Triangulation-based isogeometric analysis of the Cahn–Hilliard phase-field model

Ruochun Zhang, Xiaoping Qian *

Department of Mechanical Engineering,
University of Wisconsin-Madison,
Madison, WI 53706 - 1572

August 15, 2019

Abstract

This paper presents triangulation-based Isogeometric Analysis of the Cahn–Hilliard phase-field model. The Cahn–Hilliard phase-field model is governed by a time-dependent fourth-order partial differential equation. The corresponding primal variational form involves second-order operators, making it difficult to be directly analyzed with traditional C^0 finite element analysis. In this paper, we construct C^1 Bernstein-Bézier simplicial elements through macro-element techniques, including various triangle-split based macro-elements in both 2D and 3D space. We extend triangulation-based isogeometric analysis to solving the primal variational form of the Cahn–Hilliard equation. We validate our method by convergence analysis, showing the nodal and degree-of-freedom advantages over C^0 Finite Element Analysis. We then demonstrate detailed system evolution from randomly perturbed initial conditions in periodic two-dimensional squares and three-dimensional cubes. We incorporate an adaptive time-stepping scheme in these numerical experiments. Our numerical study demonstrates that triangulation-based isogeometric analysis offers optimal convergence and time step stability, is applicable to complex geometry and allows local refinement.

Keywords: Isogeometric analysis, triangulation, Bernstein-Bézier elements, Cahn-Hilliard equations, phase-field model, C^1 element

1 Introduction

Isogeometric Analysis (IGA) has become an established numerical method for solving Partial Differential Equations (PDEs) in which the same basis functions are used both to represent

*qian@engr.wisc.edu

the geometric models and to approximate the state fields[1]. Compared with the traditional Finite Element Analysis (FEA), IGA not only allows the exact representation of the geometry of computational domains, but also features C^1 and higher-order continuity, leading to several advantages in numerical analysis such as numerical accuracy on a per-node basis.

IGA brings new possibilities in solving high-order PDEs, with the governing equation of the Cahn–Hilliard phase-field model being one of them. The Cahn–Hilliard phase-field model can be derived from the free energy of an isotropic system of non-uniform composition or density[2]. This model replaces sharp interfaces by thin transition regions where the interfacial forces are smoothly distributed[3]. As such, by using the Cahn–Hilliard phase-field model the need for boundary tracking and mesh moving in certain physics problems is eliminated. Over the years, the Cahn–Hilliard phase-field model has seen its usage extended considerably, serving as a numerical approach in the simulation of two-phase flows, microstructures with elastic inhomogeneity, tumor growth etc., as well as in areas such as image inpainting and topology optimization[4].

Both FEA and IGA based numerical approaches have been developed for solving the Cahn–Hilliard equation. Since the primal variational form of the Cahn–Hilliard equation has second-order operators, it is not directly solvable with C^0 FEA. A variety of methods have been put forward for addressing this issue, notably the mixed method[5], the discontinuous Galerkin method[6] and the continuous/discontinuous Galerkin method[7]. In IGA however, the primal variational form can be used directly to obtain the solution of the phase-field model. For example, Non-Uniform Ration B-spline (NURBS) based IGA of the Cahn-Hillard equation is given in [3, 8]. Comparison of the advantages and disadvantages of various C^0 and C^1 spatial discretizations is given in [9, 10]. While being a powerful modeling tool, NURBS generally cannot achieve C^1-continuous parameterization of complex topology without incurring extraordinary nodes. Due to the tensor-product nature of NURBS, local refinement is also challenging. Alternative IGA techniques such as T-spline-based IGA have their own complications, including degenerated continuity[11, 12].

Recently triangulation-based Isogeometric Analysis (tIGA) has emerged as a powerful tool for both shape modeling and numerical analysis. Most noticeably in [13], a generalized framework of tIGA is introduced. A globally C^r-continuous basis for representing geometry with exact recovery of its NURBS boundary can be constructed in different spaces, including polynomial macro-element spaces, Powell–Sabin (PS) macro-element spaces[14] and Clough–Tocher (CT) macro-element spaces[15]. In addition, [16] shows optimal convergence rates can be achieved in tIGA with several Poisson and linear elasticity problems as examples. Then a three-dimensional extension with Bézier tetrahedra is demonstrted in [17]. [18] further introduces tIGA shape optimization, and [19] adds another application by developing Kirchhoff–Love shell elements in the context of tIGA. In these works, apart from the flexibility in representing domains of complex topology and higher order of continuity, tIGA also allows local refinement, thus overcoming the limitations of NURBS-based IGA mentioned above.

In this paper, we apply C^1 tIGA to solve the Cahn–Hilliard equation. We use Bernstein-Bézier triangles or tetrahedra to construct the computational domain for solving the primal

variational Cahn–Hilliard equation. This triangular mesh, consisting of various quadratic, cubic and quintic elements, has global C^1 continuity and the construction scheme has been automated. We test the tIGA on a model problem with manufactured solution and examine its convergence rate. A comparison with Lagrange polynomials based FEA in terms of convergence rates and numerical efficiency is made. We then extend this IGA method to several Cahn-Hillard modelled problems in both 2D and 3D. With numerical techniques such as generalized-α method and adaptive time stepping method, the phase separation process in two and three dimensions governed by the Cahn–Hilliard equation is simulated.

The remainder of the paper is organized as follows. The basics on B-splines and Bézier triangles are briefly described in Section 2. In Section 3, we outline the Cahn–Hilliard phase-field model and relevant numerical methodologies. The construction of C^1-continuous meshes and the numerical implementation are detailed in Section 4. The manufactured solution based convergence analysis is conducted in Section 5. In Section 6 we present a variety of physically meaningful and numerically representative test examples. Finally, conclusions are drawn in Section 7.

2 B-splines and Bézier triangles

In this work, we use Bézier triangles and tetrahedra to parametrize the computational domain and conduct isogeometric analysis. For completeness of the paper, a brief introduction of B-splines, Bézier triangles and tetrahedra is given in this section. More details on Bézier triangulations can be found in [20, 21, 22]. and its usage in isogeometric analysis in [16, 17].

2.1 B-splines

A B-spline curve of degree d and $n+1$ control points is defined as

$$\mathbf{S}(u) = \sum_{i=0}^{n} \mathbf{c}_i R_{i,d}(u), \tag{1}$$

where $\mathbf{c}_i$ is the i-th control point and $R_{i,d}$ is the i-th B-spline basis function for a given parameter u.

2.2 Bézier triangles

A single span of a B-spline curve is a Bézier curve which can be defined by Bernstein polynomials. The $d+1$ Bernstein polynomials of degree d read

$$B_{\mathbf{i},d}(\xi) = \frac{d!}{i!j!}\xi^i(1-\xi)^j, \quad |\,\mathbf{i}\,| = i+j = d. \tag{2}$$

Accordingly, a Bézier patch can also be defined by bivariate Bernstein polynomials. The d-th degree bivariate Bernstein polynomial can be defined as

$$B_{\mathbf{i},d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!}\gamma_1^i\gamma_2^j\gamma_3^k, \quad |\ \mathbf{i}\ | = i+j+k = d, \tag{3}$$

where $\mathbf{i}$ represents a triple index (i,j,k) and $(\gamma_1,\gamma_2,\gamma_3)$ is the barycentric coordinate of a point $\boldsymbol{\xi}$ in a domain triangle $\tau = \{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3\}$. Then any point $\boldsymbol{\xi} = (\xi_1,\xi_2) \in \mathbb{R}^2$ in τ can be written in the form

$$\boldsymbol{\xi} = \gamma_1\mathbf{v}_1 + \gamma_2\mathbf{v}_2 + \gamma_3\mathbf{v}_3, \quad \gamma_1+\gamma_2+\gamma_3 = 1. \tag{4}$$

A triangular Bézier patch is defined as

$$\mathbf{b}(\boldsymbol{\xi}) = \sum_{|\mathbf{i}|=d} \mathbf{p}_\mathbf{i} B_{\mathbf{i},d}(\boldsymbol{\xi}), \tag{5}$$

where $\mathbf{p}$ represent control points of the patch.

A rational Bézier triangle can be similarly defined as

$$\begin{aligned} \mathbf{b}(\boldsymbol{\xi}) &= \sum_{|\mathbf{i}|=d} \mathbf{p}_\mathbf{i}\phi_{\mathbf{i},d}(\boldsymbol{\xi}), \\ \phi_{\mathbf{i},d} &= \frac{w_\mathbf{i} B_{\mathbf{i},d}}{\sum_{|\mathbf{i}|=d} w_\mathbf{i} B_{\mathbf{i},d}} = \frac{w_\mathbf{i} B_{\mathbf{i},d}}{w}, \end{aligned} \tag{6}$$

where $w_\mathbf{i}$ is the weight of control point $\mathbf{p}_\mathbf{i}$.

With bivariate Bernstein polynomials, we can also define a polynomial function $f(\boldsymbol{\xi})$ of degree d on a triangle τ with vertices $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3$ as

$$f(\boldsymbol{\xi}) = \sum_{|\mathbf{i}|=d} b_\mathbf{i}\phi_{\mathbf{i},d}(\boldsymbol{\xi}), \tag{7}$$

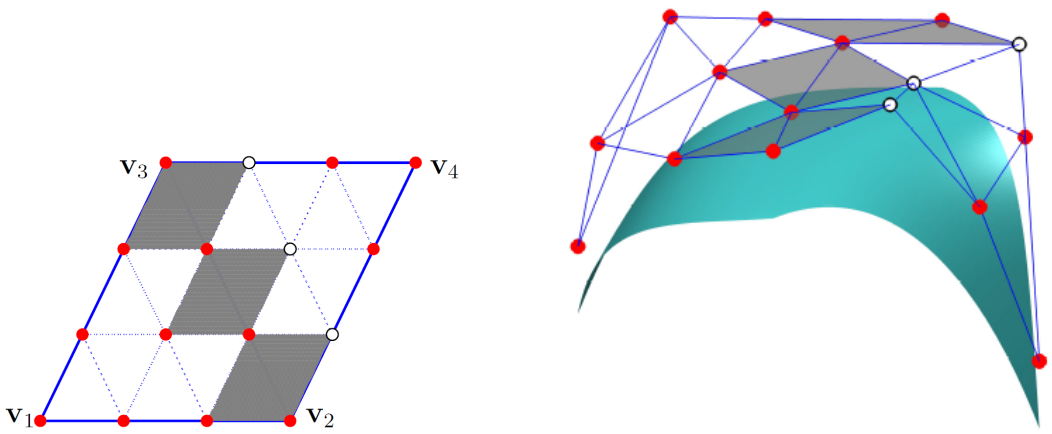
where $b_\mathbf{i}$ (or b_{ijk}) are Bézier ordinates of $f(\boldsymbol{\xi})$. The set of domain points associated with them is given as

$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijk} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3}{d},\ i+j+k = d \right\}. \tag{8}$$

For two triangles $\tau = \{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3\}$ and $\tilde{\tau} = \{\mathbf{v}_4,\mathbf{v}_3,\mathbf{v}_2\}$ who share a common edge, two polynomials f and $\tilde{f}$ of degree d join r times differentiably across this edge if and only if[20]

$$\tilde{b}_{\rho,j,k} = \sum_{\mu+\nu+\kappa=\rho} \frac{\rho!}{\mu!\nu!\kappa!} b_{\mu,k+\nu,j+\kappa}\gamma_1^\mu\gamma_2^\nu\gamma_3^\kappa, \quad j+k+\rho = d, \quad \rho = 0,...,r, \tag{9}$$

where γ_1, γ_2, γ_3 are the barycentric coordinates of vertex $\mathbf{v}_4$ with respect to triangle τ. This continuity relation suggests that, as illustrated in Fig. 1, when C^r continuity is enforced, the values associated with the white nodes are determined by the values associated with the red nodes through this constraint. The shaded areas in Fig. 1 indicate the triangles with shared edges where the continuity constraint is imposed for C^1 smoothness. Construction



(a) Two domain triangles with C^1 constraints on Bézier ordinates.

(b) Two Bézier patches with C^1 continuity.

Figure 1: Triangular Bézier patches with C^1 smoothness. The dependent nodes (white solids) are determined by the free nodes (red solids) through the continuity constraints. The shaded areas indicate the triangles with shared edges where the constraints are imposed. As can be seen in (b), the control points in each shaded triangle pair are coplanar. For better visualization, the control net is shifted up slightly in (b).[1]

of such dependency plays an important role in tIGA. In this work we use macro-element technique[13] to generate C^1-continuous mesh for solving the phase-field problem.

2.3 Bézier tetrahedra

As explained in [17], by extending the bivariate case we have the trivariate d-th degree Bernstein polynomial

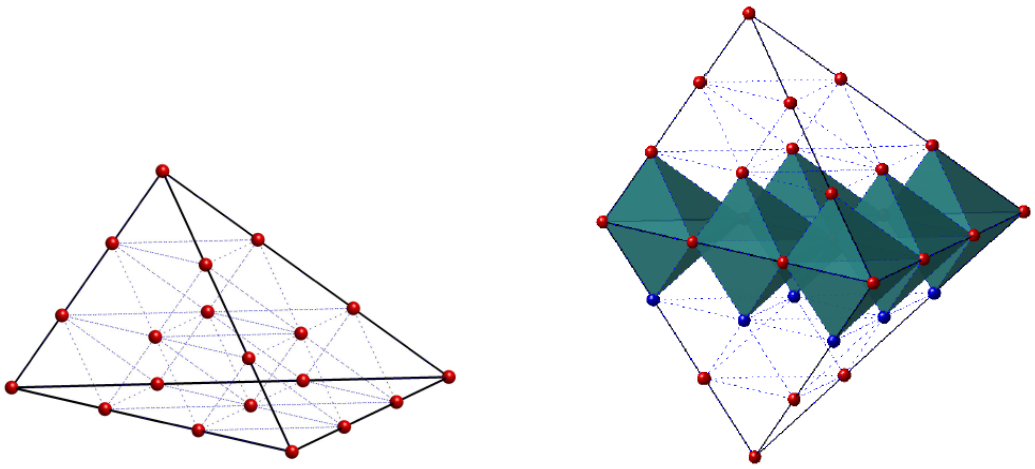
$$B_{\mathbf{i},d}(\boldsymbol{\xi}) = \frac{d!}{i!j!k!l!}\gamma_1^i\gamma_2^j\gamma_3^k\gamma_4^l, \quad |\,\mathbf{i}\,| = i+j+k+l = d, \tag{10}$$

where $\mathbf{i}$ represents a triple index (i,j,k,l) and $(\gamma_1,\gamma_2,\gamma_3,\gamma_4)$ is the barycentric coordinate of a point $\boldsymbol{\xi}$ in a triangle $\tau = \{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3,\mathbf{v}_4\}$. Then any point $\boldsymbol{\xi} = (\xi_1,\xi_2,\xi_3) \in \mathbb{R}^3$ in τ can be written in the form

$$\boldsymbol{\xi} = \gamma_1\mathbf{v}_1 + \gamma_2\mathbf{v}_2 + \gamma_3\mathbf{v}_3 + \gamma_4\mathbf{v}_4, \quad \gamma_1+\gamma_2+\gamma_3+\gamma_4 = 1. \tag{11}$$

A rational Bézier tetrahedron and its basis are defined similarly as in Eq. 6. The

[1]Reprinted from *Computer Methods in Applied Mechanics and Engineering*, Vol 297, Songtao Xia, Xilu Wang, and Xiaoping Qian, Continuity and convergence in rational triangular Bézier spline based isogeometric analysis, Pages 292–324, Copyright 2015, with permission from Elsevier.



(a) Domain points of a Bézier tetrahedron.

(b) Two domain tetrahedra with C^1 constraints on Bézier ordinates. The Bézier ordinates corresponding to the dependent nodes (blue) are determined by the free nodes (red) through the continuity constraints.

Figure 2: Domain points and continuity constraints of Bézier tetrahedra.[2]

associated domain points of a tetrahedron τ are

$$\mathcal{D}_{d,\tau} = \left\{ \mathbf{q}_{ijkl} = \frac{i\mathbf{v}_1 + j\mathbf{v}_2 + k\mathbf{v}_3 + l\mathbf{v}_4}{d},\ i+j+k+l=d \right\}. \tag{12}$$

For two tetrahedra $\tau = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ and $\tilde{\tau} = \{\mathbf{v}_5, \mathbf{v}_2, \mathbf{v}_4, \mathbf{v}_3\}$ who share a common face, two polynomials f and $\tilde{f}$ of degree d join r times differentiably across this face if and only if[20]

$$\tilde{b}_{\rho,i,j,k} - \sum_{\mu+\nu+\kappa+\delta=\rho} \frac{\rho!}{\mu!\nu!\kappa!\delta!} b_{\mu,i+\nu,j+\kappa,k+\delta} \gamma_1^\mu \gamma_2^\nu \gamma_3^\kappa \gamma_4^\delta = 0,$$
$$i+j+k+\rho = d, \rho = 0, ..., r,$$

where γ_1, γ_2, γ_3, γ_4 are the barycentric coordinates of vertex $\mathbf{v}_5$ with respect to tetrahedron τ. As presented in Fig. 2, the values associated with the blue nodes are determined by the values associated with the red nodes through this constraint if one is to impose C^r constraint. In our three-dimensional numerical examples such smooth Bézier tetrahedron is used.

