in \mathcal{P}^k(K), l = 1,\ldots,d, \forall K \in \mathcal{T}_h\}.\end{aligned}$$

Note that functions in V_h^k and Σ_h^k are allowed to be completely discontinuous across element interfaces.

3.2 The LDG Method

To define the LDG method for the FCH equation, we first rewrite (2.5) as a first order system:

$$u_t = \nabla \cdot \boldsymbol{p}, \tag{3.1a}$$
$$\boldsymbol{p} = \nabla(p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r)), \tag{3.1b}$$
$$s = W''(u)q, \tag{3.1c}$$
$$p_1 = \epsilon^2 \nabla \cdot \boldsymbol{p}_2, \tag{3.1d}$$
$$\boldsymbol{p}_2 = \nabla q, \tag{3.1e}$$
$$q = \epsilon^2 \nabla \cdot \boldsymbol{w} - r, \tag{3.1f}$$
$$\boldsymbol{w} = \nabla u, \tag{3.1g}$$
$$r = W'(u). \tag{3.1h}$$

To simplify the notation, we still use $u, \boldsymbol{p}, s, p_1, \boldsymbol{p}_2, q, \boldsymbol{w}, r$ to denote the numerical solution. Applying the LDG method to the system (3.1), we have the scheme: Find $u, s, p_1, q, r \in V_h^k$ and $\boldsymbol{p}, \boldsymbol{p}_2, \boldsymbol{w} \in \Sigma_h^k$, such that, for all test functions $\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5 \in V_h^k$ and $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3 \in \Sigma_h^k$, we have

$$\int_K u_t \varphi_1 dK = -\int_K \boldsymbol{p} \cdot \nabla \varphi_1 dK + \int_{\partial K} \widehat{\boldsymbol{p} \cdot \boldsymbol{\nu}} \varphi_1 ds, \tag{3.2a}$$
$$\int_K \boldsymbol{p} \cdot \boldsymbol{\theta_1} dK = -\int_K (p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r))(\nabla \cdot \boldsymbol{\theta_1}) dK + \int_{\partial K} (\hat{p}_1 - \hat{s} + \epsilon(\eta_1 \hat{q} + (\eta_1 - \eta_2)\hat{r}))(\boldsymbol{\theta}_1 \cdot \boldsymbol{\nu}) ds, \tag{3.2b}$$
$$\int_K s \varphi_2 dK = \int_K W''(u) q \varphi_2 dK, \tag{3.2c}$$
$$\int_K p_1 \varphi_3 dK = -\epsilon^2 \int_K \boldsymbol{p}_2 \cdot \nabla \varphi_3 dK + \epsilon^2 \int_{\partial K} \widehat{\boldsymbol{p}_2 \cdot \boldsymbol{\nu}} \varphi_3 ds, \tag{3.2d}$$
$$\int_K \boldsymbol{p}_2 \cdot \boldsymbol{\theta_2} dK = -\int_K q(\nabla \cdot \boldsymbol{\theta_2}) dK + \int_{\partial K} \hat{q}(\boldsymbol{\theta_2} \cdot \boldsymbol{\nu}) ds, \tag{3.2e}$$
$$\int_K q \varphi_4 dK = -\epsilon^2 \int_K \boldsymbol{w} \cdot \nabla \varphi_4 dK + \epsilon^2 \int_{\partial K} \widehat{\boldsymbol{w} \cdot \boldsymbol{\nu}} \varphi_4 ds - \int_K r \varphi_4 dK, \tag{3.2f}$$
$$\int_K \boldsymbol{w} \cdot \boldsymbol{\theta_3} dK = -\int_K u(\nabla \cdot \boldsymbol{\theta_3}) dK + \int_{\partial K} \hat{u}(\boldsymbol{\theta_3} \cdot \boldsymbol{\nu}) ds, \tag{3.2g}$$
$$\int_K r \varphi_5 dK = \int_K W'(u) \varphi_5 dK. \tag{3.2h}$$

The "hat" terms in (3.2) in the cell boundary terms from integration by parts are the so-called "numerical fluxes", which are functions defined on the edges and should be designed based on different guiding principles for different PDEs to ensure stability. Similar to the development in [26], it turns out that we can take the simple choices such that

$$\hat{\boldsymbol{p}}|_e = \boldsymbol{p}_L, \quad \hat{p}_1|_e = p_{1R}, \quad \hat{s}|_e = s_R, \quad \hat{q}|_e = q_R,$$
$$\hat{r}|_e = r_R, \quad \hat{\boldsymbol{p}}_2|_e = \boldsymbol{p}_{2L}, \quad \hat{\boldsymbol{w}}|_e = \boldsymbol{w}_L, \quad \hat{u}|_e = u_R. \tag{3.3}$$

We remark that the choice for the fluxes (3.3) is not unique.

By the boundary conditions (2.6), we take

$$\begin{aligned} &\hat{\boldsymbol{p}} = 0, \quad \hat{\boldsymbol{p}}_2 = 0, \quad \hat{\boldsymbol{w}} = 0, \quad \hat{p}_1 = p_1^{\text{in}}, \\ &\hat{s} = s^{\text{in}}, \quad \hat{q} = q^{\text{in}}, \quad \hat{r} = r^{\text{in}}, \quad \hat{u} = u^{\text{in}}, \end{aligned} \tag{3.4}$$

at the domain boundary, where u^{in} means the value taking from the inside of the boundary element.

3.3 Energy Stability

In this subsection, we will prove the energy stability of the LDG scheme for the general nonlinear FCH equation with the choice of the fluxes in the previous subsection.

Proposition 3.1 (Energy stability) *The solution to the LDG Scheme (3.2) and the flux (3.3) with the boundary flux (3.4) satisfies the energy stability*

$$\frac{d}{dt}\int_\Omega \left(\frac{1}{2}q^2 - \epsilon\left(\frac{\epsilon^2\eta_1}{2}\boldsymbol{w}\cdot\boldsymbol{w} + \eta_2 W(u)\right)\right)d\Omega \le 0.$$

Proof Choosing the test function $\varphi_5 = \epsilon\eta_2 u_t$ in (3.2h), we obtain

$$\epsilon\eta_2\int_K r u_t dK = \epsilon\eta_2\int_K W'(u)u_t dK. \tag{3.5}$$

After taking the time derivative to Eqs. (3.2f), (3.2g) and (3.2h), we choose the test functions $\varphi_4 = q$, $\boldsymbol{\theta_3} = -\epsilon^3\eta_1\boldsymbol{w}$, $\boldsymbol{\theta_3} = -\epsilon^2\boldsymbol{p}_2$ and $\varphi_5 = -q$ in (3.2f), (3.2g) and (3.2h), respectively. Then we get

$$\int_K q_t q dK = -\epsilon^2\int_K \boldsymbol{w}_t\cdot\nabla q dK + \epsilon^2\int_{\partial K}(\widehat{\boldsymbol{w}\cdot\boldsymbol{v}})_t q ds - \int_K r_t q dK, \tag{3.6}$$

$$-\epsilon^3\eta_1\int_K \boldsymbol{w}_t\cdot\boldsymbol{w}dK = \epsilon^3\eta_1\int_K u_t(\nabla\cdot\boldsymbol{w})dK - \epsilon^3\eta_1\int_{\partial K}\hat{u}_t(\boldsymbol{w}\cdot\boldsymbol{v})ds, \tag{3.7}$$

$$-\epsilon^2\int_K \boldsymbol{w}_t\cdot\boldsymbol{p_2}dK = \epsilon^2\int_K u_t(\nabla\cdot\boldsymbol{p_2})dK - \epsilon^2\int_{\partial K}\hat{u}_t(\boldsymbol{p_2}\cdot\boldsymbol{v})ds, \tag{3.8}$$

$$-\int_K r_t q dK = -\int_K W''(u)u_t q dK. \tag{3.9}$$

For (3.2a)–(3.2f), we take the test functions

$$\begin{aligned} &\varphi_1 = p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r), \quad \boldsymbol{\theta_1} = \boldsymbol{p}, \quad \varphi_2 = u_t, \\ &\varphi_3 = -u_t, \quad \boldsymbol{\theta_2} = \epsilon^2\boldsymbol{w}_t, \quad \varphi_4 = -\epsilon\eta_1 u_t. \end{aligned}$$

