AN EFFICIENT UNCONDITIONALLY STABLE METHOD FOR DIRICHLET PARTITIONS IN ARBITRARY DOMAINS*

DONG WANG†

Abstract. A Dirichlet k-partition of a domain is a collection of k pairwise disjoint open subsets such that the sum of their first Laplace–Dirichlet eigenvalues is minimal. In this paper, we propose a new relaxation of the problem by introducing auxiliary indicator functions of domains and develop a simple and efficient diffusion generated method to compute Dirichlet k-partitions for arbitrary domains. The method only alternates three steps: 1. convolution, 2. thresholding, and 3. projection. The method is simple, easy to implement, insensitive to initial guesses and can be effectively applied to arbitrary domains without any special discretization. At each iteration, the computational complexity is linear in the discretization of the computational domain. Moreover, we theoretically prove the energy decaying property of the method. Experiments are performed to show the accuracy of approximation, efficiency and unconditional stability of the algorithm. We apply the proposed algorithms on both 2- and 3-dimensional flat tori, triangle, square, pentagon, hexagon, disk, three-fold star, five-fold star, cube, ball, and tetrahedron domains to compute Dirichlet k-partitions for different k to show the effectiveness of the proposed method. Compared to previous work with reported computational time, the proposed method achieves hundreds of times acceleration.

Key words. Dirichlet partition, energy stability, thresholding

AMS subject classifications. 49Q10, 49R05,05B45

1. Introduction. For $d \geq 2$, let Ω be either an open bounded domain in $\mathbb{R}^d$ with Lipschitz boundary or a closed, smooth, d-dimensional manifold. For $k \geq 2$ fixed, the Dirichlet k-partition problem for Ω is to choose a k-partition, *i.e.*, k disjoint quasi-open sets $\Omega_1, \Omega_2, \ldots, \Omega_k \subseteq \Omega$, that attains

$$(1.1) \qquad \min_{\Omega = \cup_{\ell \in [k]} \Omega_\ell} \sum_{\ell \in [k]} \lambda_1(\Omega_\ell), \qquad \text{where} \quad \lambda_1(D) := \min_{\substack{u \in H_0^1(D) \\ \|u\|_2 = 1}} E(u).$$

Here, $E(u) = \int_D |\nabla u|^2 \, dx$ is the Dirichlet energy and $\lambda_1(D)$ is the first Dirichlet eigenvalue of the Laplace operator, $-\Delta$, on D with Dirichlet boundary conditions imposed on ∂D.

The existence of optimal partitions in the class of quasi-open sets was proved in [6]. The properties of optimal partitions including the regularity of the partition interfaces and the asymptotic behavior as $k \to \infty$ have been investigated in [7, 17, 5]. The consistency of Dirichlet partitions has been rigorously studied in [21]. Dirichlet partitions have been applied into the study of Bose–Einstein condensates [1, 2, 8] and models for interacting agents [11, 12, 8, 13, 14].

The development of efficient numerical methods for finding such partitions attracts much attention in recent years, especially when the dimension is high or number of partitions is large. Essentially, this is an interface related optimization problem subject to global constraints, numerical considerations usually start with the representation of interfaces. Corresponding numerical methods are mainly developed along

---

†School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China (wangdong@cuhk.edu.cn).

the directions of phase field based approaches [15], level set based approaches [10], and other optimization based approaches [5].

Along the direction of phase field based approaches, following [7], for fixed $\varepsilon > 0$, problem (1.1) can be relaxed to minimizing a relaxed energy,

$$(1.2) \qquad E_\varepsilon(u) = \sum_{\ell \in [k]} \frac{1}{2} \int_\Omega |\nabla u_\ell|^2 \ dx + \frac{1}{4\varepsilon^2} \int_\Omega F_k(u) \ dx.$$

over fields that take values in $\mathbb{R}^k$ where $u = (u_1, u_2, \cdots, u_k)$, $[k]$ denotes the set $\{1, 2, \cdots, k\}$, and