[2]Reprinted from *Computer Methods in Applied Mechanics and Engineering*, Vol 316, Songtao Xia and Xiaoping Qian, Isogeometric analysis with Bézier tetrahedra, Pages 782–816, Copyright 2017, with permission from Elsevier.

3 The Cahn–Hilliard phase-field model

This section describes the phase-field model[2] and our Galerkin discretization.

3.1 The Cahn–Hilliard equation

We follow the phase-field numerical model proposed in [3]. The governing equations of our phase-field model is the Cahn–Hilliard equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (M_c \nabla (f_c' - \lambda \Delta c)) \quad \text{in} \quad \Omega, \tag{13.1}$$

$$c = g \quad \text{on} \quad \Gamma_g, \tag{13.2}$$

$$M_c \nabla (f_c' - \lambda \Delta c) \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma_h, \tag{13.3}$$

$$M_c \lambda \nabla c \cdot \mathbf{n} = 0 \quad \text{on} \quad \Gamma, \tag{13.4}$$

$$c(\mathbf{x}, 0) = c_0(\mathbf{x}) \quad \text{in} \quad \Omega, \tag{13.5}$$

where c is the unknown chemical concentration, M_c is the mobility, f_c' is related to the derivative of chemical free energy Ψ^c, λ is related to the transient layer thickness and the derivative of surface free energy Ψ^s given in Eq. 15, $\mathbf{n}$ is the direction normal to the boundaries and $\Gamma = \overline{\Gamma_g \cup \Gamma_h}$.

According to the original derivation of the Cahn–Hilliard equation[2, 23], we model the chemical free energy Ψ^c and the surface free energy Ψ^s as

$$\Psi^c = NkT(clnc + (1-c)ln(1-c)) + N\omega c(1-c), \tag{14}$$

$$\Psi^s = N\omega \frac{1}{2} \lambda \|\nabla c\|^2, \tag{15}$$

where N is the number of molecules per unit volume, k is the Boltzmann's constant, ω is an interaction energy, and

$$\omega = 2kT_c, \tag{16}$$

with T_c being the critical temperature at which the two phases attain the same composition. If $T_c/T > 1$, the chemical free energy will drive phase separation as Ψ^c in this case has two local minimums. As such, f_c' is given by

$$f_c' = \Psi^{c'}/(N\omega) = \frac{T}{2T_c} ln \frac{c}{1-c} - 2c + 1. \tag{17}$$

For a better understanding of the system evolution, we model M_c using the degenerate mobility $Dc(1-c)$[5], where D is a constant.

In order to carry out the computation in a dimensionless space the following substitution

is enforced:

$$\begin{aligned} \mathbf{x}^* &= \mathbf{x}/L_0, \\ \nabla^* &= \frac{\partial}{\partial \mathbf{x}^*} = L_0 \cdot \nabla, \\ t^* &= t/T_0, \end{aligned} \tag{18}$$

where L_0 and T_0 are the characteristic length scale and time scale respectively. Plug them into Eq. 13.1 and we have

$$\frac{1}{T_0} \cdot \frac{\partial c}{\partial t^*} = \frac{\lambda}{L_0^4} \cdot \nabla^* \cdot (Dc(1-c)\nabla^*(f_c' \cdot \frac{L_0^2}{\lambda} - \Delta^* c)). \tag{19}$$

Then, we choose $T_0 = L_0^4/(D\lambda)$ to integrate the mobility constant D into T_0, and we omit superscript that denotes a dimensionless space, then Eq. 13.1 becomes dimensionless:

$$\dot{c} = \nabla \cdot (m_c \nabla(p_c - \Delta c)), \tag{20}$$

where $\dot{c} = \partial c/\partial t$, $p_c = f_c' \cdot L_0^2/\lambda$ and it is referred as the dimensionless chemical potential, $m_c = c(1-c)$ and it is referred as the dimensionless mobility. We remark that all reported results in the dimensionless space can be scaled back to the physical space through L_0 and T_0.

In our numerical experiments, we assume $T_c/T = 1.5$[7], so the concentration equilibrium with minimal system energy is achieved at $c = 0.07$ and $c = 0.93$ (refer to Eq. 17). L_0^2/λ becomes the only parameter that characterize our solution other than boundary conditions. We refer L_0^2/λ as α. Higher α results in a thinner transient layer relative to the length scale of the computational domain, making numerical experiments more challenging. This effect is further explained and examined in Section 6.

3.2 Weak form and Galerkin approximation

Assume periodic boundary condition on the domain boundary, and let $\mathcal{V}$ be both the weighting function space and the trial function space, the problem then becomes to find $c \in \mathcal{V}$ s.t. $\forall w \in \mathcal{V}$, the following primal variational form of the Cahn–Hilliard equation

$$\begin{aligned} &G(w,c) = \int_\Omega w \cdot \dot{c} d\Omega + \int_\Omega \nabla w \cdot (m_c \nabla p_c + \nabla m_c \Delta c) d\Omega + \int_\Omega \Delta w \cdot (m_c \Delta c) d\Omega, \\ &G(w,c) = 0, \end{aligned} \tag{21}$$

is satisfied.

In order to numerically solve this primal variational form of Cahn–Hilliard equation in a finite discretized space we use the Galerkin's approximation. Let $\mathcal{V}^h$ be a finite dimensional subset of $\mathcal{V}$, then the problem becomes to find $c^h \in \mathcal{V}^h$ such that $\forall w^h \in \mathcal{V}^h$, we have the

following:

$$
\begin{aligned}
G(w^h, c^h) &= \int_\Omega w^h \cdot \dot{c}^h d\Omega + \int_\Omega \nabla w^h \cdot (m_c^h \nabla p_c^h) d\Omega \\
&\quad + \int_\Omega \nabla w^h \cdot (\nabla m_c^h \Delta c^h) d\Omega + \int_\Omega \Delta w^h \cdot (m_c^h \Delta c^h) d\Omega, \\
G(w^h, c^h) &= 0, \\
c^h &= \sum_i^{n_{dof}} c_i \psi_i, \\
w^h &= \sum_i^{n_{dof}} w_i \psi_i,
\end{aligned}
\tag{22}
$$

where n_{dof} is the dimension of the discretized space and ψ_i are the discretized basis functions.

The presence of second-order operators in Eq. 22 necessitates C^1-smooth discretization. We will describe how we construct such C^1 smooth discretization in Section 4.

3.3 Temporal discretization and time stepping scheme

For the temporal discretization of this time-dependent problem we employ the generalized-α method[24] due to its high-order accuracy in time. For the purpose of addressing the temporal multi-scale nature of the problem we borrow the adaptive time stepping scheme from [3], which shares the idea with the embedded Runge–Kutta methods[25], and is shown to be stable and convenient. For completeness, we briefly introduce those two methods. The content of this section is from said literature.

Generalized-α method Let $\mathbf{Y}$ denote the nodal solution while $\mathbf{G}$ denote the nodal values of the non-linear residual, at each time step, we solve the following system

$$
\begin{aligned}
&\mathbf{G}(\dot{\mathbf{Y}}_{n+\alpha_m}, \mathbf{Y}_{n+\alpha_f}) = \mathbf{M}_{n+\alpha_m} \dot{\mathbf{Y}}_{n+\alpha_m} - \mathbf{N}(\mathbf{Y}_{n+\alpha_f}) = 0, \\
&\mathbf{Y}_{n+1} = \mathbf{Y}_n + \Delta_t \dot{\mathbf{Y}}_n + \gamma \Delta_t (\dot{\mathbf{Y}}_{n+1} - \dot{\mathbf{Y}}_n), \\
&\dot{\mathbf{Y}}_{n+\alpha_m} = \dot{\mathbf{Y}}_n + \alpha_m (\dot{\mathbf{Y}}_{n+1} - \dot{\mathbf{Y}}_n), \\
&\mathbf{Y}_{n+\alpha_f} = \mathbf{Y}_n + \alpha_f (\mathbf{Y}_{n+1} - \mathbf{Y}_n),
\end{aligned}
\tag{23}
$$

where Δ_t is the time step size.

To begin the non-linear iteration at time step $n+1$, as the preparation we first set the predictor

$$
\begin{aligned}
\mathbf{Y}_{n+1}^{(0)} &= \mathbf{Y}_n, \\
\dot{\mathbf{Y}}_{n+1}^{(0)} &= \frac{\gamma - 1}{\gamma} \dot{\mathbf{Y}}_n.
\end{aligned}
\tag{24}
$$

Then at i-th non-linear iteration we evaluate iterates at α-levels

$$\begin{aligned}\mathbf{Y}^{(i)}_{n+\alpha_f} &= \mathbf{Y}_n + \alpha_f(\mathbf{Y}^{(i-1)}_{n+1} - \mathbf{Y}_n),\\ \dot{\mathbf{Y}}^{(i)}_{n+\alpha_m} &= \dot{\mathbf{Y}}_n + \alpha_m(\dot{\mathbf{Y}}^{(i-1)}_{n+1} - \dot{\mathbf{Y}}_n).\end{aligned} \tag{25}$$

Then solve the following linear system for $\Delta\dot{\mathbf{Y}}^{(i)}_{n+1}$

$$\mathbf{K}^{(i)}\Delta\dot{\mathbf{Y}}^{(i)}_{n+1} = -\mathbf{G}^{(i)}(\dot{\mathbf{Y}}^{(i)}_{n+\alpha_m}, \mathbf{Y}^{(i)}_{n+\alpha_f}), \tag{26}$$

where tangent matrix $\mathbf{K}^{(i)}$ is given by

$$\mathbf{K}^{(i)} = \alpha_m \frac{\partial \mathbf{G}^{(i)}(\dot{\mathbf{Y}}^{(i)}_{n+\alpha_m}, \mathbf{Y}^{(i)}_{n+\alpha_f})}{\partial \dot{\mathbf{Y}}_{n+\alpha_m}} + \alpha_f\gamma\Delta_t \frac{\partial \mathbf{G}^{(i)}(\dot{\mathbf{Y}}^{(i)}_{n+\alpha_m}, \mathbf{Y}^{(i)}_{n+\alpha_f})}{\partial \mathbf{Y}_{n+\alpha_f}}. \tag{27}$$

Lastly, we update iterates as

$$\begin{aligned}\mathbf{Y}^{(i)}_{n+1} &= \mathbf{Y}^{(i-1)}_{n+1} + \gamma\Delta_t\Delta\dot{\mathbf{Y}}^{(i)}_{n+1},\\ \dot{\mathbf{Y}}^{(i)}_{n+1} &= \dot{\mathbf{Y}}^{(i-1)}_{n+1} + \Delta\dot{\mathbf{Y}}^{(i)}_{n+1}.\end{aligned} \tag{28}$$

This finishes the non-linear iteration in one time step.

We follow the examples in [3] by setting parameters $\alpha_m = 5/6$, $\alpha_f = 2/3$, $\gamma = 2/3$, and terminating the non-linear iteration when $\|\mathbf{G}^{(i)}\|_2 < 10^{-4} \cdot \|\mathbf{G}^{(0)}\|_2$. In this work, the linear system in Eq. 26 is solved by MATLAB built-in direct linear solver through matrix factorization.

Adaptive time stepping scheme We need to select proper strategies to address the temporal multi-scale nature of the problem. One of the strategies is to change the time step size by comparing the solution of the aforementioned generalized-α method against that of the backward-Euler method[3], then adjust the step size for each time step based on its predecessor by

$$\Delta_t^{new} = \Delta_t^{old} \cdot \rho(\frac{tol}{err})^{\frac{1}{2}}, \tag{29}$$

where $err = \|\mathbf{Y}_\alpha - \mathbf{Y}_{BE}\|_\infty / \|\mathbf{Y}_\alpha\|_\infty$, meaning the difference between the generalized-α solution and the backward-Euler solution. We also follow [3] by setting $\rho = 0.9$ and $tol = 10^{-3}$. If $err > tol$, this time step will be recomputed using Δ_t^{new}; if $err \leq tol$, the solver will move on to compute next time step with a time step size of Δ_t^{new}.

Another strategy is to control the product of the transient term and the time step size sufficiently small. So in this case, we can effectively plug $err = \Delta_t^{old}\|\dot{\mathbf{Y}}_\alpha\|_\infty$ in Eq. 29. We discover that, by setting $\rho = 0.9$ and $tol = 0.002$, this time stepping strategy delivers similar numerical behavior compared to the former one, but does not require the backward-Euler computation which makes the time step control scheme almost free in computational cost. In our numerical experiments, we consider those two strategies interchangeable.

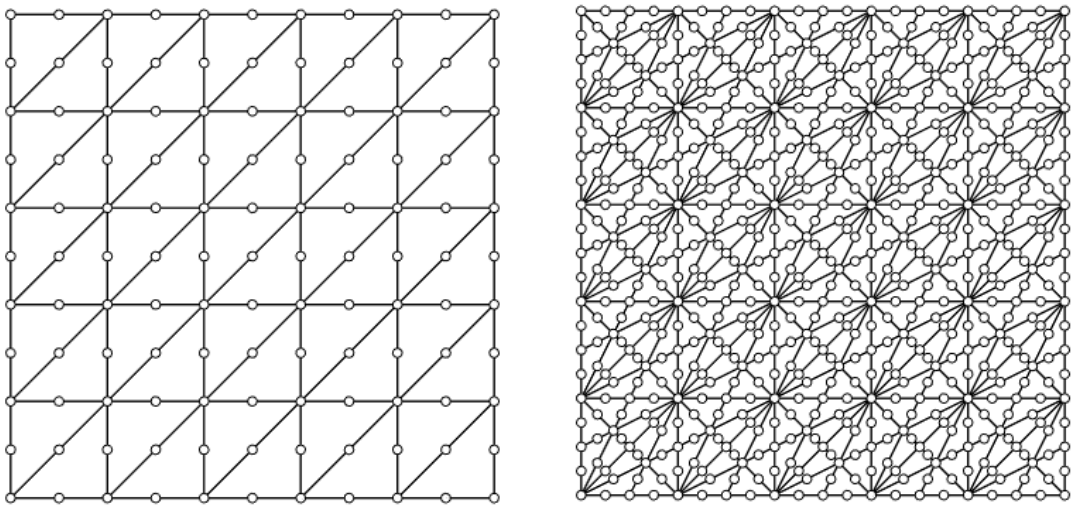
4 Constructing C^1 triangular elements and asembling Galerkin matrices

In this section, we further detail how we construct C^1 simplicial meshes and how we use the resulting smooth basis to assemble Galerkin matrices.

4.1 Macro-element based construction of C^1 simplicial mesh

For simplicity we use a Bézier triangulated uniform grid in a two-dimensional unit square domain as an example to describe the construction of C^1-continuous Bézier triangular meshes. The process is similarly followed in [16]. Our approach is to start with a C^0-continuous Bézier mesh, then enforce C^1 continuity. As pointed out in [26], the dimension of bivariate spline spaces of triangulation have dependency on the underlying geometry. To remove this dependency, we construct our C^1-continuous basis in the macro-element space[20, 13]. We now outline our procedure step by step.

Step 1, construct C^0 mesh Fig. 3(a) shows a simple C^0 quadratic mesh which can be comprehended as a triangulated uniform grid in a unit square domain.



(a) C^0 quadratic mesh. (b) C^0 quadratic mesh after PS split.

Figure 3: C^0 Bézier triangular mesh.

Step 2, Powell–Sabin split The mesh in Fig. 3(a) has only C^0 continuity. Let Ω denote a parametric domain, T denote its triangulation and $\mathcal{S}_d(T)$ denote the spline space of piecewise polynomials of degree d over T and furthermore $\mathcal{S}_d^r(T)$ denote that such space also attains continuity order r over Ω, we aim to construct a $\mathcal{S}_d^1$.

If we intend to apply Eq. 9 directly a substantial p-refinement is required. Specifically, it requires a refinement that ensures $d \geq 3r + 2$[27]. An alternative is to split each triangle in T into several micro-triangles before imposing the continuity constraints on the micro-triangles. The PS split[14] reduces the requirement to $d \geq \frac{9r-1}{4}$ for odd r and $d \geq \frac{9r+4}{4}$ for even r. PS split divides each triangle into six smaller triangles with the centroid point as the interior split point. Edges are then split by joining the centroid points of adjacent triangles. Imposing such split yields the mesh in Fig. 3(b), original elements are now called macro-elements, as opposed to micro-elements.