Then we have

$$\begin{aligned} &\int_K u_t(p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r))dK \\ &= -\int_K \boldsymbol{p}\cdot\nabla(p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r))dK \\ &\quad + \int_{\partial K}\widehat{\boldsymbol{p}\cdot\boldsymbol{v}}(p_1 - s + \epsilon(\eta_1 q + (\eta_1 - \eta_2)r))ds, \end{aligned} \tag{3.10}$$

$$\int_K \boldsymbol{p}\cdot\boldsymbol{p}dK = -\int_K (p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))(\nabla\cdot\boldsymbol{p})dK + \int_{\partial K}(\hat{p}_1 - \hat{s} + \epsilon(\eta_1\hat{q} + (\eta_1-\eta_2)\hat{r}))(\boldsymbol{p}\cdot\boldsymbol{\nu})ds, \tag{3.11}$$

$$\int_K s u_t dK = \int_K W''(u) q u_t dK, \tag{3.12}$$

$$-\int_K p_1 u_t dK = \epsilon^2\int_K \boldsymbol{p}_2\cdot\nabla u_t dK - \epsilon^2\int_{\partial K}\widehat{\boldsymbol{p}_2\cdot\boldsymbol{\nu}}u_t ds, \tag{3.13}$$

$$\epsilon^2\int_K \boldsymbol{p}_2\cdot\boldsymbol{w}_t dK = -\epsilon^2\int_K q(\nabla\cdot\boldsymbol{w}_t)dK + \epsilon^2\int_{\partial K}\hat{q}(\boldsymbol{w}_t\cdot\boldsymbol{\nu})ds, \tag{3.14}$$

$$-\epsilon\eta_1\int_K q u_t dK = \epsilon\eta_1\int_K r u_t dK + \epsilon^3\eta_1\int_K \boldsymbol{w}\cdot\nabla u_t dK - \epsilon^3\eta_1\int_{\partial K}\widehat{\boldsymbol{w}\cdot\boldsymbol{\nu}}u_t ds. \tag{3.15}$$

Summing up the Eqs. (3.5)–(3.15), we obtain

$$\begin{aligned}
&\int_K \left(q_t q - \epsilon^3\eta_1\boldsymbol{w}\cdot\boldsymbol{w}_t - \epsilon\eta_2 W'(u)u_t + \boldsymbol{p}\cdot\boldsymbol{p}\right)dK\\
&= -\int_K \boldsymbol{p}\cdot\nabla(p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))dK\\
&\quad -\int_K (p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))(\nabla\cdot\boldsymbol{p})dK\\
&\quad +\int_{\partial K}\widehat{\boldsymbol{p}\cdot\boldsymbol{\nu}}(p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))ds\\
&\quad +\int_{\partial K}(\hat{p}_1 - \hat{s} + \epsilon(\eta_1\hat{q} + (\eta_1-\eta_2)\hat{r}))(\boldsymbol{p}\cdot\boldsymbol{\nu})ds + \epsilon^2\int_K u_t(\nabla\cdot\boldsymbol{p_2})dK\\
&\quad +\epsilon^2\int_K \boldsymbol{p}_2\cdot\nabla u_t dK - \epsilon^2\int_{\partial K}\hat{u}_t(\boldsymbol{p_2}\cdot\boldsymbol{\nu})ds - \epsilon^2\int_{\partial K}\widehat{\boldsymbol{p}_2\cdot\boldsymbol{\nu}}u_t ds\\
&\quad -\epsilon^2\int_K \boldsymbol{w}_t\cdot\nabla q dK - \epsilon^2\int_K q(\nabla\cdot\boldsymbol{w}_t)dK + \epsilon^2\int_{\partial K}(\widehat{\boldsymbol{w}\cdot\boldsymbol{\nu}})_t q ds\\
&\quad +\epsilon^2\int_{\partial K}\hat{q}(\boldsymbol{w}_t\cdot\boldsymbol{\nu})ds + \epsilon^3\eta_1\int_K u_t(\nabla\cdot\boldsymbol{w})dK + \epsilon^3\eta_1\int_K \boldsymbol{w}\nabla u_t dK\\
&\quad -\epsilon^3\eta_1\int_{\partial K}\hat{u}_t(\boldsymbol{w}\cdot\boldsymbol{\nu})ds - \epsilon^3\eta_1\int_{\partial K}\widehat{\boldsymbol{w}\cdot\boldsymbol{\nu}}u_t ds\\
&= -\int_{\partial K}(p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))(\boldsymbol{p}\cdot\boldsymbol{\nu})ds\\
&\quad +\int_{\partial K}\widehat{\boldsymbol{p}\cdot\boldsymbol{\nu}}(p_1 - s + \epsilon(\eta_1 q + (\eta_1-\eta_2)r))ds\\
&\quad +\int_{\partial K}(\hat{p}_1 - \hat{s} + \epsilon(\eta_1\hat{q} + (\eta_1-\eta_2)\hat{r}))(\boldsymbol{p}\cdot\boldsymbol{\nu})ds + \epsilon^2\int_{\partial K}(\boldsymbol{p}_2\cdot\boldsymbol{\nu})u_t ds\\
&\quad -\epsilon^2\int_{\partial K}\hat{u}_t(\boldsymbol{p_2}\cdot\boldsymbol{\nu})ds - \epsilon^2\int_{\partial K}\widehat{\boldsymbol{p}_2\cdot\boldsymbol{\nu}}u_t ds - \epsilon^2\int_{\partial K}q(\boldsymbol{w}_t\cdot\boldsymbol{\nu})ds\\
&\quad +\epsilon^2\int_{\partial K}(\widehat{\boldsymbol{w}\cdot\boldsymbol{\nu}})_t q ds + \epsilon^2\int_{\partial K}\hat{q}(\boldsymbol{w}_t\cdot\boldsymbol{\nu})ds + \epsilon^3\eta_1\int_{\partial K}(\boldsymbol{w}\cdot\boldsymbol{\nu})u_t ds\\
&\quad -\epsilon^3\eta_1\int_{\partial K}\hat{u}_t(\boldsymbol{w}\cdot\boldsymbol{\nu})ds - \epsilon^3\eta_1\int_{\partial K}\widehat{\boldsymbol{w}\cdot\boldsymbol{\nu}}u_t ds
\end{aligned}$$

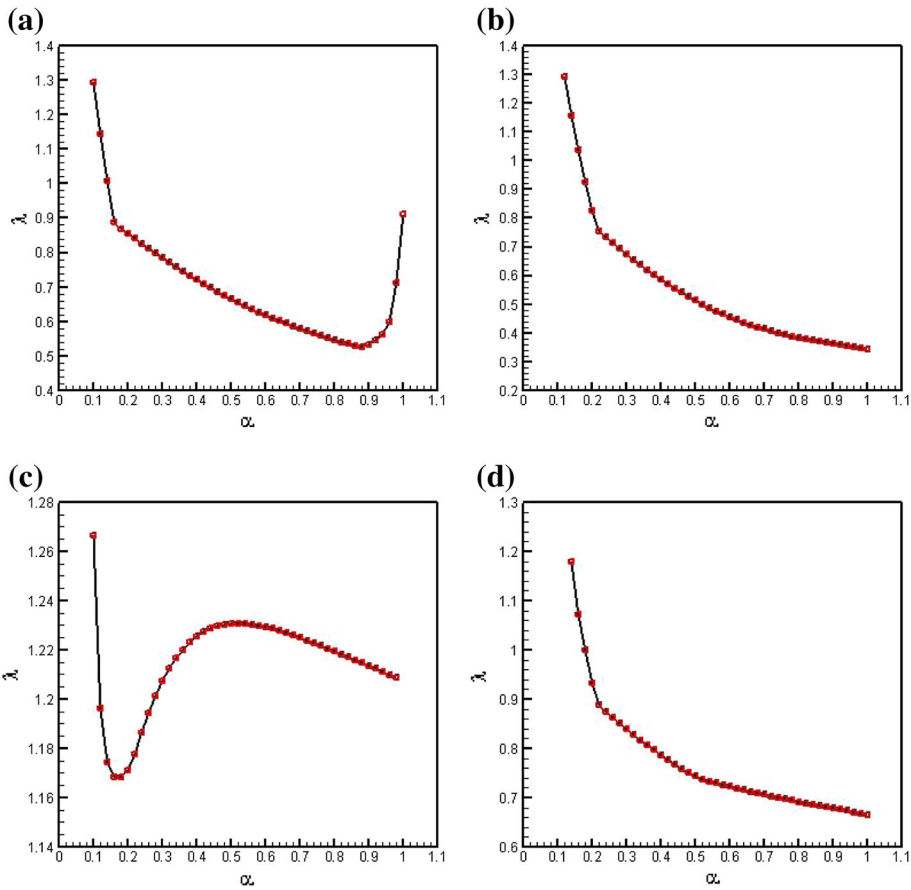

Fig. 1 The asymptotic convergence factor λ changes with the damping parameter α. **a** Jacobi smoother, $\mathcal{P}^1$, **b** Gauss–Seidel smoother, $\mathcal{P}^1$, **c** Jacobi smoother, $\mathcal{P}^2$, **d** Gauss–Seidel smoother, $\mathcal{P}^2$

Summing up over K, with the numerical fluxes (3.3) and the boundary conditions (3.4), we can cancel the boundary terms and get

$$\int_\Omega \left(q_t q - \epsilon^3 \eta_1 \boldsymbol{w}\cdot\boldsymbol{w}_t - \epsilon\eta_2 W'(u)u_t + \boldsymbol{p}\cdot\boldsymbol{p}\right) d\Omega = 0.$$

Because

$$\int_\Omega \boldsymbol{p}\cdot\boldsymbol{p} d\Omega \geq 0,$$

we obtain the energy stability

$$\frac{d}{dt}\int_\Omega \left(\frac{1}{2}q^2 - \epsilon\left(\frac{\epsilon^2\eta_1}{2}\boldsymbol{w}\cdot\boldsymbol{w} + \eta_2 W(u)\right)\right) d\Omega \leq 0.$$