$$F_k(u) = \sum_{i \neq j \in [k]} u_i^2 u_j^2.$$

Then the minimization reads

$$(1.3) \qquad \min_{u \in H^1(\Omega, \mathbb{R}^k)} E_\varepsilon(u)$$

$$\text{s.t.} \ \int_\Omega u_\ell^2 \ dx = 1, \qquad \forall \ell \in [k].$$

Note that the penalty term in the objective functional tries to penalize that the support of each function u_ℓ has no overlap with others. Based on this, in [15], Du and Lin proposed an efficient normalized gradient descent method to approximately find the minimizer of (1.2). The method is initialized with an initial condition $u^0 \in H^1(\Omega, \mathbb{R}^k)$ and alternates the following three steps until convergence. In the first step, the Cauchy problem for the gradient flow of the first term in E_ε, *i.e.*,

$$(1.4) \qquad \partial_t u(x, t) = \Delta u(x, t),$$

is computed until time $\tau > 0$, with initial condition, $u(x, t = 0) = u^0(x)$. Let $\tilde{u}_\ell(x) = u_\ell(x, \tau)$ for $\ell \in [k]$. In the second step, for each $x \in \Omega$, the following system of ordinary differential equations is solved until time τ,

$$(1.5) \qquad \frac{d}{dt} u_\ell = \frac{1}{\varepsilon^2} \left(\sum_{j \neq \ell} u_j^2 \right) u_\ell, \qquad \ell \in [k],$$

with initial condition given by $u_\ell(x, t = 0) = \tilde{u}_\ell(x)$. This is precisely the gradient flow of the second term of the relaxed energy. Numerically, this system is solved using the Gauss-Seidel method. Let $\hat{u}_\ell(x) = u_\ell(x, \tau)$ for $\ell = [k]$. Finally, in the third step, each component of u is normalized to satisfy the $L^2(\Omega)$ norm constraint, *i.e.*,

$$(1.6) \qquad u_\ell = \frac{\hat{u}_\ell}{\|\hat{u}_\ell\|_2}.$$

In this method, the small parameter ε thickens the interface between any two partitions, restricting the mesh size and making the convergence relatively slow. Recently, a scalar auxiliary variable (SAV) approach [25, 26] shows its great advantage on solving systems of gradient flow and a SAV based method for preserving global constraints is proposed in [9] to solve (1.3). However, in such a specific class of problems, the solution of interest is the minimizer instead of the dynamics from an initial guess to the minimizer. The method takes a lot of iterations to converge to the minimizer and seems to be difficult to find regular solutions, especially when k is large.

To accelerate the convergence, Wang and Osting [31] proposed a diffusion generated method to compute Dirichlet partitions approximately. The main novelty in the method is replacing the second step of Du and Lin's method (*i.e.*, (1.5)) by direct projection to make $F_k(u) = 0$. This is based on the observation that any solution satisfies that $\cup_\ell \text{supp}(u_\ell) = \Omega$ by the monotonicity of Dirichlet eigenvalues (and also the relaxed energy). To be more precise, for each x, the projection is simply done by comparing the values among $(u_1, u_2, \cdots, u_k)$, keeping the largest one and projecting other values to 0. This approach dramatically accelerates the speed of convergence from random initial guesses based on the numerical observations. It can simply and quickly find Dirichlet k-partitions in 4-dimensional space with different k even on a laptop. However, it alternates a diffusion step, a projection step, and a normalization step. No theoretical guarantee could be provided on the convergence or energy decaying properties. In addition, the results presented in [31] are limited to periodic cases or closed surfaces, it is not obvious on how to effectively extend to arbitrary domains with Dirichlet boundary conditions.

Another approach, developed first in [5], is based on a Schrödinger operator relaxation of (1.1) and was further used in [3, 4]. The idea here is to replace the shape optimization problem for a partition $\Omega = \cup_\ell \Omega_\ell$ in (1.1) with the following relaxed optimization problem for a collection of functions $\{\varphi_\ell\}_{\ell \in [k]}$:

$$(1.7) \qquad \min_{\{\varphi_\ell\} \in K} \sum_{\ell \in [k]} \lambda_1^\alpha(\varphi_\ell).$$

Here, the constraint set is given by

$$(1.8) \qquad K = \left\{ \{\varphi_\ell\}_{\ell \in [k]} : \sum_\ell \varphi_\ell(x) = 1 \ \text{ and } \ \varphi_\ell(x) \in [0,1] \ \ a.e. \ x \in \Omega \right\}.$$