Step 3, construct C^1 mesh Let $\mathcal{D}_{d,T}$ denote the set of all domain points of T, $b_{\mathbf{v}}$ denote the Bézier ordinate of a domain point $\mathbf{v} \in \mathcal{D}_{d,T}$. A piecewise polynomial function $f(\boldsymbol{\xi}) \in \mathcal{S}_d$, $\boldsymbol{\xi} \in \mathbb{R}^2$ can be written in a discretized form with rational C^0 Bernstein basis ϕ and corresponding nodal ordinates $b_{\mathbf{v}}$

$$f(\boldsymbol{\xi}) = \sum_{i}^{n_{dof}} b_i \phi_i(\boldsymbol{\xi}) = \mathbf{b}_{\mathcal{D}_{d,T}}^T \boldsymbol{\phi}(\boldsymbol{\xi}). \tag{30}$$

If instead $f \in \mathcal{S}_d^r$, $r \geq 1$, Eq. 30 no longer holds true because we cannot assign arbitrary values to all coefficients of f but only to those correspond to a determining subset of domain points, and the remaining coefficients will be determined by them via Eq. 9; when this subset is the smallest among all possible determining sets we call it a minimal determining set (MDS) $\mathcal{M}_{d,T}$.

The continuity equation Eq. 9 suggests a linear system

$$\mathbf{A}\mathbf{b}_{\mathcal{D}_{d,T}} = \mathbf{0}, \tag{31}$$

where $\mathbf{A}$ is a coefficient matrix governed by the domain connectivity and $\mathbf{b}_{\mathcal{D}_{d,T}}$ are Bézier ordinates for all domain points in $\mathcal{D}_{d,T}$. The i-th row of $\mathbf{A}$ corresponds to the coefficients in the Bézier ordinates of the i-th constraint equation presented in Eq. 9. The j-th column in $\mathbf{A}$ corresponds to the Bézier ordinate b_j. Let $m = dim(\mathcal{S}_d^r(T))$ and $n = dim(\mathcal{S}_d^0(T))$, we have $rank(\mathbf{A}) = n - m$.

Apply Gauss–Jordan elimination to the linear system Eq. 31 to match each Bézier ordinates with their governing constraints we can zero out the corresponding items below and above the diagonal, resulting in

$$\begin{bmatrix} \mathbf{I}_{(n-m)\times(n-m)} & \mathbf{R}_{(n-m)\times m} \\ \mathbf{0}_{m\times(n-m)} & \mathbf{0}_{m\times m} \end{bmatrix} \begin{bmatrix} \mathbf{b}^d_{(n-m)\times 1} \\ \mathbf{b}^f_{m\times 1} \end{bmatrix} = \mathbf{A}\mathbf{b}_{\mathcal{D}_{d,T}} = \mathbf{0}, \tag{32}$$

where $\mathbf{b}^d$ and $\mathbf{b}^f$ are the set of dependent and free ordinates, respectively. The m free Bézier ordinates $\mathbf{b}^f$ correspond to the domain points in $\mathcal{M}_{d,T}$, we denote these ordinates as $\mathbf{b}_{\mathcal{M}_{d,T}}$.

It then becomes clear that

$$\mathbf{b}_{\mathcal{D}_{d,T}} = \mathbf{C}^T \mathbf{b}_{\mathcal{M}_{d,T}}, \tag{33}$$

where $\mathbf{C} = \begin{bmatrix} -\mathbf{R}_{(n-m)\times m} & \mathbf{I}_{(n-m)\times(n-m)} \end{bmatrix}$ is referred as the continuity matrix. That is, the

Bézier ordinates in $\mathcal{D}_{d,T}$ can be transformed from the Bézier ordinates in $\mathcal{M}_{d,T}$ with the continuity matrix $\mathbf{C}$.

Combining Eq. 30 and Eq. 33 yields the following observation:

$$f(\boldsymbol{\xi}) = \mathbf{b}^T_{\mathcal{D}_{d,T}} \boldsymbol{\phi}(\boldsymbol{\xi}) = \mathbf{b}^T_{\mathcal{M}_{d,T}} \mathbf{C} \boldsymbol{\phi}(\boldsymbol{\xi}) = \mathbf{b}^T_{\mathcal{M}_{d,T}} \boldsymbol{\psi}(\boldsymbol{\xi}), \tag{34}$$

where $\boldsymbol{\psi}(\boldsymbol{\xi}) = \mathbf{C}\boldsymbol{\phi}(\boldsymbol{\xi})$. This suggests that a set of C^r-continuous basis functions $\boldsymbol{\psi}(\boldsymbol{\xi})$ has been constructed as a linear combination of C^0 Bernstein basis $\boldsymbol{\phi}(\boldsymbol{\xi})$ since $\mathbf{C}$ implies the continuity constraints.

The matrix manipulation above serves as a mathematical explanation of MDS construction and is also a viable way to generate an MDS, but in this work we choose to use the direct construction method outlined in [20]. In this method, one can directly choose a set of free domain points based on which all other domain points are determined through continuity constraints, as demonstrated in Fig. 4. It has been shown in [20] that this method is capable of generating a plethora of polynomial spaces of different orders of continuity.

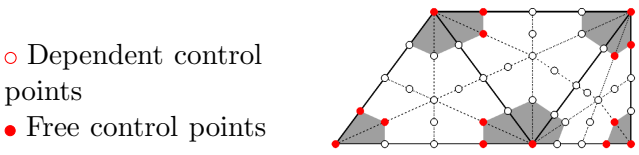


Figure 4: Using PS-split macro-element technique to impose continuity constraints for generating C^1 Bézier triangular meshes. Shaded areas indicate where the continuity constraint is imposed.[3]

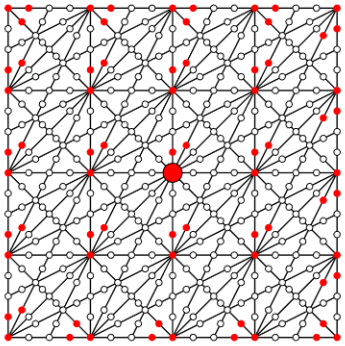
With $\mathcal{M}_{d,T}$ and corresponding $\boldsymbol{\psi}(\boldsymbol{\xi})$ we have the C^1-continuous mesh we need as shown in Fig. 5. For the three-dimensional tetrahedra we impose Alfeld split before imposing corresponding continuity constraint[28], where a tetrahedron is subdivided into four subtetrahedra by connecting its barycenter to each of its vertices.

4.2 Numerical implementation of Galerkin matrices

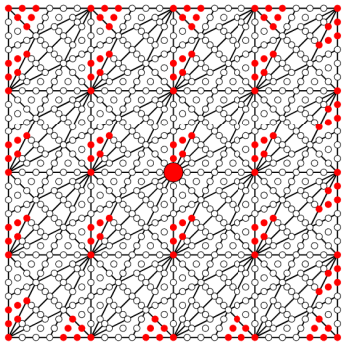
In order to numerically solve the primal variational form in Eq. 22, the discretized basis functions ψ_i must have C^1 continuity. We have such basis shown in Eq. 34. The residual G in Eq. 22 is a scalar function but according to the spatial discretization, we can write down the residual $\mathbf{G}$ (as in Eq. 26) in a matrix form explicitly:

$$\begin{aligned} \mathbf{G}_1 = &\int \boldsymbol{\psi} \cdot \dot{c}^h d\Omega + \int \nabla \boldsymbol{\psi} \cdot \frac{\alpha}{3} (\nabla c^h - 6c^h(1-c^h)\nabla c^h) d\Omega \\ &+ \int \nabla \boldsymbol{\psi} \cdot (1-2c^h)\Delta c^h \nabla c^h d\Omega + \int \Delta \boldsymbol{\psi} \cdot c^h(1-c^h)\Delta c^h d\Omega. \end{aligned} \tag{35}$$

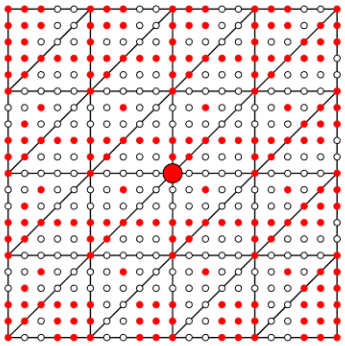
[3]Reprinted from *Computer Methods in Applied Mechanics and Engineering*, Vol 297, Songtao Xia, Xilu Wang, and Xiaoping Qian, Continuity and convergence in rational triangular Bézier spline based isogeometric analysis, Pages 292–324, Copyright 2015, with permission from Elsevier.



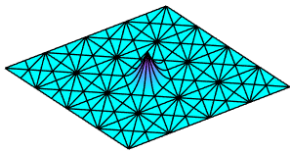
(a) C^1 quadratic mesh with PS split.



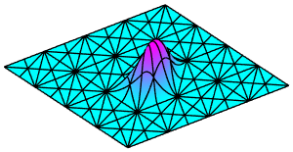
(b) C^1 cubic mesh with PS split.



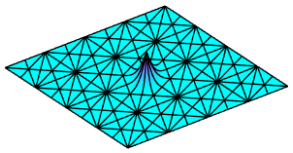
(c) C^1 quintic mesh with no split.



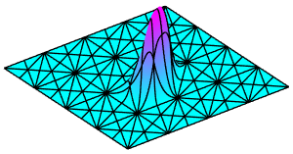
(d) C^0 quadratic basis function.



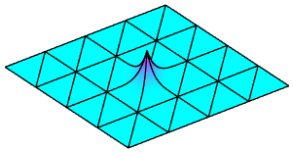
(e) C^1 quadratic basis function.



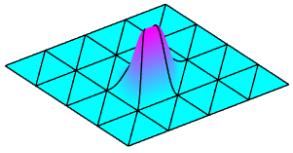
(f) C^0 cubic basis function.



(g) C^1 cubic basis function.



(h) C^0 quintic basis function.



(i) C^1 quintic basis function.

Figure 5: C^1 Bézier triangular meshes of various degrees and their basis functions. (a), (b) and (c) show dependent and free nodes with underlying meshes. Each of the larger red nodes in the centers has its associated basis functions plotted to its right. (d), (f) and (h) each shows a C^0 basis function associated with the corresponding center red node. It has cross-element continuity but its derivative does not. (e), (g) and (i) each shows a C^1 basis function associated with the corresponding center red node, which is a linear combination of nearby C^0 basis. Its derivative and itself both have cross-element continuity.

With the generalized-α time stepping method we have the tangent matrix $\mathbf{K}$ given by Eq. 27. The explicit matrix form is

$$
\begin{aligned}
\mathbf{K}_1^m &= \int \boldsymbol{\psi} \cdot \boldsymbol{\psi}^T d\Omega, \\
\mathbf{K}_1^k &= \int \nabla\boldsymbol{\psi} \cdot (\frac{\alpha}{3}\nabla\boldsymbol{\psi}^T) - 2\alpha\nabla\boldsymbol{\psi} \cdot (c^h\nabla\boldsymbol{\psi}^T + \nabla c^h\boldsymbol{\psi}^T - (c^h)^2\nabla\boldsymbol{\psi}^T - 2c^h\nabla c^h\boldsymbol{\psi}^T)d\Omega \\
&+ \int \nabla\boldsymbol{\psi} \cdot (\nabla c^h\Delta\boldsymbol{\psi}^T + \Delta c^h\nabla\boldsymbol{\psi}^T) - 2\nabla\boldsymbol{\psi} \cdot (c^h\nabla c^h\Delta\boldsymbol{\psi}^T + c^h\Delta c^h\nabla\boldsymbol{\psi}^T + \nabla c^h\Delta c^h\boldsymbol{\psi}^T)d\Omega \\
&+ \int \Delta\boldsymbol{\psi} \cdot (c^h\Delta\boldsymbol{\psi}^T + \Delta c^h\boldsymbol{\psi}^T - (c^h)^2\Delta\boldsymbol{\psi}^T - 2c^h\Delta c^h\boldsymbol{\psi}^T)d\Omega, \\
\mathbf{K}_1 &= \alpha_m\mathbf{K}_1^m + \alpha_f\gamma\Delta_t\mathbf{K}_1^k.
\end{aligned}
\tag{36}
$$

In our numerical implementation, we follow a procedure [16] that is similar to traditional Lagrange polynomials based FE implementation. That is, we first use C^0 Bernstein basis for assembling the matrices and then use the continuity $\mathbf{C}$ matrix to obtain the C^1 basis based matricies, a process that has also been used in NURBS-based isogeometric analysis and is referred to as Bézier extraction technique[29]. Recall Eq. 34, we substitute $\boldsymbol{\psi}$ by $\mathbf{C}\boldsymbol{\phi}$ in Eq. 35 and 36 and have

$$
\begin{aligned}
\mathbf{G}_1 &= \mathbf{C}^T\mathbf{G}_0\mathbf{C}, \\
\mathbf{K}_1 &= \mathbf{C}^T\mathbf{K}_0\mathbf{C},
\end{aligned}
\tag{37}
$$

where $\mathbf{G}_0$ and $\mathbf{K}_0$ are the residual vector and the tangent matrix assembled with C^0 Bézier basis, respectively. Their explicit matrix form are Eq. 35 and 36 with $\boldsymbol{\psi}$ replaced by $\boldsymbol{\phi}$, subscript 1 replaced by subscript 0. In each non-linear iteration, $\mathbf{G}_0$ and $\mathbf{K}_0$ are assembled, we then recover $\mathbf{G}_1$ and $\mathbf{K}_1$ with Eq. 37, solve the system and proceed as described in Section 3.3.

5 Numerical study of convergence rates

In this section, we adopt a manufactured solution based method [30] to analyze the convergence rate of our isogeometric approach. This method aims to provide reference solutions for complicated numerical problems where analytical solutions are not available, so it is also sometimes referred as a synthetic test. Further, we report the convergence comparison with Lagrange polynomials based FE to demonstrate the distinctive traits and potential numerical advantages of our approach.

5.1 A test problem with manufactured solution

In our test case we borrow the computational framework from [9] and we conduct our test computation in a two-dimensional periodic square $(0, 1)^2$, where the manufactured solution

takes the following form

$$\begin{aligned} &\dot{c} - \nabla(M_c \nabla(p_c - \Delta c)) = \mathcal{F}(\mathbf{x}, t), \\ &\mathcal{F}(\mathbf{x}, t) = \dot{c}^m - \nabla(M_c \nabla(p_c - \Delta c^m)), \end{aligned} \tag{38}$$

or explicitly

$$\begin{aligned} \mathcal{F}(\mathbf{x}, t) = \dot{c}^m - [(\nabla c^m - 2c^m \nabla c^m)(\frac{\partial p_c}{\partial c} \cdot \nabla c^m - \nabla(\Delta c^m)) \\ + c^m(1 - c^m)(\frac{\partial^2 p_c}{\partial c^2} \cdot \nabla c^m \cdot \nabla c^m + \frac{\partial p_c}{\partial c} \cdot \Delta c^m - \Delta(\Delta c^m))]. \end{aligned} \tag{39}$$

To comply with the periodic boundary condition, we select our manufactured solution to be

$$c^m(\mathbf{x}, t) = \frac{1}{2}(2\bar{c} + bt \prod_{i=1}^{2} \cos(a\pi x_i)), \tag{40}$$

where a can be set to satisfy the boundary condition and b is used to control the time scale.

Let $L = 1$ and $T^m = 0.01$ denote the dimension of the computational domain and the final time, respectively, we select $a = 6/L$ and $b = 0.3/T^m$, start with $\alpha = 1$ and a uniform concentration $\bar{c} = 0.5$, solve Eq. 38 in one time step, then perform convergence analysis with meshes of different resolutions and orders. One example of our numerical solution is shown in Fig. 6, where the solution was obtained with 3200 PS-split quadratic macro-elements (4800 free nodes).

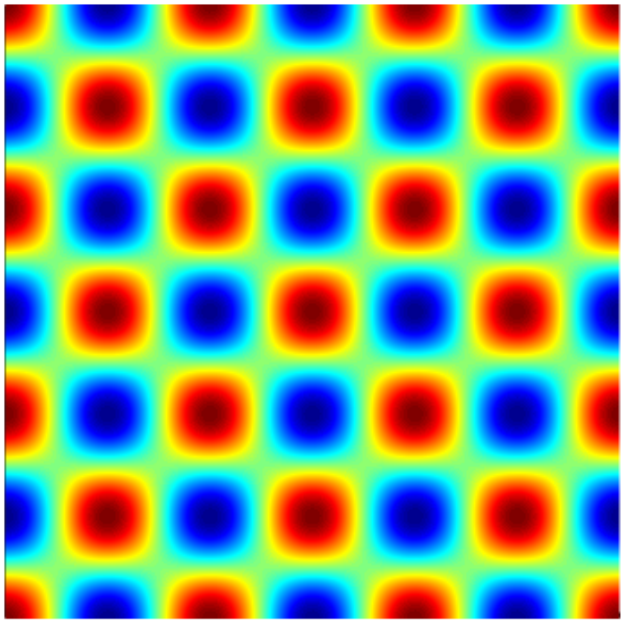


Figure 6: Numerical solution to Eq. 38. For this specific plot 3200 macro-elements (4800 free nodes) are in use.

5.2 The mixed formulation and element comparison

To perform a Lagrange polynomials based FEA with the manufactured solution for comparison we use a mixed form of the Cahn–Hilliard equation

$$\frac{\partial c}{\partial t} - \nabla \cdot (M_c \nabla \mu) = \mathcal{F}(\mathbf{x}, t) \quad \text{in} \quad \Omega, \tag{41.1}$$

$$\mu = (f_c' - \lambda \Delta c) \quad \text{in} \quad \Omega. \tag{41.2}$$

This coupled system of equations has been studied extensively in literature. It introduces two second-order PDEs by adding an extra field variable μ thus suitable for analysis with C^0 basis, which is why we use it for FEA. In our experiment with Lagrange polynomials based FEA we are using FEniCS solver version 2018.1.0.r2[31].