□

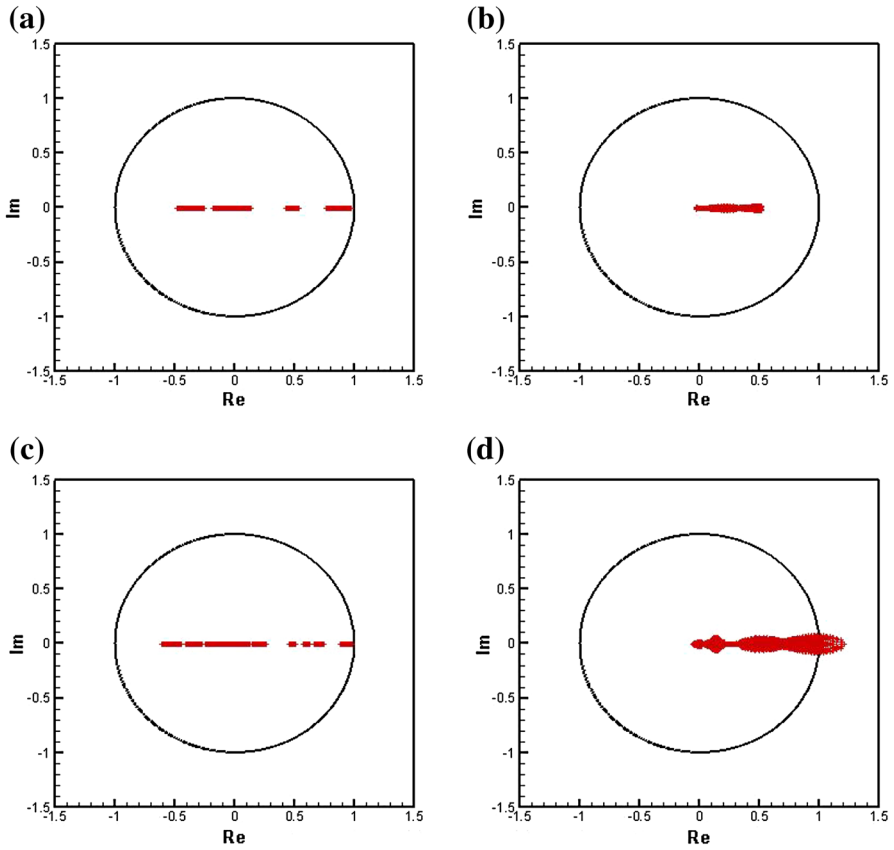

Fig. 2 Eigenvalue spectral of E_h and E_h^{2grid} with damped Jacobi smoother for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. **a** Eigenvalue spectral of $E_h, \alpha = 0.85, \mathcal{P}^1$, **b** eigenvalue spectral of $E_h^{2grid}, \alpha = 0.85, \mathcal{P}^1$, **c** eigenvalue spectral of $E_h, \alpha = 0.85, \mathcal{P}^2$, **d** eigenvalue spectral of $E_h^{2grid}, \alpha = 0.85, \mathcal{P}^2$

4 The Semi-implicit Time Discretization Method

The LDG spatial discretization for the FCH equation typically results in an ordinary differential equation (ODE). Explicit time marching methods require a suitably small time step ($\Delta t \sim O(\Delta x^6)$) for stability, so we will explore a semi-implicit time marching method to remove the severe time step restriction. Due to the high order spatial derivatives for the FCH equation, the system of equations arising by the LDG spatial discretization and semi-implicit time marching method will be ill-conditioned. How to solve these equations efficiently and accurately present a great challenge.

Following the method in [4], we introduce a semi-implicit time marching method to remove the time step restriction for explicit methods. The semi-implicit scheme is

$$\frac{u^{n+1} - u^n}{\Delta t} = \nabla \cdot \boldsymbol{p}^{n+1}, \tag{4.1a}$$

$$\boldsymbol{p}^{n+1} = \nabla(p_1^{n+1} - s^n + \epsilon(\eta_1 q^{n+1} + (\eta_1 - \eta_2) r^n)) + s_1(q^{n+1} - q^n), \tag{4.1b}$$

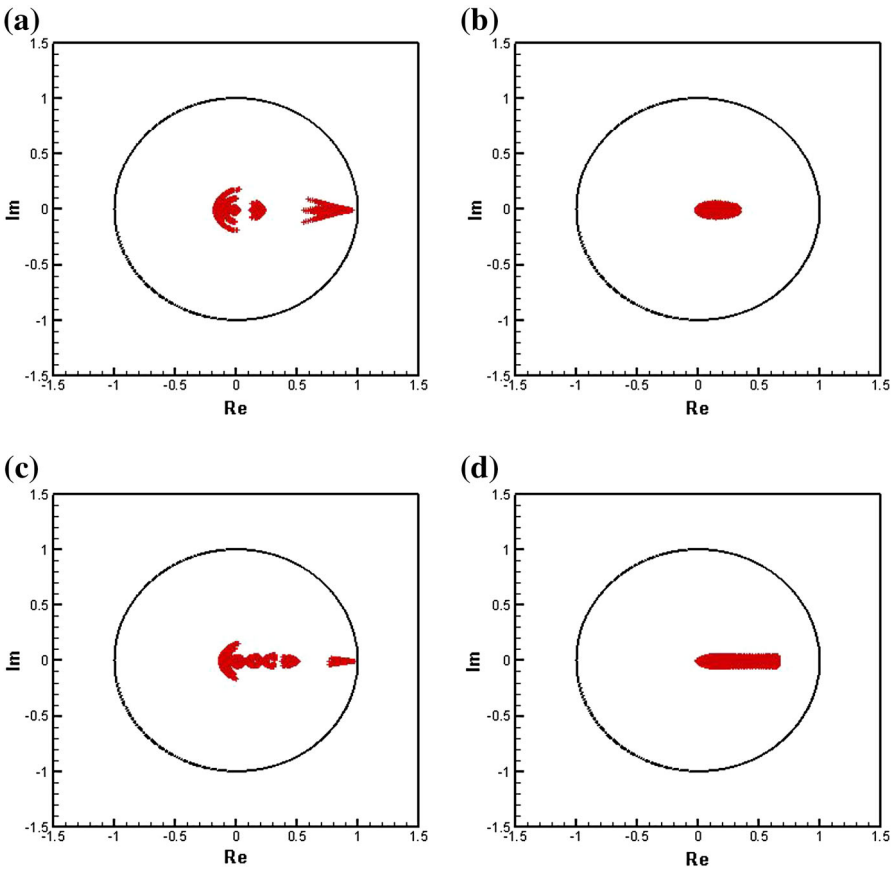

Fig. 3 Eigenvalue spectral of E_h and E_h^{2grid} with Gauss–Seidel smoother for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation. **a** Eigenvalue spectral of E_h, $\alpha = 1.0$, $\mathcal{P}^1$, **b** eigenvalue spectral of E_h^{2grid}, $\alpha = 1.0$, $\mathcal{P}^1$, **c** eigenvalue spectral of E_h, $\alpha = 1.0$, $\mathcal{P}^2$, **d** eigenvalue spectral of E_h^{2grid}, $\alpha = 1.0$, $\mathcal{P}^2$

$$s^n = W''(u^n)q^n, \tag{4.1c}$$
$$p_1^{n+1} = \epsilon^2 \nabla \cdot \boldsymbol{p}_2{}^{n+1}, \tag{4.1d}$$
$$\boldsymbol{p}_2{}^{n+1} = \nabla q^{n+1}, \tag{4.1e}$$
$$q^{n+1} = \epsilon^2 \nabla \cdot \boldsymbol{w}^{n+1} - r^n + s_2(u^{n+1} - u^n), \tag{4.1f}$$
$$\boldsymbol{w}^{n+1} = \nabla u^{n+1}, \tag{4.1g}$$
$$r^n = W'(u^n), \tag{4.1h}$$

where

$$q^n = \epsilon^2 \nabla \cdot \boldsymbol{w}^n - r^n, \tag{4.2a}$$
$$\boldsymbol{w}^n = \nabla u^n, \tag{4.2b}$$

Table 1 The number of MG iterations required to reduce the norm of the residual below the tolerance $\tau = 1.0 \times 10^{-8}$

$\Delta t = 1.0 \times 10^{-3}$

Δx	Jacobi smoother	Gauss–Seidel smoother
$2\pi/32$	10	7
$2\pi/64$	10	7
$2\pi/128$	10	7
$2\pi/256$	10	7

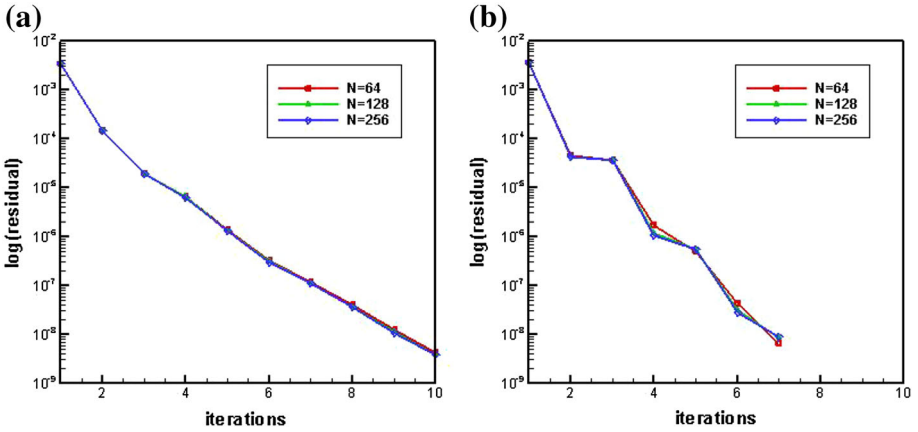

Fig. 4 Convergence rates of MG solver for $\mathcal{P}^1$ approximation. $\Delta t = 1.0 \times 10^{-3}$. **a** Damped Jacobi smoother with $\alpha = 0.85$, **b** Gauss–Seidel smoother, no damping

s_1 and s_2 are two suitable stabilization constants. Numerical experiments in [4] show that the Scheme (4.1) is unconditionally stable with $s_1 = s_2 = 2$. This allows for adaptive time stepping.