For $\alpha > 0$, $\lambda_1^\alpha(\phi)$ is defined as the first eigenvalue for the Schrödinger operator $-\Delta + \frac{1}{\alpha}(1 - \phi)$ by

$$(1.9) \qquad \lambda_1^\alpha(\phi) := \min_{\substack{u \in H^1(\Omega) \\ \|u\|_2 = 1}} \frac{1}{2} \int_\Omega |\nabla u|^2 + \frac{1}{2\alpha}(1 - \phi)u^2 \ dx.$$

It was shown that if $\phi = \chi_D$ where χ_D denotes the indicator function of domain D, then $\lambda_1^\alpha(\phi) \to \lambda_1(D)$ as $\alpha \to 0$ [5] (see also in [24]). Furthermore, the objective functional in (1.7) is concave with respect to φ, so the minimum in (1.7) is attained at extreme points of K, which are exactly indicator functions of domain, giving partition solutions. One could interpret the second term in (1.9) as a penalty term to penalize the support of u to be the region where $\phi = 1$ if ϕ is an indicator function of a domain.

In this paper, we propose a novel relaxation to the Dirichlet partition problem. Instead of considering the relaxation using the Schrödinger operator, we propose to approximate $\lambda_1(\phi)$ using a small τ as follows

$$\lambda_1^\tau(\phi) = \min_{\substack{v \in L^2(\Omega) \\ \|v\|_2 = 1}} \frac{1}{\tau} - \frac{1}{\tau} \int_\Omega \phi |e^{\frac{\tau}{2}\Delta} v|^2 \ dx$$

where $e^{\frac{\tau}{2}\Delta} v$ denotes the solution at $t = \tau/2$ of the following free space heat diffusion equation:

$$(1.10) \qquad \begin{cases} \partial_t u = \Delta u, \\ u(x, t = 0) = v(x) \end{cases}$$

which can also explicitly be written by

$$u(x, \tau/2) = G_{\tau/2} * v, \quad G_\tau(x) = \frac{1}{(4\pi\tau)^{d/2}} \exp(-\frac{|x|^2}{4\tau}).$$

Based on the new relaxed problem, we derive a novel and simple iterative method for finding the approximate solution. The method only alternates three steps: 1. convolution, 2. thresholding, and 3. projection. Because of the use of auxiliary functions, it can be applied into Dirichlet partition problems in arbitrary domains. The convolution is between a free-space heat kernel and a function with finite support, it can be efficiently computed using the fast Fourier transform (FFT) even for the cases of arbitrary domains by simply extending to a relatively larger square domain. Furthermore, we rigorously prove the unconditional stability of the proposed method. In other words, each iteration in the proposed algorithms enjoys the energy decaying property.

The paper is organized as follows. In Section 2, we describe the new relaxation of the Dirichlet k-partition problem and some basic properties of the relaxed objective functional. We derive the algorithm in Section 3 and describe the detail of implementation in Section 4. We show the performance of the proposed algorithm via extensive numerical experiments in Section 5 and draw some conclusions and future discussions in Section 6.

2. New relaxation of the problem. Let $d \geq 2$ and $\Omega \in \mathbb{R}^d$ be a bounded open connected set. For every open (or quasi-open) subset $A \subset \Omega$ we denote by λ_1^A the first Dirichlet eigenvalue of the Laplace operator:

$$(2.1) \qquad \begin{cases} -\Delta u = \lambda_1^A u \ \ \text{in} \ \ A, \\ u = 0 \ \ \text{on} \ \ \partial A. \end{cases}$$

This can be understood in a weak sense to find $u \in H_0^1(A)$ such that,

$$(2.2) \qquad \forall v \in H_0^1(A), \quad \int_A \nabla u \cdot \nabla v \ dx = \lambda_1^A \int_A uv \ dx.$$

The eigenvalue can now be given by the minimum principle:

$$(2.3) \qquad \lambda_1^A = \min_{u \in H_0^1(A)} \frac{\int_A |\nabla u|^2 \ dx}{\int_A |u|^2 \ dx}$$