We give a visual comparison between tIGA elements and FEA elements in Fig. 7. Naturally, almost all nodes in FEA meshes are free expect for those boundary nodes rendered dependent by the periodic boundary condition. In tIGA meshes dependent nodes are created because of enforcing continuity constraints. For the same number of elements, it can be clearly seen in the figure that these two classes of meshes employ different numbers of Degrees of Freedom (DoF). This leads to a discrepancy in numerical performance which will be discussed in the next section.

Remark: In tIGA since we use the primal variational form of the Cahn–Hilliard equation the number of DoF is the same as the number of free nodes. But in our FEA, there are 2 DoF per free node. Since we consider both free nodes and DoF in our analysis, we note this fact here to avoid confusion.

5.3 The convergence

We compare the numerical solutions against the manufactured solution. In this work L^2 error and H^1 error are defined as

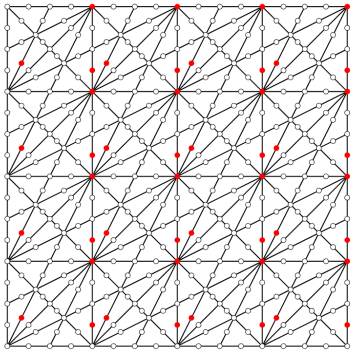
$$\|c^h - c^m\|_{L^2} = \sqrt{\int [c^h - c^m]^2 dV}, \tag{42}$$

$$\|c^h - c^m\|_{H^1} = \sqrt{\|c^h - c^m\|_{L^2}^2 + \|\nabla(c^h - c^m)\|_{L^2}^2}. \tag{43}$$

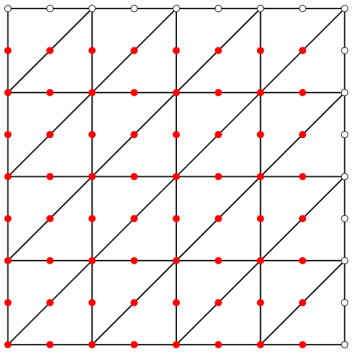
And the element size h is defined as the length of a leg of a triangle macro-element in tIGA or an element in FEA.

The result of the convergence analysis can be found in Fig. 8, 9 and 10. Table 1 contains corresponding mesh resolution information of the points used to fit each line in those plots.

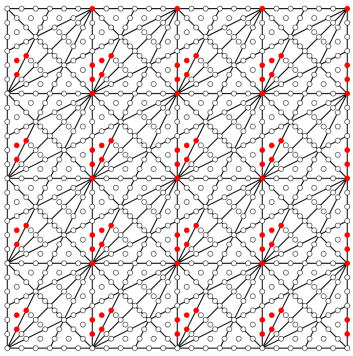
Fig. 8 is about the convergence of L^2 error in a log–log scale. The 3 plots in the left column show the convergence of quadratic, cubic and quintic elements, with respect to the element size h. The 3 plots in the right column show the convergence of these 3 types of elements with respect to the square root of the total number of free nodes, one type in each row. From Fig. 8(a) and (d) we see tIGA with quadratic elements shows the convergence



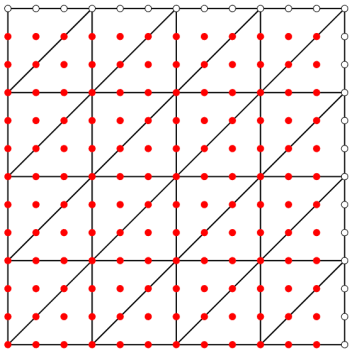
(a) C^1 quadratic tIGA mesh.



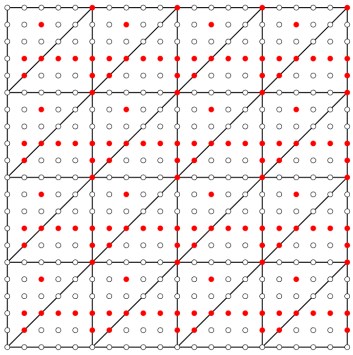
(d) C^0 quadratic FEA mesh.



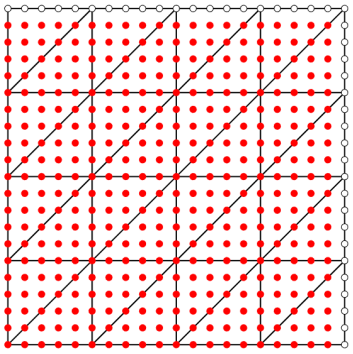
(b) C^1 cubic tIGA mesh.



(e) C^0 cubic FEA mesh.



(c) C^1 quintic tIGA mesh.



(f) C^0 quintic FEA mesh.

Figure 7: Comparison of tIGA elements and FEA elements of different orders (with periodic boundary conditions) used in the convergence analysis. Red dots are free nodes while white dots are dependent nodes.

Table 1: Mesh data used in the convergence analysis. n_M stands for the number of macro-elements; n_m stands for the number of micro-elements; n_e stands for the number of elements (in cases without split); n_{node} stands for the number of free nodes; n_{dof} stands for the number of DoF.

Element Type	Point 1	Point 2	Point 3	Point 4	Point 5
tIGA Quadratic	n_M: 2×20^2 n_m: 12×20^2 n_{node}: 1200 n_{dof}: 1200	n_M: 2×30^2 n_m: 12×30^2 n_{node}: 2700 n_{dof}: 2700	n_M: 2×40^2 n_m: 12×40^2 n_{node}: 4800 n_{dof}: 4800	n_M: 2×50^2 n_m: 12×50^2 n_{node}: 7500 n_{dof}: 7500	n_M: 2×60^2 n_m: 12×60^2 n_{node}: 10800 n_{dof}: 10800
tIGA Cubic	n_M: 2×10^2 n_m: 6×10^2 n_{node}: 600 n_{dof}: 600	N_M: 2×12^2 n_m: 6×12^2 n_{node}: 864 n_{dof}: 864	n_M: 2×15^2 n_m: 6×15^2 n_{node}: 1350 n_{dof}: 1350	n_M: 2×20^2 n_m: 6×20^2 n_{node}: 2400 n_{dof}: 2400	n_M: 2×25^2 n_m: 6×25^2 n_{node}: 3750 n_{dof}: 3750
tIGA Quintic	n_e: 2×8^2 n_{node}: 576 n_{dof}: 576	n_e: 2×12^2 n_{node}: 1296 n_{dof}: 1296	n_e: 2×15^2 n_{node}: 2025 n_{dof}: 2025	n_e: 2×20^2 n_{node}: 3600 n_{dof}: 3600	n_e: 2×25^2 n_{node}: 5625 n_{dof}: 5625
Lagrange Quadratic	n_e: 2×20^2 n_{node}: 1600 n_{dof}: 3200	n_e: 2×30^2 n_{node}: 3600 n_{dof}: 7200	n_e: 2×40^2 n_{node}: 6400 n_{dof}: 12800	n_e: 2×50^2 n_{node}: 10000 n_{dof}: 20000	n_e: 2×60^2 n_{node}: 14400 n_{dof}: 28800
Lagrange Cubic	n_e: 2×12^2 n_{node}: 1296 n_{dof}: 2592	n_e: 2×15^2 n_{node}: 2025 n_{dof}: 4050	n_e: 2×20^2 n_{node}: 3600 n_{dof}: 7200	n_e: 2×25^2 n_{node}: 5625 n_{dof}: 11250	n_e: 2×30^2 n_{node}: 8100 n_{dof}: 16200
Lagrange Quintic	n_e: 2×5^2 n_{node}: 625 n_{dof}: 1250	n_e: 2×8^2 n_{node}: 1600 n_{dof}: 3200	n_e: 2×10^2 n_{node}: 2500 n_{dof}: 5000	n_e: 2×12^2 n_{node}: 3600 n_{dof}: 7200	n_e: 2×15^2 n_{node}: 5625 n_{dof}: 11250

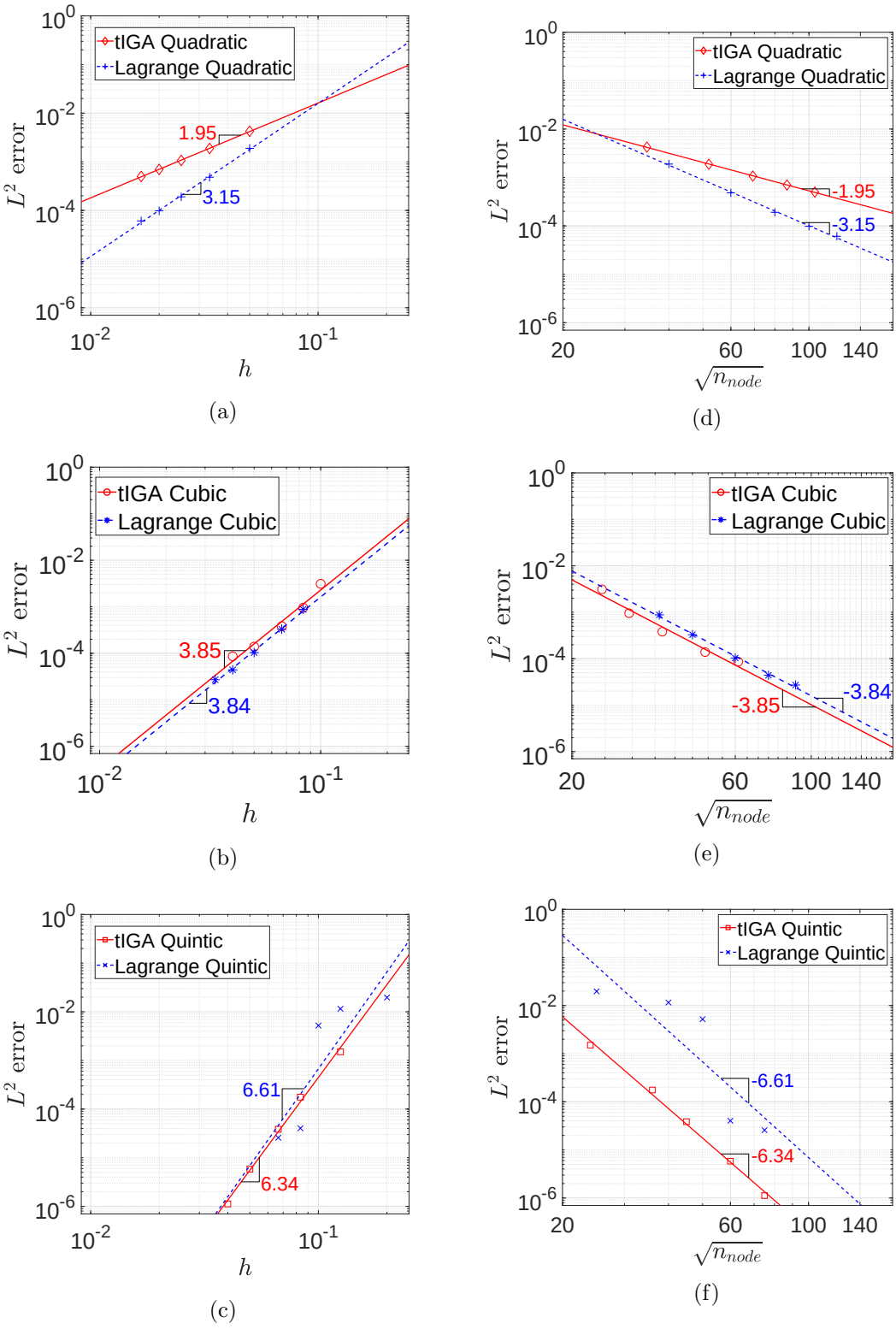


Figure 8: L^2 error plots for elements of different orders. (a), (b) and (c) show error versus the element size h. (d), (e) and (f) show error versus $\sqrt{n_{node}}$, the number of free nodes along each direction.

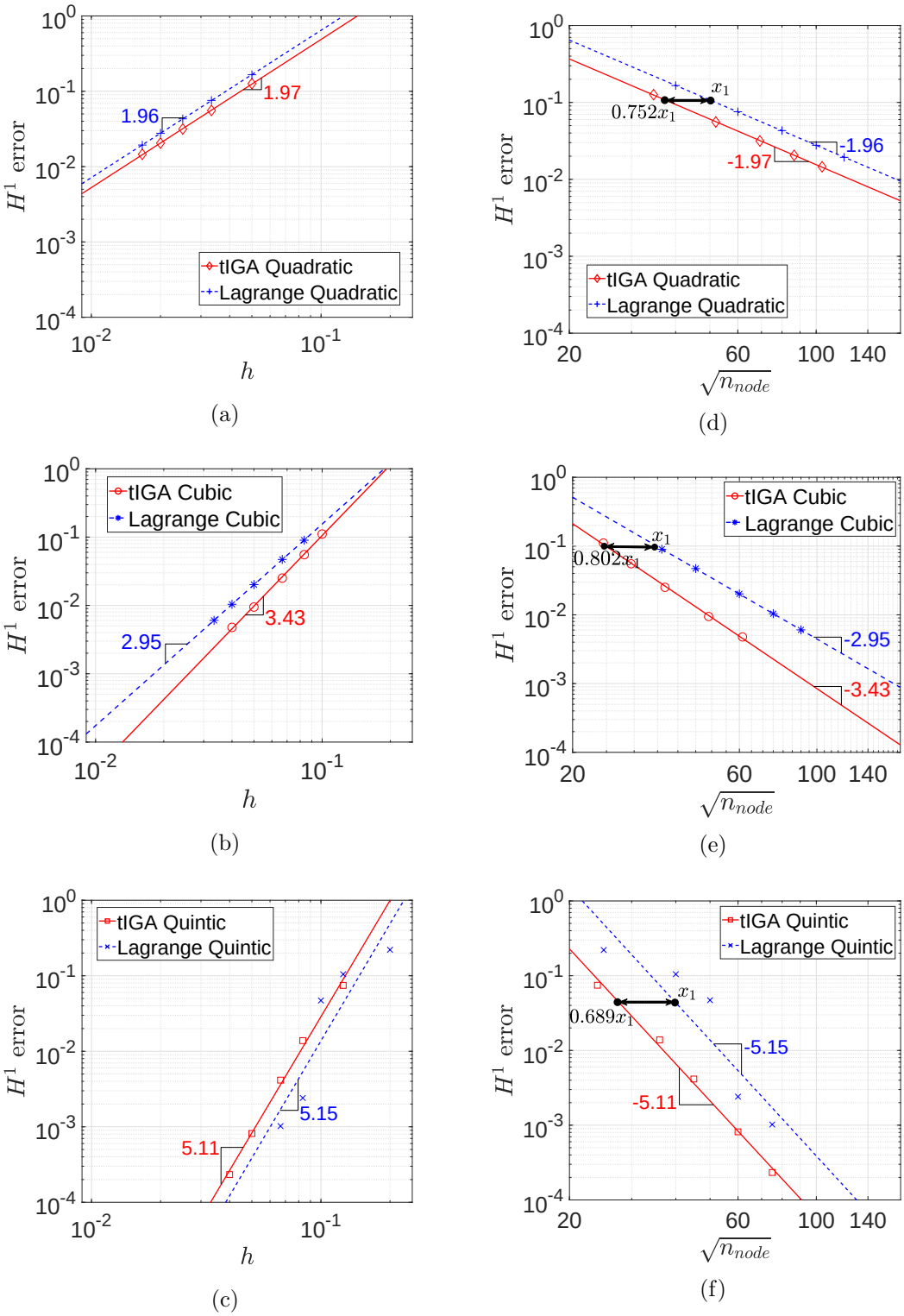
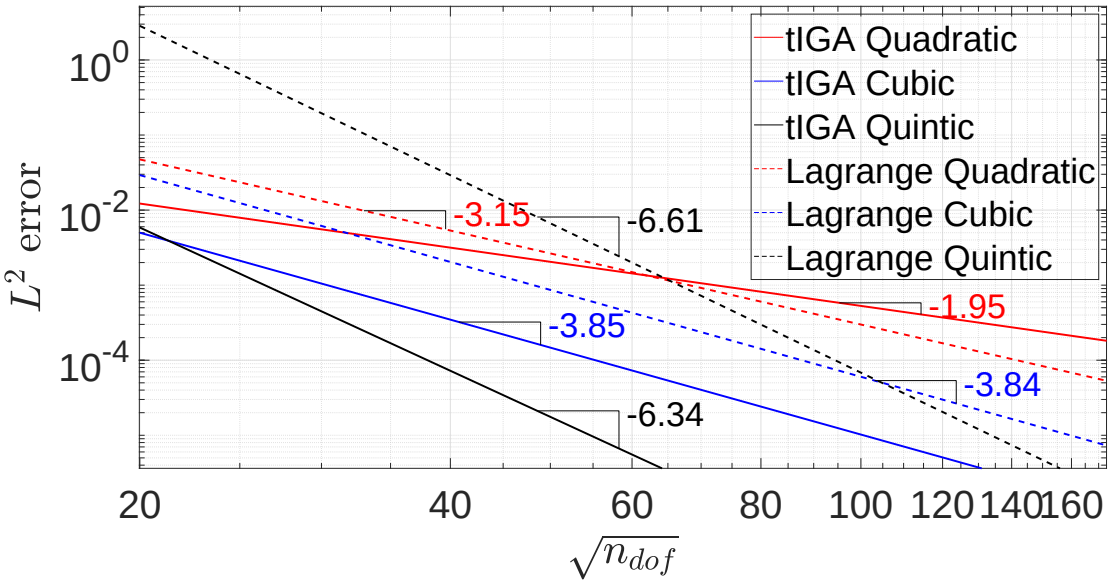
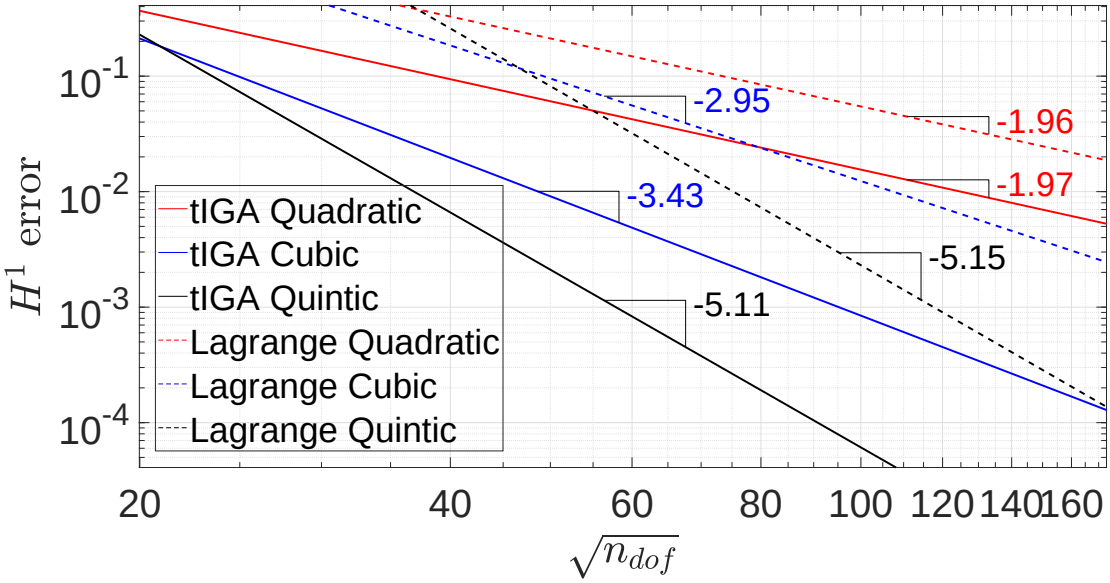