Given u^n, p_1^n and q^n, the algorithm to get u^{n+1} , p_1^{n+1} and q^{n+1} is

1. Choosing a local basis in cell K, we can eliminate $\boldsymbol{p}^{n+1}$, s^n, $\boldsymbol{p_2}^{n+1}$, $\boldsymbol{w}^{n+1}$, $\boldsymbol{w}^n$ and r^n from Eqs. (4.1b), (4.1c), (4.1e), (4.1g), (4.2b) and (4.1h), respectively, by simply inverting a small mass matrix in each case.
2. After the LDG spatial discretization, we get a system of linear equations

$$\begin{cases} u^{n+1} = L_1(p_1^{n+1}, q^n, q^{n+1}, u^n), \\ p_1^{n+1} = L_2(q^{n+1}), \\ q^{n+1} = L_3(u^{n+1}, u^n). \end{cases} \tag{4.3}$$

3. Solving the system of three coupled second-order Eq. (4.3) for $\{u^{n+1}, p_1^{n+1}, q^{n+1}\}$ at each time step.

The overall performance highly depends on the efficiency of the solver. Traditional iterative methods such as Gauss–Seidel method suffers from slow convergence rates and the MG method is demonstrated as an efficient solver. We will show numerically that the number of iterations is independent of the problem size. For a detailed description of the MG algorithm, we refer the readers to Appendix 1.

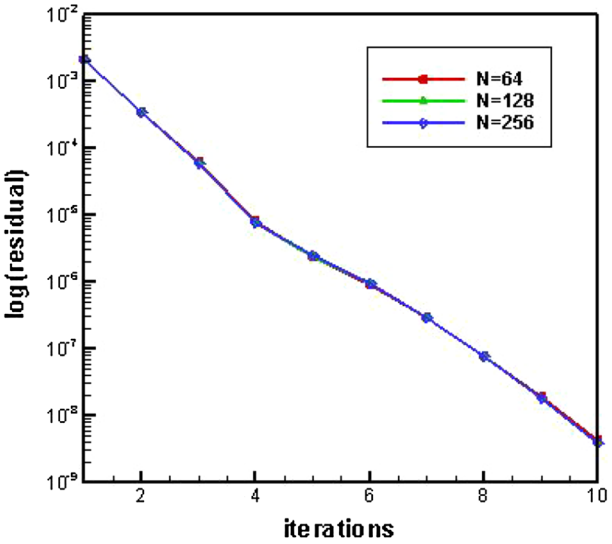

Fig. 5 Convergence rates of MG solver for $\mathcal{P}^2$ approximation with Gauss–Seidel smoother (no damping). $\Delta t = 1.0 \times 10^{-3}$

Table 2 Accuracy test for the Scheme (4.1) at $t = 0.2$. To maintain the accuracy, the time step is chosen as $\Delta t = 0.2\Delta x$, $\Delta t = 0.2\Delta x^2$ and $\Delta t = 0.2\Delta x^3$ for $\mathcal{P}^0$, $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation, respectively

	N	L^2 Error	Order	L^∞ error	Order
$\mathcal{P}^0$	16	4.54E−01	–	1.76E−01	–
	32	2.43E−01	0.90	9.69E−02	0.87
	64	1.31E−01	0.89	5.28E−02	0.87
	128	6.88E−02	0.93	2.76E−02	0.93
$\mathcal{P}^1$	16	1.22E−01	–	6.19E−02	–
	32	3.91E−02	1.64	1.67E−02	1.89
	64	1.05E−02	1.89	4.29E−03	1.96
	128	2.71E−03	1.97	1.08E−03	1.98
$\mathcal{P}^2$	16	5.50E−02	–	1.52E−02	–
	32	8.06E−03	2.77	2.14E−03	2.83
	64	1.03E−03	2.96	2.73E−04	2.97
	128	1.34E−04	2.94	3.47E−05	2.98

Table 3 Accuracy test for the Scheme (4.1) at $t = 0.2$. The time step is chosen as $\Delta t = 0.2\Delta x$ for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation

	N	L^2 error	Order	L^∞ error	Order
$\mathcal{P}^1$	32	2.65E−01	–	7.40E−02	–
	64	1.59E−01	0.74	4.01E−02	0.88
	128	8.90E−02	0.84	2.13E−02	0.91
$\mathcal{P}^2$	32	2.65E−01	–	6.71E−02	–
	64	1.59E−01	0.74	3.84E−02	0.80
	128	8.90E−02	0.84	2.09E−02	0.88

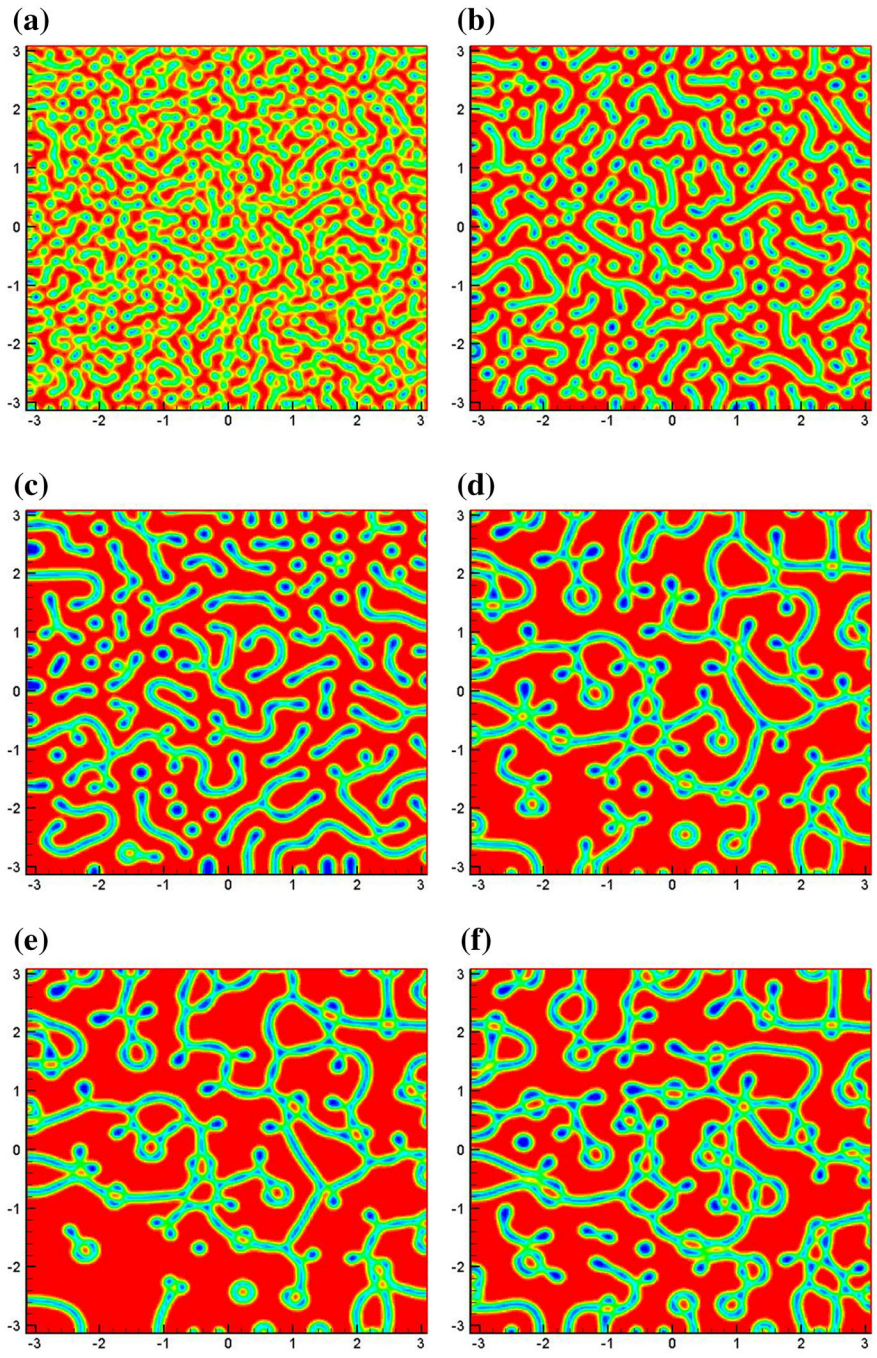

Fig. 6 The time evolution of the FCH equation implemented with Scheme (4.1) with $\mathcal{P}^2$ elements for Example 6.2. **a** $T = 0.01,\ \Delta t = 0.0001$, **b** $T = 0.1,\ \Delta t = 0.0001$, **c** $T = 1.0,\ \Delta t = 0.0001$, **d** $T = 100.0,\ \Delta t = 0.01$, **e** $T = 200.0,\ \Delta t = 0.01$, **f** $T = 400.0,\ \Delta t = 0.1$