Figure 9: H^1 error plots for elements of different orders. (a), (b) and (c) show error versus the element size h. (d), (e) and (f) show error versus $\sqrt{n_{node}}$, the number of free nodes along each direction.



(a) L^2 error plot.



(b) H^1 error plot.

Figure 10: Summary of L^2 and H^1 error analysis. The error is plotted against $\sqrt{n_{dof}}$, the number of DoF along each direction. In this plot the scattered data points are omitted.

rate of 1.95. It is about 1 order lower than that of FEA which is 3.15. Unlike the previous case, in Fig. 8(b) and (e) tIGA with cubic elements shows the convergence rate of 3.85, almost the same as that of FEA, 3.84. In Fig. 8(c) and (f) tIGA with quintic elements shows the convergence rate of 6.34 and is also considered roughly the same order of FEA which is 6.61.

Fig. 9 is about the convergence of H^1 error in a log–log scale. The 3 plots in the left column show the convergence of quadratic, cubic and quintic elements, with respect to the element size h. The 3 plots in the right column show the convergence of these 3 types of elements with respect to the square root of the total number of free nodes. From Fig. 9(a) and (d) we see tIGA with quadratic elements shows the convergence rate of 1.97. It is about the same order of FEA which has the rate 1.96. The similar conclusion can be drawn for cubic and quintic elements, where tIGA also shows the convergence rate of the same order of FEA. This means all tIGA H^1 error test cases recover the convergence rate of p, the same rate expected in FEA.

It is notable that tIGA with quadratic and quintic elements has nodal advantages over FEA. In Fig. 9(d), for the same H^1 error tIGA with quadratic elements uses 0.752 times the free nodes along each direction compared to FEA. In Fig. 9(e), tIGA with cubic elements needs 0.802 times the free nodes along each direction to get the same H^1 error as FEA. For quintic elements in tIGA the number is 0.689.

Because FEA employs 2 DoF per free node due to the usage of the mixed form, when taking this into account the performance gain of tIGA can be more considerable. To this end, we summarize our convergence analysis in Fig. 10, where error is plotted against the numbers of DoF along each direction.

Remarks:

(1) In our experiments tIGA exhibits the optimal L^2 error convergence rate of $p+1$ for $p \geq 3$, where p is the element order. Unfortunately, tIGA with quadratic elements gives the rate of 2 instead (see Fig. 8(a) and (d)). In fact, this is not unexpected. Studies[32, 33, 34] show both mathematically and numerically that due to the presence of high-order operators in the primal variational form of the Cahn–Hilliard equation, the optimal L^2 convergence rate in IGA with B-spline or NURBS is indeed 2 for quadratic elements, and recovers to be $p+1$ for higher-order elements. This is in line with our results in tIGA.

(2) We experimented using both PS and CT split to generate C^1 cubic macro-element spaces[15]. As shown in the paper, tIGA with PS-split cubic elements gives the L^2 error convergence rate of 4, which is optimal. Unfortunately, for CT-split cubic elements the rate is 3 instead. We have not yet concluded why and how the split scheme affects the L^2 error convergence. Note that PS-split and CT-split macro-element spaces are different, but for the same mesh refinement they incur the same number of free nodes, because both of them remove geometrical dependency of the dimension of triangulated bivariate spline spaces.

(3) We provide a brief explanation of the nodal advantages of tIGA shown in H^1 error tests. For two-dimensional meshes, give a certain element order p and element size h, let m_{tIGA} be the number of free nodes in the tIGA mesh, and m_{FEA} be the number of free nodes in the FEA mesh. Table 2 analyzes quadratic, cubic and quintic elements and reveals each of them has a fixed ratio between m_{tIGA} and m_{FEA}. This ratio is always smaller than 1, meaning for the same element size tIGA uses fewer free nodes.

To further show this ratio is related to the nodal advantages, we provide Table 3, where for a fixed H^1 error, we compare h^{FEA}/h^{tIGA} (the ratio of element number per unit length between tIGA and FEA), $n_{node}^{tIGA}/n_{node}^{FEA}$ (the ratio of the number of free nodes between tIGA and FEA) and $n_{dof}^{tIGA}/n_{dof}^{FEA}$ (the ratio of the number of DoF between tIGA and FEA). Again quadratic, cubic and quintic elements are analyzed. For all cases in Table 3, though the ratio h^{FEA}/h^{tIGA} is close to 1, tIGA requires much fewer free nodes, since the ratio $n_{node}^{tIGA}/n_{node}^{FEA}$ is way below 1. This nodal performance gain is the result of the aforementioned nodal ratio, namely

$$\left(\frac{h^{FEA}}{h^{tIGA}}\right)^2 \times \frac{m_{tIGA}}{m_{FEA}} = \frac{n_{node}^{tIGA}}{n_{node}^{FEA}}. \tag{44}$$

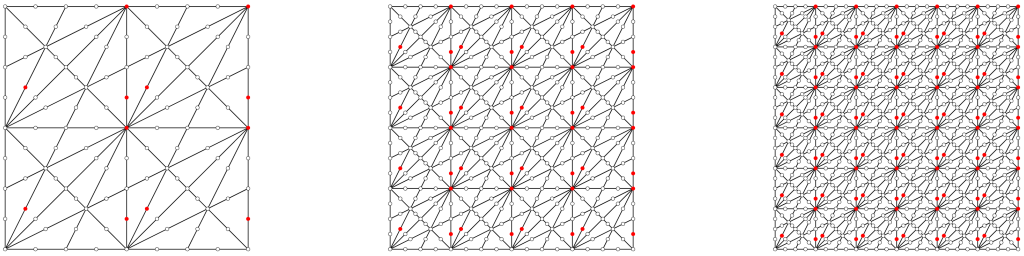
In terms of DoF, tIGA elements show even more significant per-degree-of-freedom advantages over FEA elements of the same order.

The nodal advantages claimed in Table 3 agree with Fig. 9(d), (e), (f) and Fig. 10(b).

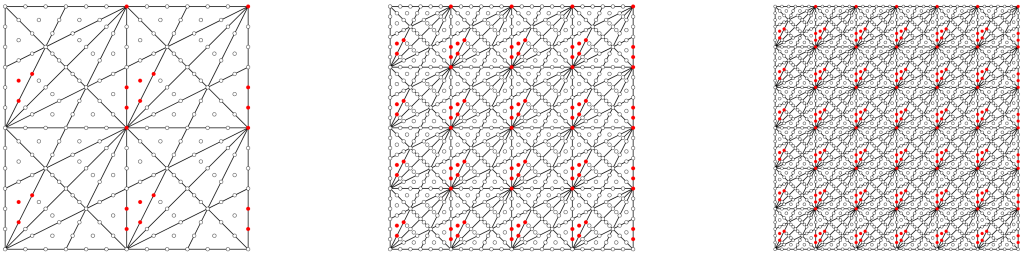
Table 2: The comparison between tIGA and FEA meshes in terms of the node numbers.

Element Order	Refinement 1	Refinement 2	Refinement 3	Nodal Ratio
Quadratic	h: 0.5 m_{tIGA}: 12 m_{FEA}: 16	h: 0.25 m_{tIGA}: 48 m_{FEA}: 64	h: 0.125 m_{tIGA}: 108 m_{FEA}: 144	$\frac{m_{tIGA}}{m_{FEA}} = \frac{3}{4}$
Cubic	h: 0.5 m_{tIGA}: 24 m_{FEA}: 36	h: 0.25 m_{tIGA}: 96 m_{FEA}: 144	h: 0.125 m_{tIGA}: 216 m_{FEA}: 324	$\frac{m_{tIGA}}{m_{FEA}} = \frac{2}{3}$
Quintic	h: 0.5 m_{tIGA}: 36 m_{FEA}: 100	h: 0.25 m_{tIGA}: 144 m_{FEA}: 400	h: 0.125 m_{tIGA}: 324 m_{FEA}: 900	$\frac{m_{tIGA}}{m_{FEA}} = \frac{9}{25}$

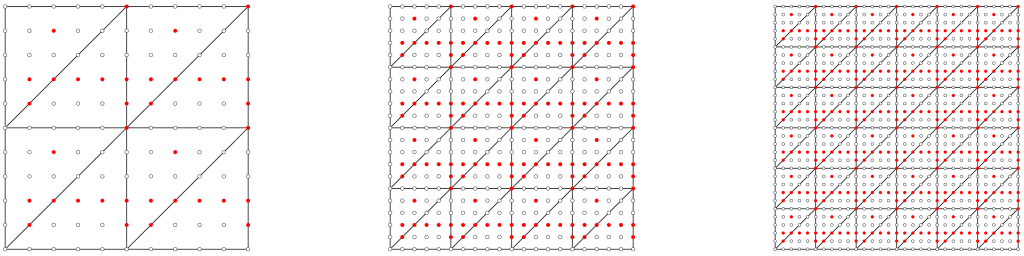
(4) In our FEA with quintic Lagrange elements, we observe that data points are somewhat distant from the regression lines. This is probably because the numbers of elements in each direction are low, only on pair with the wave number of the manufactured solution. We suspect, as claimed in [35], higher-order Lagrange basis does not adjust as well to fast changes compared to its lower-order counterparts thus fluctuates in



(a) Refining quadratic tIGA meshes.



(b) Refining cubic tIGA meshes.



(c) Refining quintic tIGA meshes.

Figure 11: Refinement of tIGA meshes of different orders that are present in Table 2.

Table 3: Nodal and DoF advantages comparison in terms of a fixed H^1 error 10^{-1}.

Element Order	$\frac{h^{FEA}}{h^{tIGA}}$	$\frac{n_{node}^{tIGA}}{n_{node}^{FEA}}$	$\frac{n_{dof}^{tIGA}}{n_{dof}^{FEA}}$
Quadratic elements	0.868	$0.868^2 \times \frac{3}{4} = 0.565$	$0.565 \times \frac{1}{2} = 0.283$
Cubic elements	1.018	$1.018^2 \times \frac{2}{3} = 0.691$	$0.691 \times \frac{1}{2} = 0.346$
Quintic elements	1.148	$1.148^2 \times \frac{9}{25} = 0.474$	$0.474 \times \frac{1}{2} = 0.237$

performance in this experiment. Nevertheless, the FEA data with quintic Lagrange polynomials still exhibits convergence rates close to that of the theory so we consider them valid for comparison. In contrast, we do not see such performance fluctuation in our tIGA with quintic basis in Fig. 8(f) and Fig. 9(f).

6 Numerical examples

In this section, we present numerical examples of the Cahn–Hilliard phase-field model, which can provide understanding into the evolution of the systems and demonstrate the capacity of our tIGA. We show numerical results under various initial conditions, model parameters and domain configurations. The first example is the system evolution from a random initial condition in two-dimensional periodic squares, which serve as a common benchmark in phase-field problems. To demonstrate the parameterization of complex domains with tIGA, the second example simulates phase separation in such a domain. The third example conducts the computation with a locally refined Bézier triangular mesh and explain the relevance of local refinement in phase-field problems. The last example is a three-dimensional extension of the periodic square example.

6.1 Two-dimensional system evolution

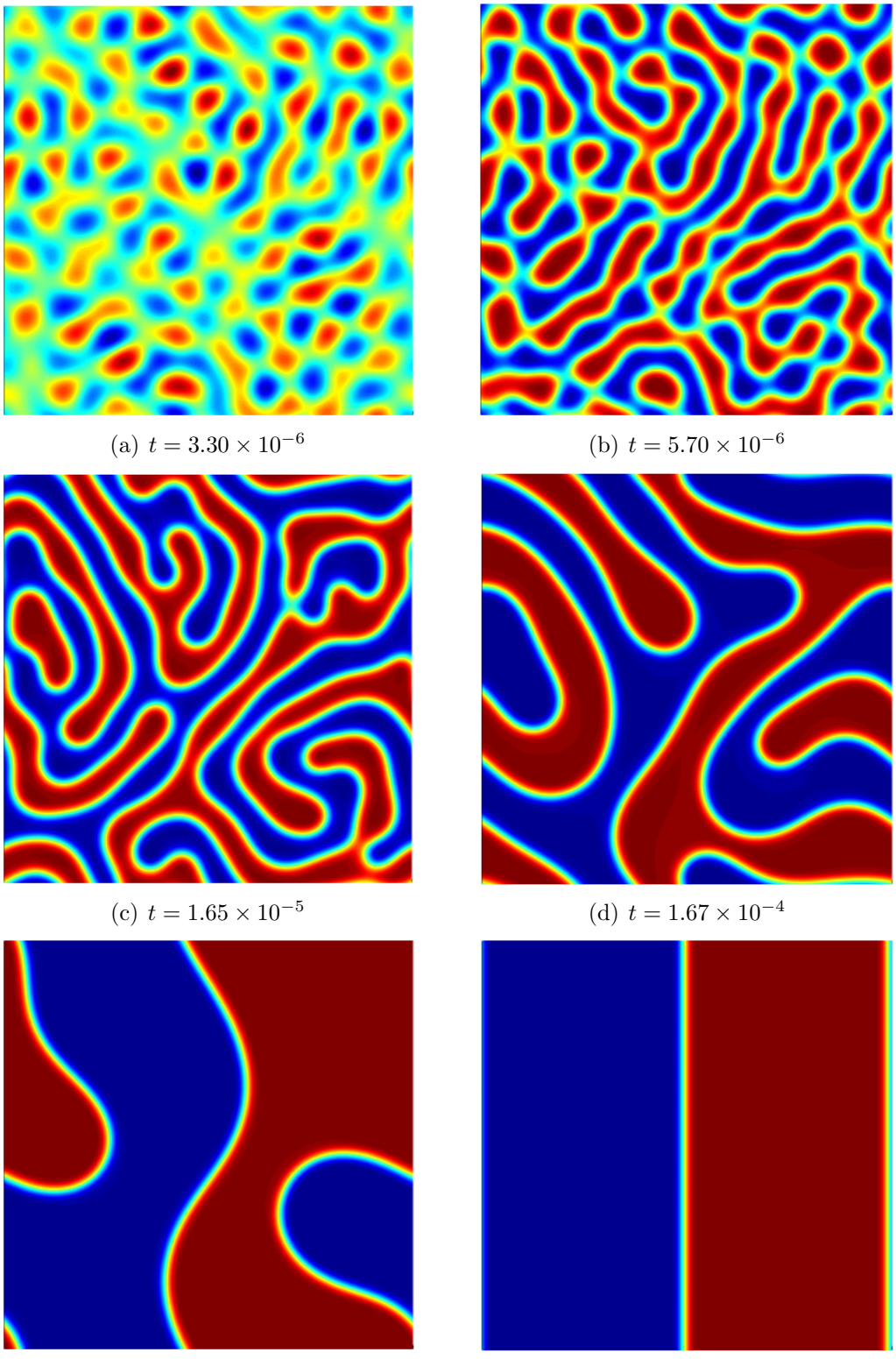
We first conduct this computation in a two-dimensional unit square domain $(0, 1)^2$. Periodic boundary condition, which allows the species that leave the domain reenter from the opposite side, is enforced on domain boundary. The initial concentration $c(\mathbf{x}, t = 0)$ is set to be a small perturbation around an average concentration $\bar{c}$, namely $c(\mathbf{x}, t = 0) = \bar{c} + r(\mathbf{x})$. Throughout the numerical examples in this paper, different average concentrations were studied and $r(\mathbf{x})$ is kept to be a random variable with a uniform distribution in $[-0.05, 0.05]$.