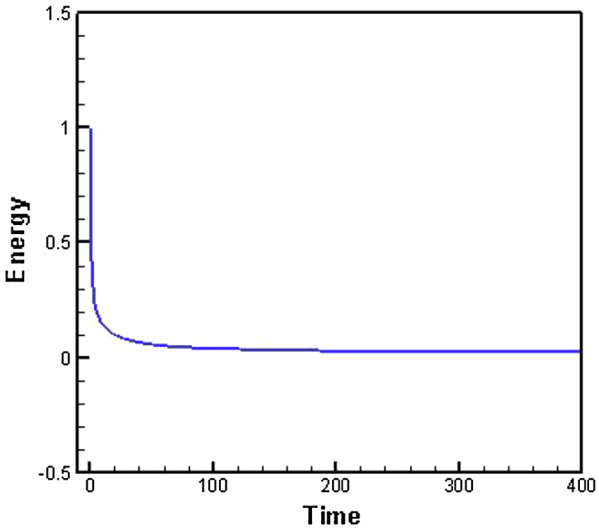

Fig. 7 Energy trace of numerical solution for Example 6.2

5 Local Mode Analysis of the Two-Level Algorithm

In order to predict the MG behavior, a two-level local mode analysis is introduced to study the convergence of the MG method. At the core of any MG algorithm is the two-level algorithm. Multilevel methods are obtained by recursively applying the two-level algorithm. The convergence behavior of the two-level algorithm is given by the eigenvalue spectral of the error amplification operator, i.e. $\rho(E_h^{2grid})$.

For convenience, we consider the local mode analysis for the Scheme (4.1) in two dimension. Extension to higher dimensions follows immediately by means of the tensor-product principle. For a detailed description of the local mode analysis for the two-grid algorithm, we refer the readers to Appendix 2.

We choose $\epsilon = 0.03$, $\eta_1 = \eta_2 = 5\epsilon$ and $s_1 = s_2 = 2$ in Scheme (4.1) and the eigenvalue spectral of the smoother (E_h) and the two-level algorithm (E_h^{2grid}) are shown in Figs. 1, 2, 3. From these figures, we have

1. From Fig. 1, we choose $\alpha = 0.85$ with Jacobi smoother and $\alpha = 1.0$ (no damping) with Gauss–Seidel smoother for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation.
2. From Figs. 2, 3, we can see that the two-level algorithm with Gauss–Seidel smoother has better convergence behavior than with Jacobi smoother.
3. The two-level algorithm is not convergent for $\mathcal{P}^2$ approximation with damped Jacobi smoother according to Fig. 2.

6 Numerical Results

In this section, we discuss the numerical solution of the Scheme (4.1) with LDG spatial discretization. The resulting linear system is solved by the multigrid method and we show numerically that the method has mesh-independent convergence rates. For the spatial discretization we use uniform meshes. In our numerical experiments, the values of the number

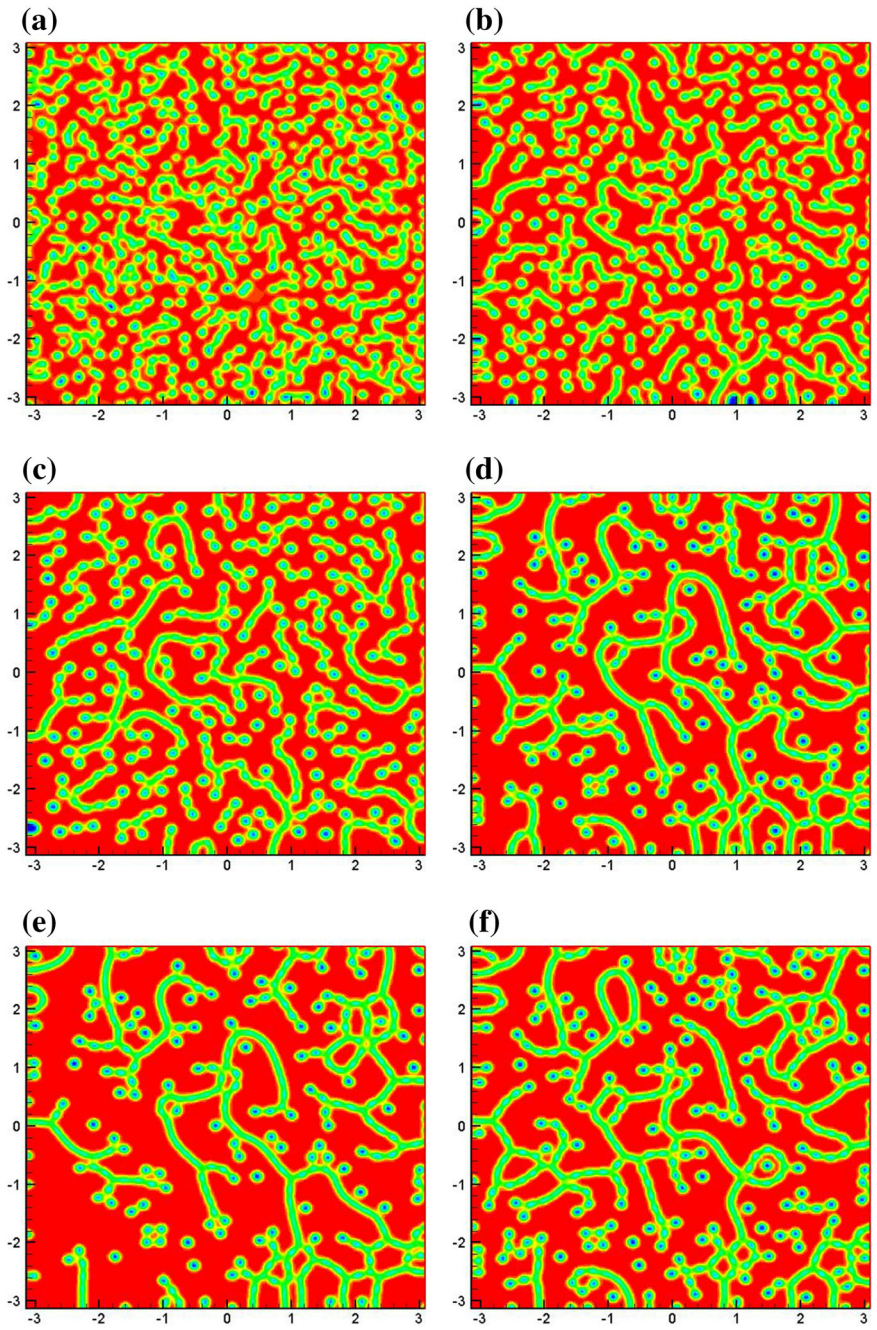

Fig. 8 The time evolution of the FCH equation implemented with Scheme (4.1) with $\mathcal{P}^2$ elements for Example 6.3. **a** $T = 0.01\Delta t = 0.0001$, **b** $T = 0.1\Delta t = 0.0001$, **c** $T = 1.0\Delta t = 0.0001$, **d** $T = 100.0\Delta t = 0.01$, **e** $T = 200.0\Delta t = 0.01$, **f** $T = 400.0\Delta t = 0.1$

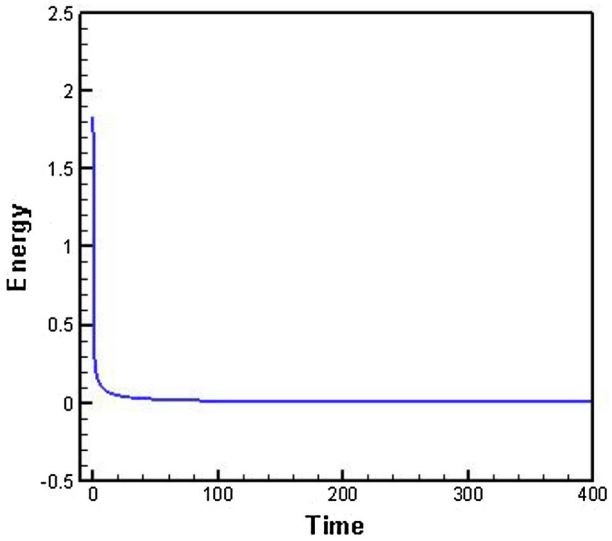

Fig. 9 Energy trace of numerical solution for Example 6.3

of pre- and post- relaxations is taken as $\nu_1 = \nu_2 = 3$. All the computations are performed in double precision.