For the first example we set $\bar{c} = 0.5$ and $\alpha = 9000$. Our spatial discretization here is a C^1-continuous quadratic mesh with PS split. It consists of 7200 macro-elements with 10800 free nodes (86400 total nodes). The complete system evolution is demonstrated in Fig. 12. The system evolves into connected bands across the entire domain. After a binodal concentration is roughly achieved, the coarsening effect takes place in a much larger time scale, as the transient surface area gradually decreases. The steady state contains two parallel surfaces between phases.

Motivated by [3], we further examine the second order statistical moment M_2 of the system defined by

$$M_2 = \int (c - \bar{c})^2 d\Omega, \tag{45}$$

in order to characterize and quantify the system evolution. The result in Fig. 13(a) sees a sharp increase from $t = 5 \times 10^{-6}$ to $t = 10^{-5}$, but a much slower climb after that. This further demonstrates our observation in the previous paragraph and highlights the importance of addressing the temporal multi-scale nature of this problem through the adaptive



(a) $t = 3.30 \times 10^{-6}$

(b) $t = 5.70 \times 10^{-6}$

(c) $t = 1.65 \times 10^{-5}$

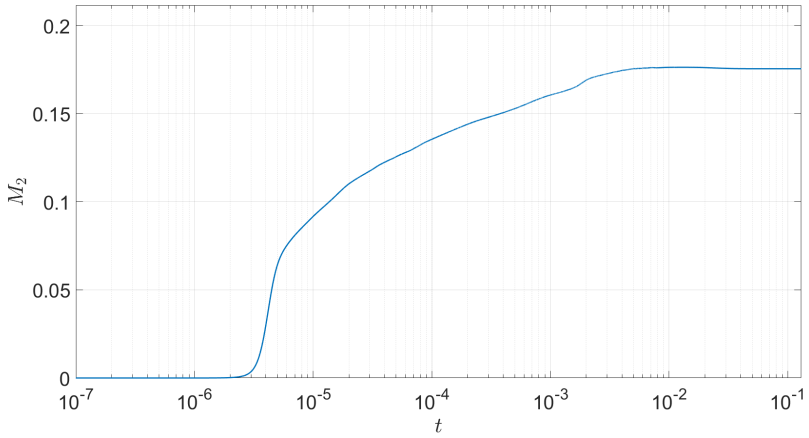
(d) $t = 1.67 \times 10^{-4}$

(e) $t = 2.14 \times 10^{-3}$

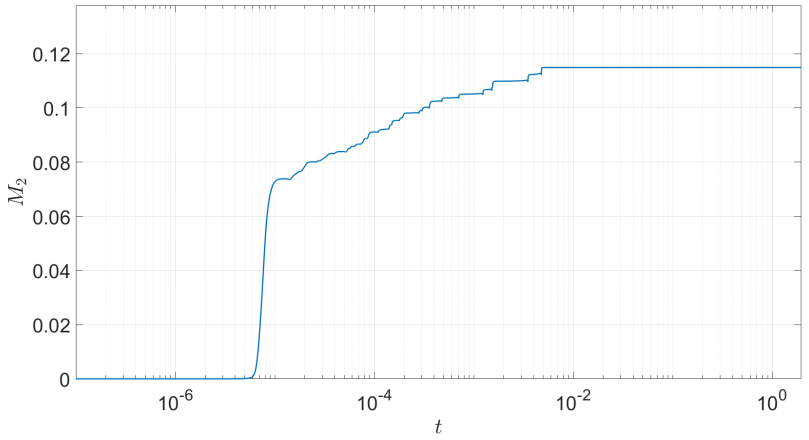
(f) Steady state

Figure 12: The evolution of the concentration from an initial concentration $\bar{c} = 0.5$ and $\alpha = 9000$ in a periodic square.

time stepping scheme.



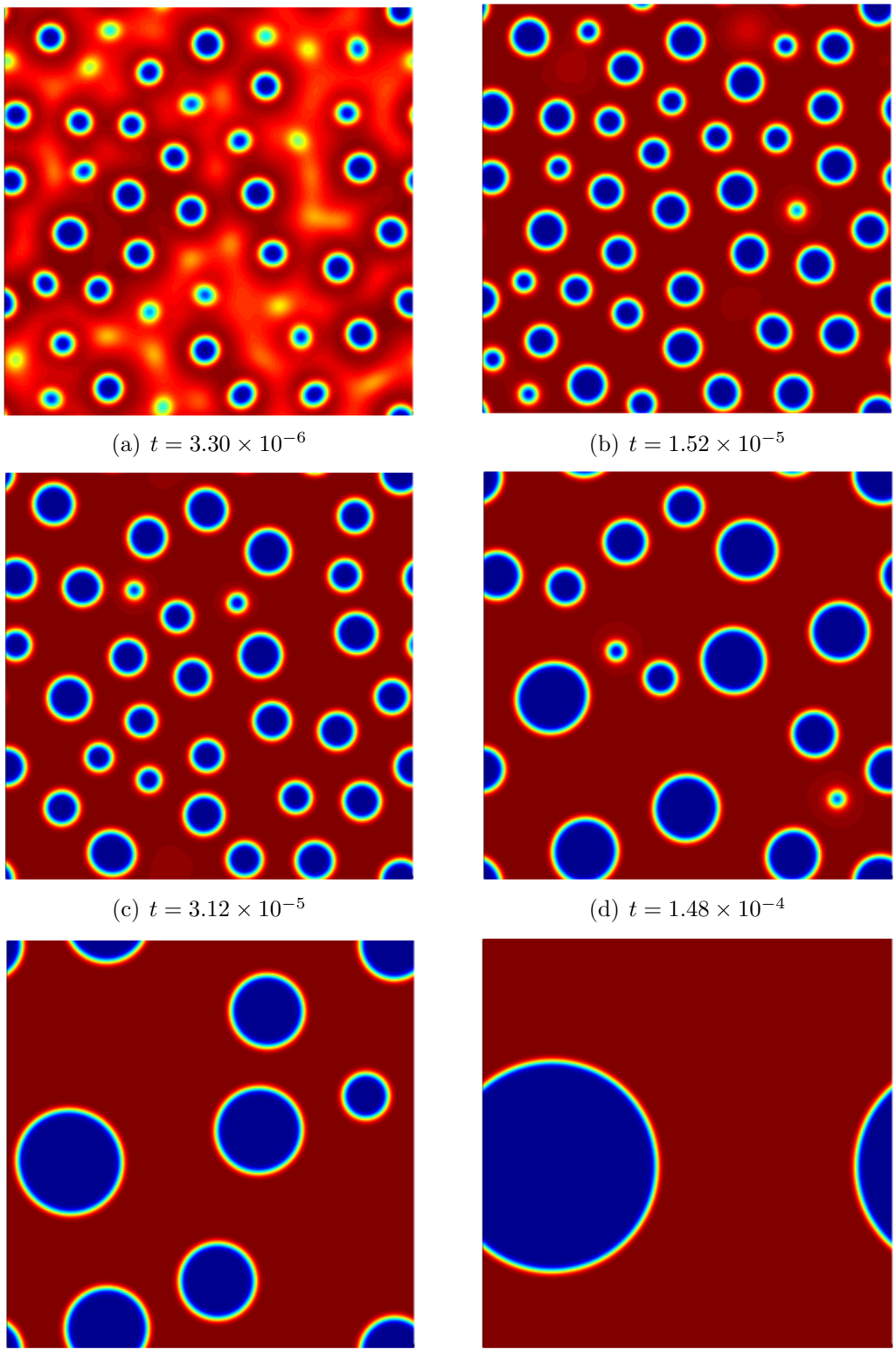
(a) Initial concentration $\bar{c} = 0.5$.



(b) Initial concentration $\bar{c} = 0.75$.

Figure 13: The evolution of the second-order moment in a periodic square.

In the second example we employ a C^1-continuous quintic mesh with 12800 quintic elements, 57600 free nodes (160000 total nodes). No split scheme is enforced. α is set to be 20000 and $\bar{c} = 0.75$. The system evolution is shown in Fig. 14. For the corresponding moment evolution see Fig. 13(b). Compared to the previous experiment, we observe generally faster phase separation. The topology of the solution differs from the former case, separated nucleation happens. In the end, the system reaches its steady state with a single bubble being present.



(a) $t = 3.30 \times 10^{-6}$

(b) $t = 1.52 \times 10^{-5}$

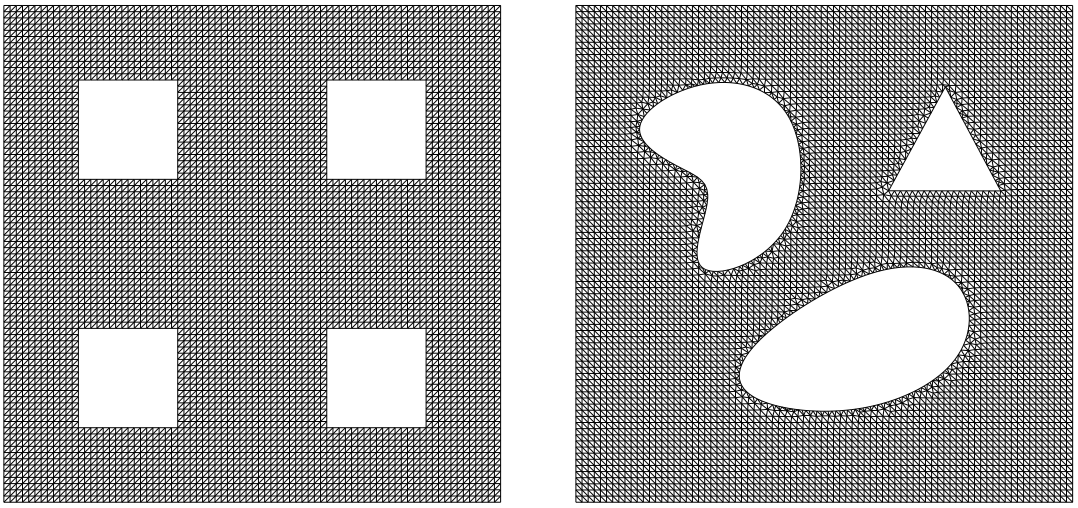
(c) $t = 3.12 \times 10^{-5}$

(d) $t = 1.48 \times 10^{-4}$

(e) $t = 6.25 \times 10^{-4}$

(f) Steady state

Figure 14: The evolution of the concentration from an initial concentration $\bar{c} = 0.75$ and $\alpha = 20000$ in a periodic square.



(a) The mesh with 4 square holes.

(b) The mesh with 3 irregular holes.

Figure 15: The triangular mesh parameterizing a domain involving complex topology. The wireframes of macro-elements is shown.

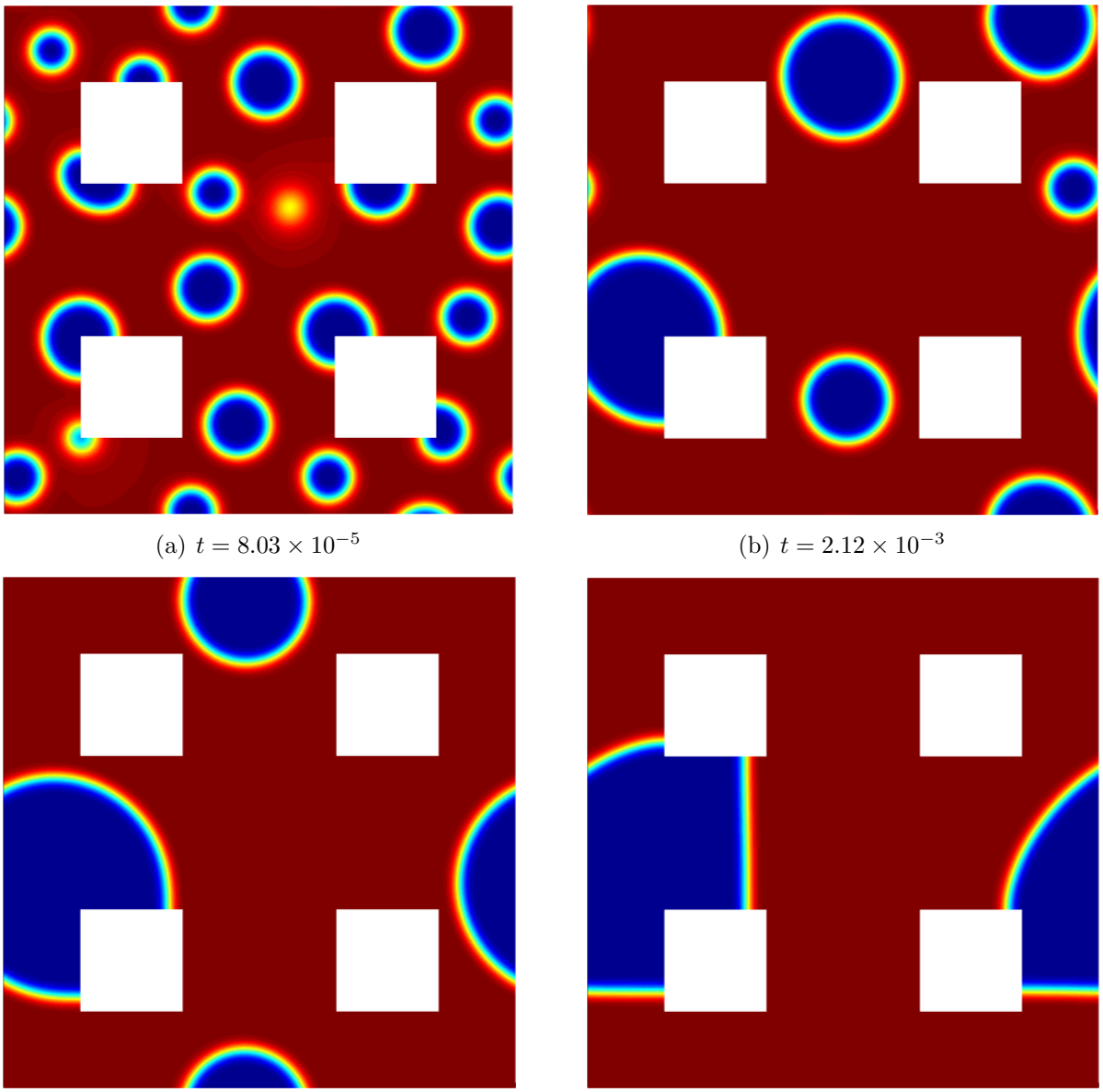
6.2 Phase-field model over complex topology

Previous examples include only simple domains. Such problem setup may be useful as a numerical benchmark but it is not easily extendable to realistic applications. In practical problems complex topology is often required. Solving phase field models over such complex topology poses difficulties for tensor-product NURBS based IGA, and is especially suitable for Bézier triangles and tetrahedra due to their topological flexibility. To this end, we present two two-dimensional benchmark examples containing topology that are difficult to model with single NURBS patch.

We start with a different topology in the first example. On top of the aforementioned periodic unit square domain, the computational domain now consists of four solid squares that each has the dimension of 0.2×0.2. The centers of the squares are positioned at coordinate $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.75, 0.25)$ and $(0.75, 0.75)$, respectively. This experiment uses a C^1 mesh consisting of 10752 quadratic macro-elements with PS split, 16500 free nodes out of 134045 total nodes. The exact mesh is given in Fig. 15(a). We show the wireframe of macro-elements here.

For this numerical example we set $\alpha = 9000$ and start from an initial concentration of $\bar{c} = 0.75$ with perturbation. The system evolution is given in Fig. 16. One of the species clearly nucleates. The spheres formed in the process merge over time. We observe that in the process the spheres attached to the walls have the contact angles kept 90 degrees (as suggested by Eq. 13.4). The steady state solution features a single deformed bubble. The surface of this bubble adheres to the solid walls so is divided into 3 parts, with some of them being straight. This is unlike the previous experiment in a simple periodic domain.