6.1 Two Space Dimension

6.1.1 Convergence of the MG Solver

Example 6.1 To demonstrate the superiority of the multigrid solver, we present the convergence rate of the method at the 10th time step. For the tests we take the exact solution of

$$u(x, y, t) = e^{-2t} \sin(x) \sin(y), \tag{6.1}$$

with the source term $f(x, y, t)$, where $f(x, y, t)$ is a given function so that make the exact solution. The initial condition is

$$u(x, y, 0) = \sin(x) \sin(y), \tag{6.2}$$

and the domain is $[-\pi, \pi] \times [-\pi, \pi]$ with periodic boundary conditions. $W(u)$ is taken as (2.3). The values of the parameters are $\epsilon = 0.4$ and $\eta_1 = \eta_2 = 1.0$.

The number of multigrid iterations required to reduce the norm of the residual below the tolerance $\tau = 1.0 \times 10^{-8}$ for $\mathcal{P}^1$ approximation is given in Table 1 with Gauss–Seidel smoother and Jacobi smoother. We can find that the required number of iterations is independent of Δx and the multigrid method with Gauss–Seidel smoother shows better convergence behavior than with damped Jacobi smoother, which agrees with the results of the local mode analysis in Sect. 5, thus we further restrict our study to the former. By the local mode analysis in Sect. 5, we know that the two-grid method with Jacobi smoother is not convergent for $\mathcal{P}^2$ approximation, so we only consider Gauss–Seidel smoother here. Figures 4 and 5 suggest that the multigrid solver is of optimal complexity for $\mathcal{P}^1$ and $\mathcal{P}^2$ approximation.

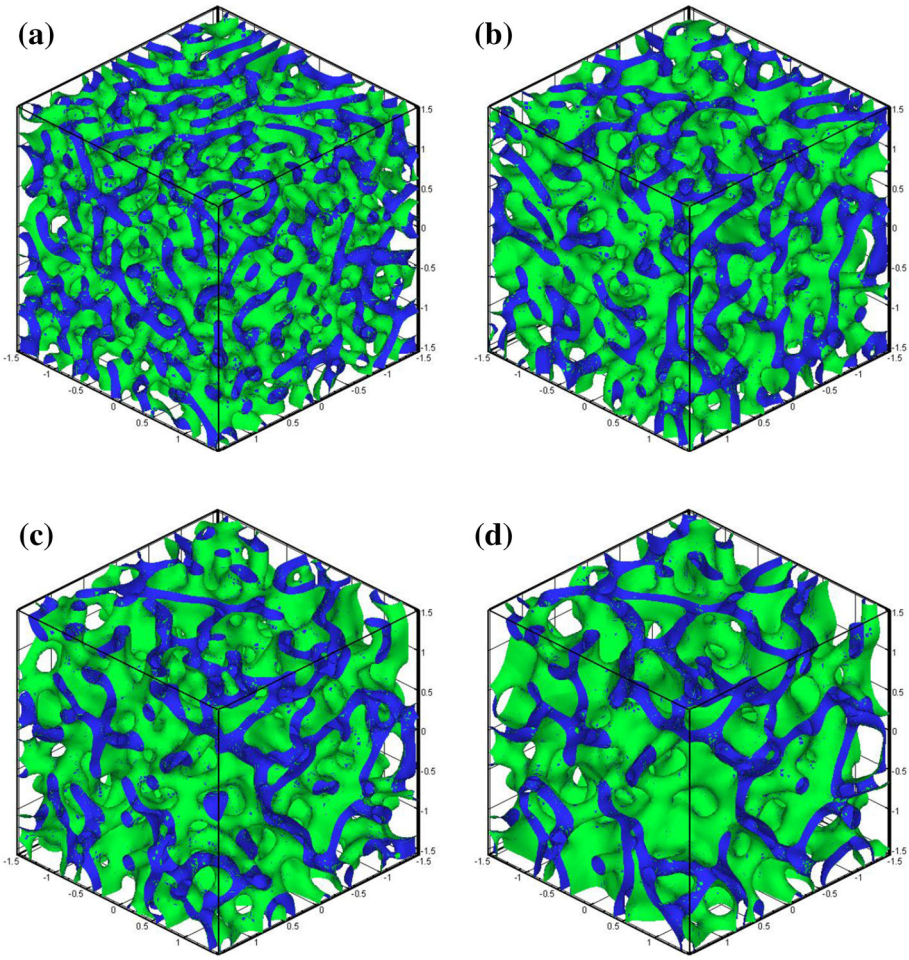

Fig. 10 Level sets $u = 0.4$ (*green*) and $u = 0.45$ (*blue*) of the solvent phase resulting from the FCH equation implemented with Scheme (4.1) with $\mathcal{P}^2$ elements for Example 6.4. **a** $T = 0.1\Delta t = 0.002$, **b** $T = 1.0\Delta t = 0.002$, **c** $T = 50.0\Delta t = 0.1$, **d** $T = 200.0\Delta t = 0.1$ (Color figure online)

The L^2 and L^∞ errors and the numerical orders of accuracy at time $t = 0.2$ can be found in Table 2. We can see that the method with $\mathcal{P}^k$ elements gives a $(k+1)$-th order of accuracy in both L^2 and L^∞ norms. What we should keep in mind is that the choice of this refinement path has nothing to do with any time step restriction for stability. In Table 3, we also show the convergence results for $\mathcal{P}^1$ and $\mathcal{P}^2$ elements with large time step $\Delta t = 0.2\Delta x$, which indicate the time discretization scheme is consistent.

6.1.2 Spinodal Decomposition and Energy Dissipation

Example 6.2 We consider the Scheme (4.1) with LDG spatial discretization. The initial data is a random number between -1 and 1 on each grid point. The domain is $[-\pi, \pi] \times [-\pi, \pi]$ and the boundary conditions are (2.6). We take $\epsilon = 0.03$ and $\eta_1 = \eta_2 = 5\epsilon$, and $W(u)$ is taken as (2.3). Numerical results of Scheme (4.1) for $\mathcal{P}^2$ approximation are shown in Fig. 6.

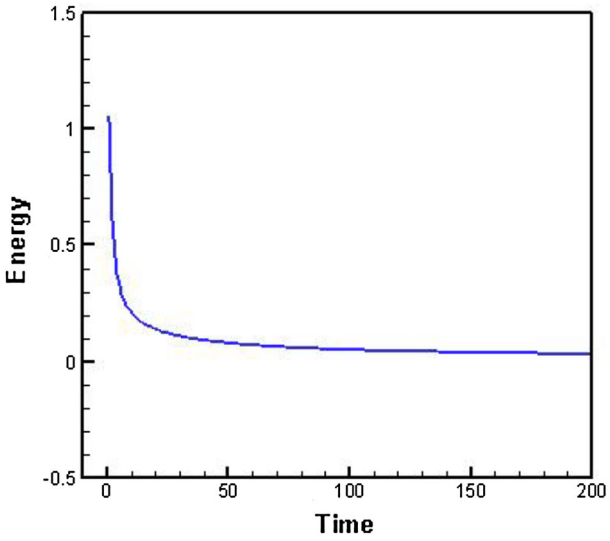

Fig. 11 Energy trace of numerical solution for Example 6.4

Figure 6 shows statistically similar patterns in the numerical solution as those in [20,24]. The system experiences rapid mixing of the two components in the early stage, and phase separation occurs on nearly the same time scale (the spinodal phase). The porous structure forms after a very short time ($T = 0.01$), followed by merging and further phase separation. On longer time scales, the system evolves slowly. The results are qualitatively comparable to the results reported in [17,20].

The energy trace of the numerical solution is shown in Fig. 7 and we find that the energy (2.2) of FCH equation decays with respect to time t, which agrees with the theoretical results.

Example 6.3 We consider the FCH equation (2.5) with

$$W(u) = \frac{(u-1)^2}{2}\left(\frac{(u+1)^2}{2} - 0.18\right), \tag{6.3}$$

$\eta_1 = 10\epsilon$, $\eta_2 = 5\epsilon$ and $\epsilon = 0.03$. The initial condition is taken as

$$u(x, y, 0) = \text{sign}(\delta + vf), \tag{6.4}$$

where $vf = -0.2$ and δ is randomly assigned between 0 and 1. The domain is $[-\pi, \pi] \times [-\pi, \pi]$ and the boundary conditions are (2.6). Numerical results of Scheme (4.1) for $\mathcal{P}^2$ approximation are shown in Fig. 8, which is qualitatively consistent with those presented in [20].

Figure 9 shows the energy trace of the numerical solution and we can see that the energy (2.2) of FCH equation decays with respect to time t.

6.2 Three Space Dimension

6.2.1 Spinodal Decomposition and Energy Dissipation

Example 6.4 We consider the FCH equation (2.5) with $\eta_1 = \eta_2 = 5\epsilon$ and $\epsilon = 0.03$. $W(u)$ is taken as (2.3). The initial data is a random number between -1 and 1 on each grid point.