We further demonstrate the advantages of tIGA with a second example, where we incor-



(a) $t = 8.03 \times 10^{-5}$

(b) $t = 2.12 \times 10^{-3}$

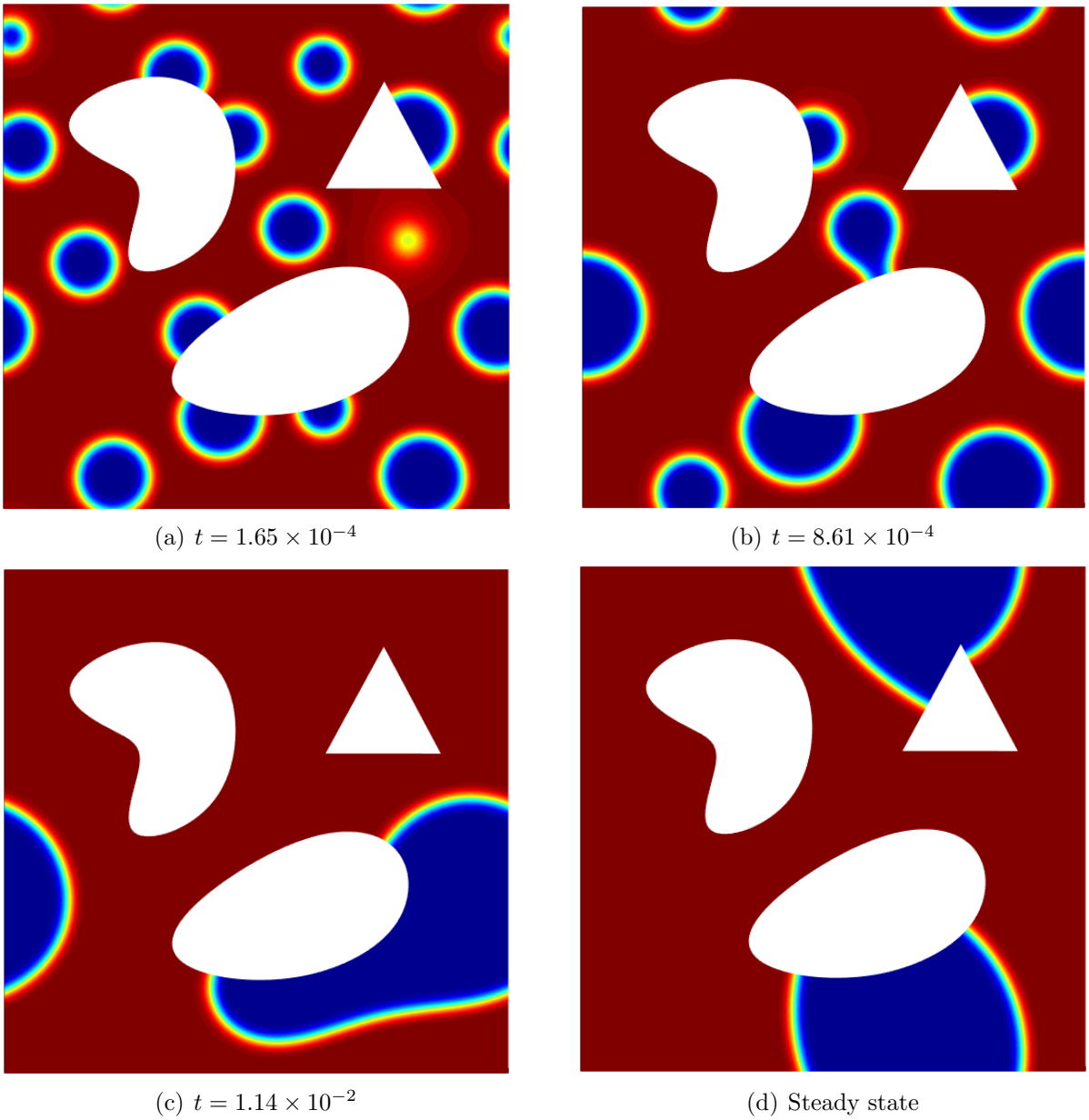
(c) $t = 7.92 \times 10^{-3}$

(d) Steady state

Figure 16: The system evolution of the concentration from an initial profile with $\bar{c} = 0.75$ and $\alpha = 9000$. The periodic domain contains square islands with solid-wall boundaries.

porate "general" shapes and asymmetry. The corresponding mesh is in Fig. 15(b), where again the wireframe of macro-elements is shown. 3 "islands" of irregular shapes are added to a periodic unit square domain. The C^1 mesh consists of 10450 quadratic marco-elements, 16176 free nodes out of 130590 total nodes.

In this example, we also set $\alpha = 9000$ and $\bar{c} = 0.75$. The system evolution is provided in Fig. 17. Again, the steady state solution is a single bubble attached to the walls, minimizing the surface area.



(a) $t = 1.65 \times 10^{-4}$ (b) $t = 8.61 \times 10^{-4}$

(c) $t = 1.14 \times 10^{-2}$ (d) Steady state

Figure 17: The system evolution of the concentration from an initial profile with $\bar{c} = 0.75$ and $\alpha = 9000$. The periodic domain contains irregular islands with solid-wall boundaries.

We see our approach is applicable to domains of complex topology without extraneous processing. Our mesh in this experiment remains to be in one single patch and the C^1 continuity is preserved everywhere. These examples demonstrate that our C^1 tIGA can adapt to Cahn–Hillard problems of complex topology.

Remark: The walls of the internal islands has boundary conditions Eq. 13.3 and Eq. 13.4 imposed. Our numerical implementation is a weak imposition, the same as presented in [36]. Since the right hand sides of Eq. 13.3 and 13.4 are always 0 in our examples, we only need to add the following terms to the Galerkin form Eq. 22:

$$\int_\Gamma (\frac{a}{h_e}\nabla w^h \cdot \mathbf{n} - \Delta w^h) \cdot (m_c^h \nabla c^h \cdot \mathbf{n}) d\Gamma - \int_\Gamma (\nabla w^h \cdot \mathbf{n}) \cdot (m_c^h \Delta c^h) d\Gamma, \tag{46}$$

where a is a penalty parameter, $\mathbf{n}$ is the surface outward normal and h_e is the element length scale. We follow [36], set $a = 5$ and $h_e = \sqrt{2A_e}$, where A_e is the element area.

6.3 Local refinement example

For the Cahn–Hilliard model, the physics in the transient layer needs to be captured by the mesh. In many problems this layer is required to be extremely thin, thus could benefit from the use of local refinement in the solution process. tIGA is particularly suitable for such local refinement. It is often argued that for quality control, the thin transient layer needs to be captured by at least three elements[3, 35]. Since not all regions in the computational domain demand such resolution, the computational cost of the Cahn–Hilliard phase-field model can be reduced to a fraction by the local refinement of the transient region on a relatively coarse mesh. Here, the efficacy of tIGA concerning local refinement is demonstrated with a numerical example.

The initial concentration is shown in Fig. 18. The purpose is to mimic a scenario where the position of the transient layer is roughly known. In this computation the domain remains to be $(0, 1)^2$ two-dimensional square, but we use soild wall boundary conditions, meaning Eq. 13.3 and 13.4 are satisfied on all 4 sides. In this way, we focus only on capturing the layer sufficiently in the middle of the domain. We set $\alpha = 15000$ which is higher to make the affect of mesh resolution more pronounced.

In Fig. 19(a), 968 quintic elements are used to parameterize the domain in a triangulated uniform grid style. Fig. 19(b) shows an oscillation pattern and the distorted layer since the mesh does not possess enough resolution to capture the layer. The concentration along the line $x = 0.5$ is given in Fig. 19(c) as an illustration of this statement, where the transient layer appears to be captured by only 2 elements, with oscillations being present in neighboring elements.

In Fig. 19(d) however, only 952 quintic elements are present but since they are locally refined at the approximate position of the layer, the resolution is shown to be sufficient to recover the physically correct transient layer, as in Fig. 19(e). Again, the concentration along the line $x = 0.5$ is given in Fig. 19(f). We see in this case the layer is captured by 4 elements.

Due to the triangulation nature, tIGA allows local refinement precisely at the loca-

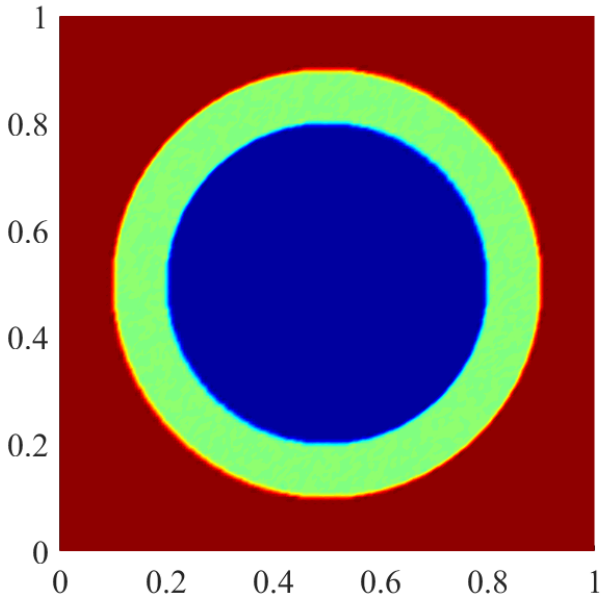
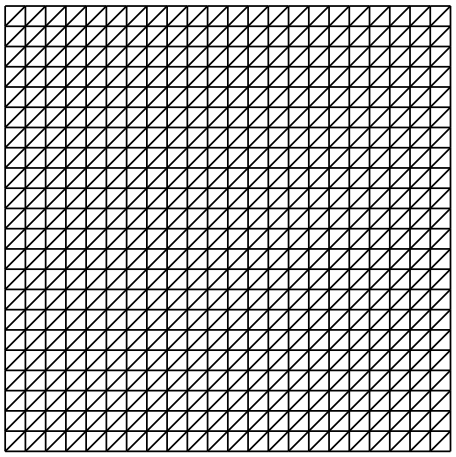
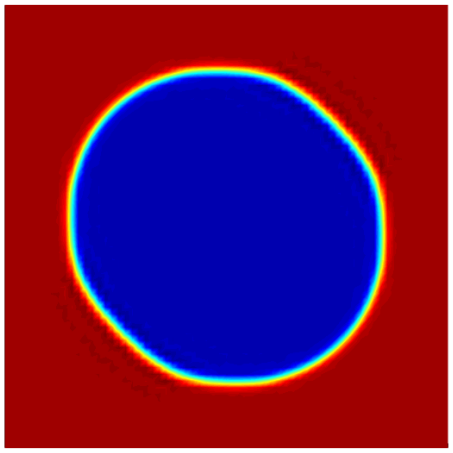


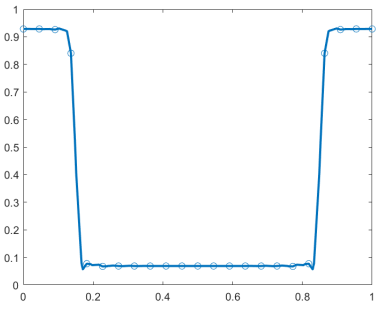
Figure 18: The initial concentration for the local refinement experiment. The ring area has a inner radius of 0.3 and an outer radius 0.4, and is centered at the domain center. In the ring region is a random concentration with $\bar{c} = 0.5$ and a perturbation of $[-0.05, 0.05]$. The concentration in the inner circle is set to be 0.08, and in the rest of the domain 0.92.



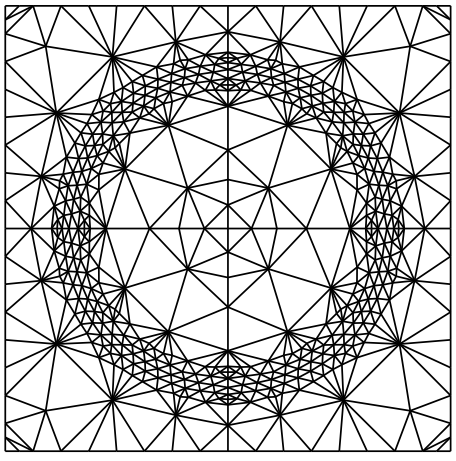
(a) The uniform triangular mesh with 968 macro-elements.



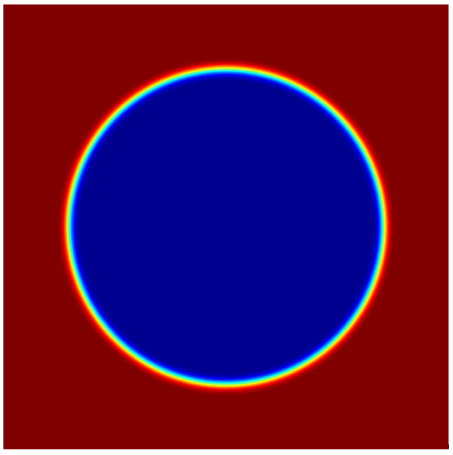
(b) The steady state solution with the uniform mesh.



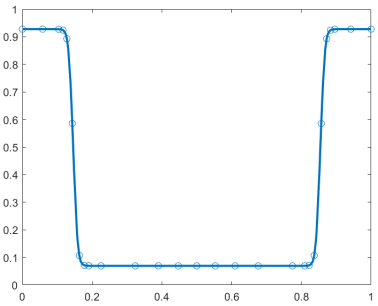
(c) The concentration along $x = 0.5$ for the uniform mesh case. Dots represent element vertex points.



(d) The locally refined triangular mesh with 952 macro-elements.



(e) The steady state solution with the locally refined mesh.



(f) The concentration along $x = 0.5$ for the locally refined case. Dots represent element vertex points.

Figure 19: The local refinement experiment starts the computation with the initial profile shown in Fig. 18. Quintic elements are used in this test. This test shows locally refined meshes can resolve the transient layer in the steady state solution.

tions needed. In [13] a tIGA adaptive refinement algorithm was implemented using Rivara method[37], where the macro-elements with large energy error were sequentially bisected in the process. As a direction of our future tIGA research, we plan to incorporate the said technique, use the residual as our error metric, so that we can adaptively capture the thin transient layers with local refinements during the computation.

6.4 Three-dimensional system evolution

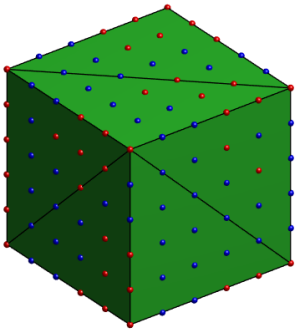
Similar to the two-dimensional case, we carry out this computation in the three-dimensional unit cube domain $(0, 1)^3$ with periodic boundary condition on all directions. The initial concentration $c(\mathbf{x}, t = 0)$ is again set to be a small perturbation around an average concentration $\bar{c}$.

Different from the two-dimensional case, the isoperimetric problem related to the three-dimensional Cahn–Hilliard model in a periodic cube remains an open question[9, 35], with the topology of the steady state solution having several hypothetical candidates, which are lamellar, cylindrical, spherical structures as well as the Lawson and the Schwarz P minimal surfaces[38, 39]. Nevertheless, the discoveries reported in the literature can be useful references for us to understand both the morphing process and the final topology.

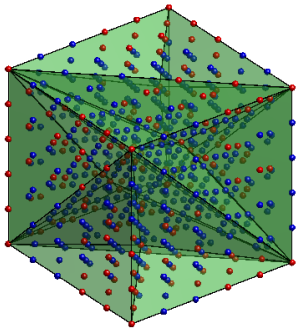
For the two tests in this section, we follow the examples in [3]. The three-dimensional computation uses 6000 quintic tetrahedral macro-elements with Alfeld split in each case, resulting in 24000 micro-elements in total, 66000 free nodes out of 515000 total nodes. The method to generate the three-dimensional C^1-continuous tetrahedral mesh is introduced in Section 4 and also detailed in [17]. The mesh is, similar to the two-dimensional cases, arranged in a tetrahedralized uniform grid style. For a quick look at one "block" of this mesh see Fig. 20. Fig. 20(a) and (b) give an overview of node distribution, with red indicating free nodes and blue dependent nodes. (c) shows the connectivity. Each of these blocks is made of 6 tetrahedral macro-elements, with an example in (d). As discussed, Alfeld split is enforced on macro-elements and one resulting micro-element is shown in (e).

The first example sets $\alpha = 600$ and $\bar{c} = 0.63$. The complete system evolution is shown in Fig. 21. We see in this case, the species separate into two interconnected regions in a short period of time. Unlike in two dimensions, this time no nucleation is observed. The surface has its area decreased as the system evolves. At last, the remaining feature of the surface classifies itself to be a Lawson minimal surface. This result differs from that reported in [3] which turns out to be a cylindrical surface, however is supported by similar test case reported in [9, 35]. We note, because this experiment uses relatively coarse mesh and the numerical research suggests the steady state solution has mesh dependency.

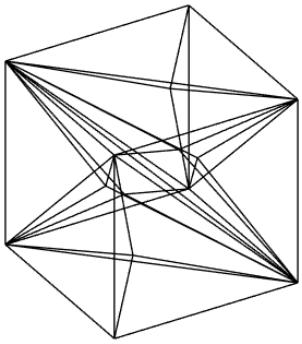
The second example has the configuration of $\alpha = 1800$ and $\bar{c} = 0.75$. Examining the morphology presented in Fig. 22 we can observe clear nucleation, suggesting the significance of initial concentration on the topology of the time-dependent solution.



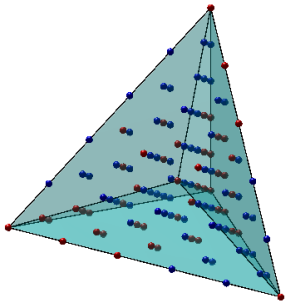
(a) Element surfaces and surface nodes.



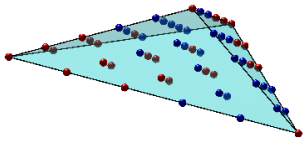
(b) Interior nodes.



(c) Connectivity wireframe.

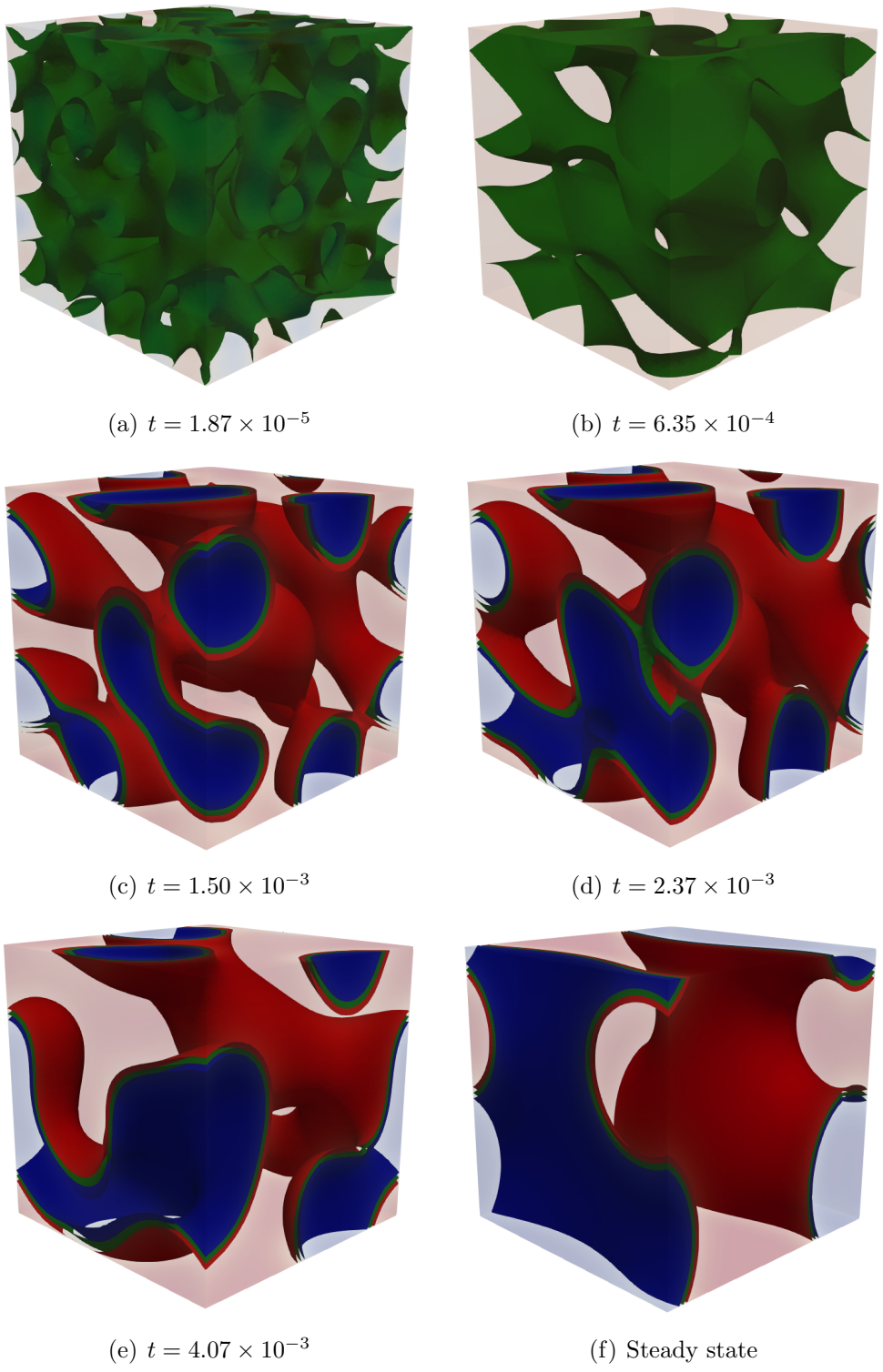


(d) Alfeld-split macro-element and all associated nodes.



(e) Alfeld-split micro-element and all associated nodes.

Figure 20: An example of C^1 quintic tetrahedral mesh with Alfeld split, red nodes represent free nodes and blue nodes represent dependent nodes: (a), (b) and (c) are the overview of one such mesh "block", which can easily used to discretize a unit cube; (d) shows one macro-element within a "block"; (e) shows one micro-element within a macro-element.



(a) $t = 1.87 \times 10^{-5}$

(b) $t = 6.35 \times 10^{-4}$

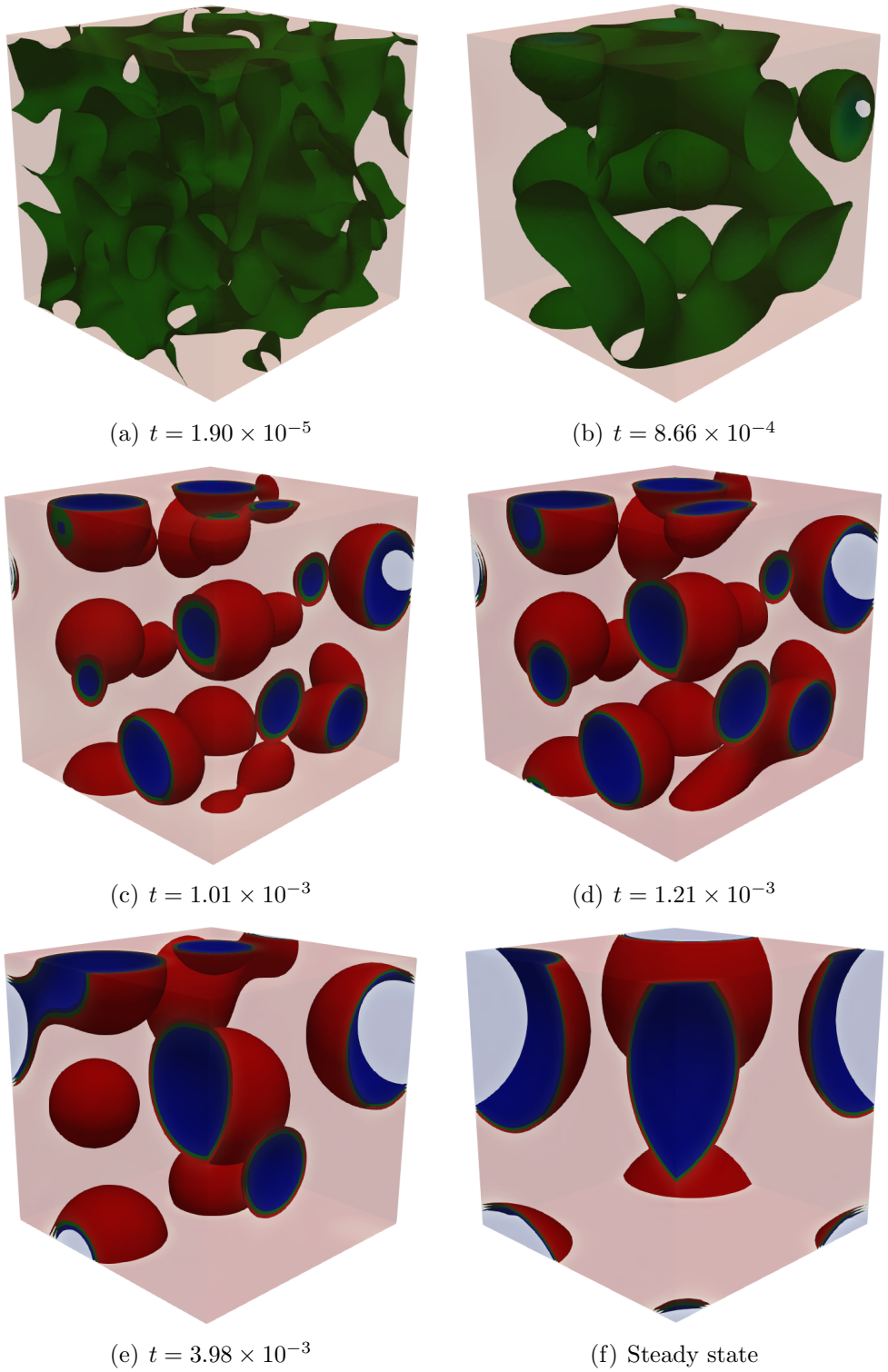
(c) $t = 1.50 \times 10^{-3}$

(d) $t = 2.37 \times 10^{-3}$

(e) $t = 4.07 \times 10^{-3}$

(f) Steady state

Figure 21: The evolution of the concentration from an initial concentration $\bar{c} = 0.63$ and $\alpha = 600$ in a periodic cube.



(a) $t = 1.90 \times 10^{-5}$ (b) $t = 8.66 \times 10^{-4}$

(c) $t = 1.01 \times 10^{-3}$ (d) $t = 1.21 \times 10^{-3}$

(e) $t = 3.98 \times 10^{-3}$ (f) Steady state

Figure 22: The evolution of the concentration from an initial concentration $\bar{c} = 0.75$ and $\alpha = 1800$ in a periodic cube.

7 Conclusion

This paper presents a C^1 triangulation-based isogometric analysis of the Cahn–Hilliard phase-field model. We have obtained C^1 simplicial elements through macro-element techniques. PS split and macro polynomial elements are used to generate quadratic, cubic and quintic C^1 Bézier triangular elements. For 3D, we have used Alfeld-split to create C^1 Bézier tetrahedral mesh. Based on the C^1 simplicial elements, we obtained optimal convergence rates for a model Cahn-Hillard problem with manufactured solution.

This approach has been applied to several phase-field problems, such as the two- and three-dimensional system evolutions starting form an average concentration with a small perturbation and the periodic isoperimetric problem. These examples demonstrate that C^1 triangulation-based isogeometric analysis has several salient features in solving high-order PDEs such as the Cahn-Hillard equation. These features include nodal and DoF advantages over C^0 FEA, local refinement, applicable to complex topology and C^1 smooth elements.

Acknowledgment

This work is supported by ARO grant W911NF-17-1-0020 and NSF grant 1435072.

References

[1] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.

[2] John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics*, 28(258), 1958.

[3] Héctor Gómez, Victor M. Calo, Yuri Bazilevs, and Thomas J.R. Hughes. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4333 – 4352, 2008.

[4] Junseok Kim, Seunggyu Lee, Yongho Choi, Seok-Min Lee, and Darae Jeong. Basic principles and practical applications of the Cahn–Hilliard equation. *Mathematical Problems in Engineering*, 2016, 2016.

[5] John W. Barrett, James F. Blowey, and Harald Garcke. Finite element approximation of the Cahn–Hilliard equation with degenerate mobility. *SIAM Journal on Numerical Analysis*, 37(1):286–318, 1999.

[6] Yinhua Xia, Yan Xu, and Chi-Wang Shu. Local discontinuous Galerkin methods for the Cahn–Hilliard type equations. *Journal of Computational Physics*, 227(1):472 – 491, 2007.

[7] Garth N. Wells, Ellen Kuhl, and Krishna Garikipati. A discontinuous Galerkin method for the Cahn–Hilliard equation. *Journal of Computational Physics*, 218(2):860 – 877, 2006.

[8] Ju Liu, Luca Dede, John A Evans, Michael Borden, and Thomas J.R. Hughes. Isogeometric analysis of the advective Cahn–Hilliard equation: Spinodal decomposition under shear flow. *Journal of Computational Physics*, 242:321350, 06 2013.

[9] S. Kaessmair and P. Steinmann. Comparative computational analysis of the Cahn–Hilliard equation with emphasis on C^1-continuous methods. *Journal of Computational Physics*, 322:783 – 803, 2016.

[10] Liangzhe Zhang, Michael R. Tonks, Derek Gaston, John W. Peterson, David Andrs, Paul C. Millett, and Bulent S. Biner. A quantitative comparison between C^0 and C^1 elements for solving the Cahn–Hilliard equation. *Journal of Computational Physics*, 236:74 – 80, 2013.

[11] Yongjie Zhang, Wenyan Wang, and Thomas J.R. Hughes. Solid t-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering*, 249-252:185 – 197, 2012. Higher Order Finite Element and Isogeometric Methods.

[12] Wenyan Wang, Yongjie Zhang, Lei Liu, and Thomas J.R. Hughes. Trivariate solid t-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design*, 45(2):351 – 360, 2013. Solid and Physical Modeling 2012.

[13] Noah Jaxon and Xiaoping Qian. Isogeometric analysis on triangulations. *Computer-Aided Design*, 46:45–57, 2014.

[14] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Softw.*, 3(4):316–325, December 1977.

[15] Hendrik Speleers. A normalized basis for reduced Clough–Tocher splines. *Computer Aided Geometric Design*, 27(9):700 – 712, 2010.

[16] Songtao Xia, Xilu Wang, and Xiaoping Qian. Continuity and convergence in rational triangular Bézier spline based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 297:292–324, 2015.

[17] Songtao Xia and Xiaoping Qian. Isogeometric analysis with Bézier tetrahedra. *Computer Methods in Applied Mechanics and Engineering*, 316:782 – 816, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.

[18] Cunfu Wang, Songtao Xia, Xilu Wang, and Xiaoping Qian. Isogeometric shape optimization on triangulations. *Computer Methods in Applied Mechanics and Engineering*, 331:585 – 622, 2018.

[19] Mehrdad Zareh and Xiaoping Qian. Kirchhoff–Love shell formulation based on triangular isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 347:853 – 873, 2019.

[20] Ming-Jun Lai and Larry L Schumaker. *Spline functions on triangulations*. Number 110. Cambridge University Press, 2007.

[21] Gerald E Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.

[22] Michael Andreas Matt. *Trivariate Local Lagrange Interpolation and Macro Elements of Arbitrary Smoothness*. Springer Science & Business Media, 2012.

[23] John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. iii. nucleation in a two-component incompressible fluid. *The Journal of Chemical Physics*, 31(3):688–699, 1959.

[24] Kenneth E. Jansen, Christian H. Whiting, and Gregory M. Hulbert. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3):305 – 319, 2000.

[25] Yinnian He, Yunxian Liu, and Tao Tang. On large time-stepping methods for the Cahn–Hilliard equation. *Applied Numerical Mathematics*, 57(5):616 – 628, 2007. Special Issue for the International Conference on Scientific Computing.

[26] Peter Alfeld. Bivariate spline spaces and minimal determining sets. *Journal of Computational and Applied Mathematics*, 119(1):13 – 27, 2000.

[27] Hong Dong. Spaces of bivariate spline functions over triangulation. *Approx. Theory Appl*, 7, 03 1996.

[28] Ming-Jun Lai and Michael A. Matt. A C^r trivariate macro-element based on the Alfeld split of tetrahedra. *Journal of Approximation Theory*, 175:114 – 131, 2013.

[29] Michael J. Borden, Michael A. Scott, John A. Evans, and Thomas J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2010.

[30] Patrick J. Roache. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, 124(1):4–10, November 2001.

[31] Martin Alns, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie Rognes, and Garth Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[32] Y. BAZILEVS, L. BEIRO DA VEIGA, J. A. COTTRELL, T. J. R. HUGHES, and G. SANGALLI. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[33] Markus Kastner, Philipp Metsch, and René de Borst. Isogeometric analysis of the Cahn–Hilliard equation - a convergence study. *Journal of Computational Physics*, 305:360 – 371, 2016.

[34] Stan Cabay. An analysis of the finite element method (Gilbert Strang and George J. Fix). *SIAM Review*, 17(3):577–578, 1975.

[35] Olga Wodo and Baskar Ganapathysubramanian. Computationally efficient solution to the Cahn–Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3d isoperimetric problem. *Journal of Computational Physics*, 230(15):6037 – 6060, 2011.

[36] Garth N. Wells, Ellen Kuhl, and Krishna Garikipati. A discontinuous galerkin method for the cahnhilliard equation. *Journal of Computational Physics*, 218(2):860 – 877, 2006.

[37] Maria-Cecilia Rivara. New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations. 1999.

[38] Antonio Ros. The isoperimetric problem. In *Global Theory of Minimal Surfaces*, pages 175–209. 2001.

[39] Rustum Choksi and Peter Sternberg. Periodic phase separation: the periodic Cahn–Hilliard and isoperimetric problems. *Interfaces and Free Boundaries*, 8:371–392, 2006.