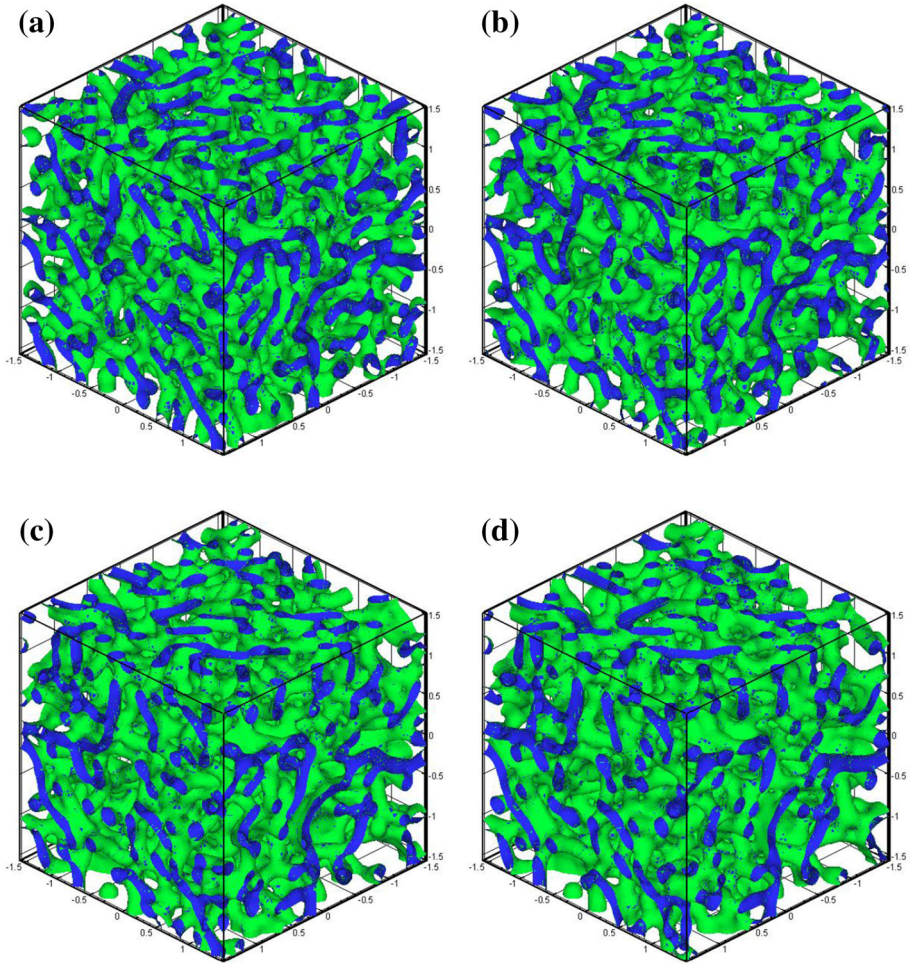

Fig. 12 Level sets $u = 0.4$ (*green*) and $u = 0.45$ (*blue*) of the solvent phase resulting from the FCH equation implemented with Scheme (4.1) with $\mathcal{P}^2$ elements for Example 6.5. **a** $T = 0.1\Delta t = 0.002$, **b** $T = 1.0\Delta t = 0.002$, **c** $T = 50.0\Delta t = 0.2$, **d** $T = 200.0\Delta t = 1.0$ (Color figure online)

The domain is $[-\pi/2, \pi/2] \times [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$ and the boundary conditions are (2.6). Numerical results of Scheme (4.1) for $\mathcal{P}^2$ approximation are shown in Fig. 10. The system experiences rapid mixing of the two components in the early stage, and phase separation occurs on nearly the same time scale (the spinodal phase). After a short time, the porous structure appears.

Figure 11 shows the energy trace of the numerical solution and we can see that the energy (2.2) of FCH equation decays with respect to time t, which agrees with the theoretical result.

Example 6.5 We consider the FCH equation (2.5) with

$$W(u) = \frac{(u-1)^2}{2}\left(\frac{(u+1)^2}{2} - 0.18\right), \tag{6.5}$$

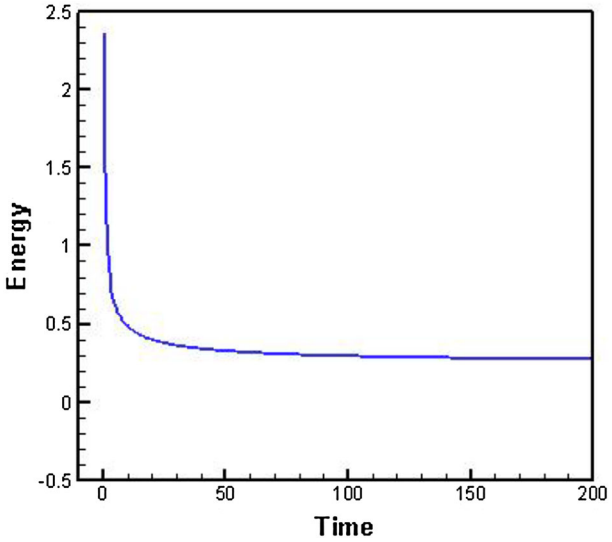

Fig. 13 Energy trace of numerical solution for Example 6.5

$\eta_1 = 10\epsilon$, $\eta_2 = 5\epsilon$ and $\epsilon = 0.025$. The initial condition is taken as

$$u(x, y, z, 0) = \text{sign}(\delta + vf), \tag{6.6}$$

where $vf = -0.2$ and δ is randomly assigned between 0 and 1. The domain is $[-\pi/2, \pi/2] \times [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$ and the boundary conditions are (2.6). Numerical results of Scheme (4.1) for $\mathcal{P}^2$ approximation are shown in Fig. 12, which shows similar patterns as those in Example 6.4.

Figure 13 shows the energy trace of the numerical solution and we can see that the energy (2.2) of FCH equation decays with respect to time t.

Finally, what we should have in mind is that the DG spatial discretization does allow for more flexibility in several other ways. DG methods are a class of finite element methods, which can handle the irregular computational domain and complex boundary conditions easily. Meanwhile, since the basis functions can be completely discontinuous, discontinuous Galerkin methods have the flexibility which is not shared by typical finite element methods, such as the allowance of arbitrary triangulation with hanging nodes, complete freedom in changing the polynomial degrees in each element independent of that in the neighbors (p-adaptivity), and extremely local data structure (elements only communicate with immediate neighbors regardless of the order of accuracy of the scheme) and the resulting embarrassingly high parallel efficiency.

7 Concluding Remarks

In this paper, we have constructed a local discontinuous Galerkin method to solve the functionalized Cahn–Hilliard equation. The energy stability is proved for the general nonlinear case. A semi-implicit method is used for time discretization to remove the severe time step restriction for explicit methods. The use of the semi-implicit method will result in a linear

algebraic system at each time step and the multigrid solver is used to solve the system. In addition, we show numerically that the number of the multigrid iterations is independent of the problem size and is of optimal complexity. These results in two and three dimensions indicate that the local discontinuous Galerkin method and the multigrid method are good tools for solving such high order and nonlinear partial differential equations in mathematic physics.

In addition to the advantages of the local discontinuous Galerkin method and the multigrid solver, we can speed up the simulation by adapting the time step to the evolution of the solution. We numerically show the unconditional stability of the Scheme (4.1) with $s_1 = s_2 = 2$. This allows us to choose a time step as large as we would like at the cost of numerical error.

8 Appendix 1: The Linear MG Solver

We start by introducing the basic notations to describe the general setting of a two-level or bi-grid method. Together with the family of partitions $\{\mathcal{T}_h\}_{h>0}$ used for the LDG discretization, we consider a coarse family of mesh partitions, $\{\mathcal{T}_H\}_{H>0}$ with $H > h$ and satisfying the basic assumption $\mathcal{T}_H \subset \mathcal{T}_h$. One can think $H = 2h$, since in many circumstances it will already be coarse enough. Associated to the coarse mesh partition we have the corresponding finite element space V_H which is defined as

$$V_H = \{v \in L^2(D) : v|_D \in \mathcal{P}^k(D); \quad \forall D \in \mathcal{T}_H\}. \tag{8.1}$$

Throughout the whole description we assume the polynomial degree k is fixed.

To link functions in both spaces, we define the prolongation and restriction operators. The prolongation operator $P_{hH} : V_H \vee V_h$ is defined as the natural inclusion. The restriction operator $R_{Hh} : V_h \vee V_H$ is defined as the transpose of P_{hH} with respect to the standard L^2-inner product. That is, it is obtained by solving:

$$\sum_{D\in\mathcal{T}_H} \int_D R_{Hh}(u_h) v_H dx = \sum_{D\in\mathcal{T}_h} \int_D u_h P_{hH}(v_H) dx, \quad \forall v_H \in V_H. \tag{8.2}$$

We now denote S_h as a general relaxation or smoothing operator. We later specify and study several choices. The basic property that S_h should have is to damp the high frequencies of the approximate solution and smooth the error. The coarse solver is defined by $A_H = R_{Hh} A_h P_{hH}$. The coarse grid correction step is to reduce the smooth components of the error that can not be reduced by the smoother.

The linear scheme requires the solution at each time step of algebraic non-symmetric linear systems i.e.

$$A_h u_h = f_h. \tag{8.3}$$

The MG method is used to solve the system and the main points of the algorithm is the bi-grid cycle. Following [2], we can formulate the bi-grid cycle as follows:

Algorithm 1: Two-grid Cycle Starting with an initial approximation, say u^0_{PRE}:

1. *Pre-relaxation* apply ν_1 pre-relaxation sweeps: for $m = 1 \ldots, \nu_1$, solve

$$u^m_{PRE} = u^{m-1}_{PRE} + S_h(f_h - A_h u^{m-1}_{PRE}),$$

Table 4 The possible choices of the smoother operator

Smoother	Jacobi	Gauss–Seidel	Damped Jacobi	Damped Gauss–Seidel
S_h	D_h^{-1}	$(D_h + L_h)^{-1}$	αD_h^{-1}	$\alpha(D_h + L_h)^{-1}$

2. *Coarse-grid correction* update the solution $u_{PRE}^{\nu_1}$ by a coarse-grid correction step, solve the problem once on coarse grid

$$A_H v_H = R_{Hh}(f_h - A_h u_{PRE}^{\nu_1}),$$

and set $u_{CG} = u_{PRE}^{\nu_1} + P_{hH}(v_H)$.
3. *Post-relaxation* starting with u_{CG}, apply ν_2 post-relaxation sweeps, that is, set $u_{POST}^0 = u_{CG}$ and for $m = 1 \ldots, \nu_2$, solve

$$u_{POST}^m = u_{POST}^{m-1} + S_h(f_h - A_h u_{POST}^{m-1}).$$

The integers ν_1 and ν_2 are parameters in the scheme that control the number of relaxation sweeps before and after visiting the coarse grid. ν_1 and ν_2 are called the number of pre- and post- relaxations, respectively.

Solving the coarse grid problem at the second step of the above algorithm could be done again with the two-level algorithm. Hence, the V-cycle multi-level algorithm in terms of the two-level algorithm is defined by applying the two-level algorithm recursively.

Due to the block structure of A_h, we focus only on very simple block-relaxation. In particular, we decompose A_h into a strict block-lower, a block-diagonal, and a strict block-upper matrix, i.e.

$$A_h = L_h + D_h + U_h. \tag{8.4}$$

Table 4 shows some possible choices of the smoother operator.

9 Appendix 2: The Local Mode Analysis of the Two-Grid Algorithm

In [1], the author considered a general framework for performing the local mode analysis for analyzing the convergence of two-level or bi-grid algorithms, and also provided some quantitative information about the performance and design of the solvers. Although the approach is applied to constant coefficients, linear problems and uniform grids, some general results are established, based on the fact that some local mode analysis can be performed at the matrices. A different treatment has to be given to the part of the matrix associated with interior unknowns and that associated to boundary degrees of freedom. Ignoring the treatment of the boundaries, we now revise some of the results given in [1]. There, the author defined the convergence factor of the two-grid method by

$$\lambda = \sup \frac{\|u_{POST}^{\nu_2}\|}{\|u_{PRE}^0\|} \tag{9.1}$$

in some appropriate chosen norm that might depend on the problem. It is also shown, that under the assumptions described above (constant coefficients, linear problems, uniform grids and neglecting the boundary conditions), the convergence factor might be computed in terms of the symbol of the error propagation operator.

For the linear iteration, the error propagation operator is defined as:

$$E_h := I - S_h A_h, \tag{9.2}$$

where I is the identity operator in V_h^k. The spectral radius, or some norm of this operator allows to quantify how the error is reduced at each iteration. If it is less than 1 we will get a convergent iteration. The smaller it is, the faster is the iteration.

In the case of the two level or two-grid cycle defined in Algorithm 1, the error propagation is:

$$E_h^{2grid} := E_h^{\nu_2}[I - P_{hH} A_H^{-1} R_{Hh} A_h] E_h^{\nu_1}. \tag{9.3}$$

Following [1], if one could choose the L^2-norm, denotes by $\widehat{E}_h(\theta)$ the symbol (in the frequency space) of the error propagation operator, from Parseval's identity it is formally obtained

$$\lambda = \sup_{\theta \neq 0} \|\widehat{E}_h(\theta)\|. \tag{9.4}$$

While for symmetric problems, the estimation of the spectral radius of E_h could be reduced to the computation of its largest eigenvalue, in the present situation, since A_h is non-symmetric and also S_h, one can not guarantee that their spectral information contain the relevant information.

To compute the spectral radius, a possible way is to compute the first singular value of E_h. In particular, one can define the *asymptotic convergence factor* (see [1]) as:

$$\lambda_{asymp} = \sup_{\theta \neq 0} \sigma_1(\widehat{E}_h(\theta)), \tag{9.5}$$

where σ_1 is the spectral radius of E, (i.e. largest absolute eigenvalue). On the other hand, a more restrictive way of ensuring that $\|E_h^{2grid}\| < 1$ would be to study the norms of each of the terms in the product. From the definition, we see that an essential requirement is to guarantee that the smoother has norm strictly less than 1.

References

1. Brandt, A.: Rigorous quantitative analysis of multigrid. I. Constant coefficients two-level cycle with L_2-norm. SIAM J. Numer. Anal. **31**, 1695–730 (1994)
2. Brandt, A.: Multigrid techniques: 1984 guide with applications to fluid dynamics. GMD-Studien [GMD Studies], 85. Gesellschaft für Mathematik und Datenverarbeitung mbH, St. Augustin (1984)
3. Cahn, J.W., Hilliard, J.E.: Free energy of non-uniform system. I. Interfacial free energy. J. Chem. Phys. **28**, 258–267 (1958)
4. Chen, F., Shen, J.: Efficient spectral-Galerkin methods for systems of coupled second-order equations and their applications. J. Comput. Phys. **231**, 5016–5028 (2012)
5. Chen, L.-Q., Wang, Y.-Z.: The continuum field approach to modeling microstructural evolution. J. Miner. Metals Mater. Soc. **48**, 13–18 (1996)
6. Chen, L.-Q., Wolverton, C., Vaithyananthan, V., Liu, Z.-K.: Modeling solid-state phase transformations and microstructure evolution. MRS Bull. **26**, 197–202 (2001)
7. Cockburn, B., Hou, S., Shu, C.-W.: The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. Math. Comput. **54**, 545–581 (1990)
8. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. J. Comput. Phys. **84**, 90–113 (1989)
9. Cockburn, B., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**, 411–435 (1989)
10. Cockburn, B., Shu, C.-W.: The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)

11. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal. **35**, 2440–2463 (1998)
12. Du, Q., Liu, C., Wang, X.Q.: Retrieving topological information for phase field models. SIAM J. Appl. Math. **65**, 1913–1932 (2005)
13. Elliott, C.M., French, D.A.: A nonconforming finite-element method for the two-dimensional Cahn–Hilliard equation. SIAM J. Numer. Anal. **26**, 884–903 (1989)
14. Elliott, C.M., French, D.A., Milner, F.A.: A second order splitting method for the Cahn–Hilliard equation. Numer. Math. **54**, 575–590 (1989)
15. Elliott, C.M., Garcke, H.: On the Cahn–Hilliard equation with degenerate mobility. SIAM J. Math. Anal. **27**, 404–423 (1996)
16. Gavish, N., Hayrapetyan, G., Promislow, K., Yang, L.: Curvature driven flow of bilayer interfaces. Phys. D **240**, 675–693 (2011)
17. Gavish, N., Jones, J., Xu, Z., Christlieb, A., Promislow, K.: Variational models of network formation and ion transport: applications to perfluorosulfonate ionomer membranes. Polymers **4**, 630–655 (2012)
18. Gompper, G., Schick, M.: Correlation between structural and interfacial properties of amphiphilic systems. Phys. Rev. Lett. **65**, 1116–1119 (1990)
19. Guo, R., Xu, Y.: Efficient solvers of discontinuous Galerkin discretization for the Cahn–Hilliard equations. J. Sci. Comput. **58**, 380–408 (2014)
20. Jones, J., Christlieb, A., Promislow, K., Xu, Z.: A gradient stable scheme for simulating long time adiabatic evolution of a solvent functionalized-polymer system (Preprint)
21. Kay, D., Welford, R.: A multigrid finite element solver for the Cahn–Hilliard equation. J. Comput. Phys. **212**, 288–304 (2006)
22. Kim, J., Kang, K., Lowengrub, J.: Conservative multigrid methods for Cahn–Hilliard fluids. J. Comput. Phys. **193**, 511–543 (2004)
23. Kim, J., Kang, K., Lowengrub, J.: Conservative multigrid methods for ternary Cahn–Hilliard systems. Commun. Math. Sci. **2**, 53–77 (2004)
24. Promislow, K., Wetton, B.: PEM fuel cells: a mathematical overview. SIAM J. Appl. Math. **70**, 369–409 (2009)
25. Reed, W., Hill, T.: Triangular mesh methods for the neutrontransport equation, La-ur-73-479, Los Alamos Scientific Laboratory (1973)
26. Xia, Y., Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for the Cahn–Hilliard type equations. J. Comput. Phys. **227**, 472–491 (2007)
27. Xia, Y., Xu, Y., Shu, C.-W.: Application of the local discontinuous Galerkin method for the Allen–Cahn/Cahn–Hilliard system. Commun. Comput. Phys. **5**, 821–835 (2009)
28. Xu, Y., Shu, C.-W.: Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. Commun. Comput. Phys. **7**, 1–46 (2010)