Using the fact

$$\int_A |\nabla u|^2 \ dx = \int_A -u\Delta u \ dx,$$

a simple expansion

$$(2.4) \qquad \begin{cases} e^{\frac{\tau}{2}\Delta}u = u + \frac{\tau}{2}\Delta u + o(\tau) & \tau \to 0, \\ |e^{\frac{\tau}{2}\Delta}u|^2 = |u|^2 + \tau u\Delta u + o(\tau) & \tau \to 0, \end{cases}$$

and $\int_A |u|^2 \ dx = 1$, one can obtain an approximation to $\lambda_1^{\tau,A}$ through

$$(2.5) \quad \lambda_1^{\tau,A} = \min_{\substack{u \in H_0^1(A) \\ \|u\|_2 = 1}} \int_A |\nabla u|^2 \ dx = \min_{\substack{u \in H_0^1(A) \\ \|u\|_2 = 1}} \frac{1}{\tau} - \frac{1}{\tau}\int_A |e^{\frac{\tau}{2}\Delta}u|^2 \ dx + o(1) \quad \tau \to 0.$$

Let $\phi : \Omega \to \{0, 1\}$ be a bounded variation function, we further consider a relaxed problem:

$$(2.6) \qquad \lambda_1^\tau(\phi) = \inf_{\substack{v \in L^2(\Omega) \\ \|v\|_2 = 1}} \frac{1}{\tau} - \frac{1}{\tau} \int_\Omega \phi |e^{\frac{\tau}{2}\Delta}v|^2 \, dx.$$

Remark 2.1. 1. In the relaxation, we also relax the space for v from $H_0^1(\Omega)$ to $L^2(\Omega)$.
2. Furthermore, we note that when $\phi = \chi_A$, the direct relaxation using ϕ in the form

$$\inf_{\substack{v \in H_0^1(\Omega) \\ \|v\|_2 = 1}} \int_\Omega \phi |\nabla v|^2 \, dx \quad \text{or} \quad \inf_{\substack{v \in L^2(\Omega) \\ \|v\|_2 = 1}} \int_\Omega \phi |\nabla v|^2 \, dx$$

would obviously attain the global minimum value 0 at many choices of $v \in H_0^1(\Omega)$ satisfying $\|v\|_2 = 1$. For example, one can simply take $\tilde{v} \in H_0^1(\Omega)$ satisfying $\|\tilde{v}\|_2 = 1$ and $\text{supp}(\tilde{v}) \in \Omega \setminus \bar{A}$ where $\bar{A}$ is the closure of A.

Denote

$$B = \{u = (u_1, u_2, \cdots, u_k) | u \in L^2(\Omega, \mathbb{R}^k), \|u_\ell\|_2 = 1\},$$

we propose a new relaxation to problem (1.1) by

$$(2.7) \qquad \inf_{\varphi \in K} \inf_{u \in B} E^\tau(\varphi, u) := \frac{k}{\tau} - \sum_{\ell \in [k]} \frac{1}{\tau} \int_\Omega \varphi_\ell |e^{\frac{\tau}{2}\Delta}u_\ell|^2 \, dx.$$

We first list some properties of $E^\tau(\varphi, u)$ in the following lemma.

LEMMA 2.2. *Assume $\tau > 0$, then the following properties hold for the functional $E^\tau(\varphi, u)$ defined in (2.6).*
  (i) $E^\tau(\varphi, u)$ is nonnegative for any $(\varphi, u) \in K \times B$.
  (ii) Given φ, $E^\tau(\varphi, u)$ is continuous with respect to u on $L^2(\Omega; \mathbb{R}^k)$.
  (iii) $E^\tau(\varphi, u)$ is concave with respect to u on $L^2(\Omega; \mathbb{R}^k)$.
  (iv) The Fréchet derivative of $E^\tau(\varphi, u)$ with respect to $u_\ell \in L^2(\Omega; \mathbb{R})$ is

$$\frac{\delta E^\tau(\varphi, u)}{\delta u_\ell} = -\frac{2}{\tau} e^{\frac{\tau}{2}\Delta}(\varphi_\ell e^{\frac{\tau}{2}\Delta}u_\ell).$$

Proof. (i) For any $(\varphi, u) \in K \times B$ and $\ell \in [k]$, $\varphi_\ell(x) \in [0, 1]$ and $\|u_\ell\| = 1$, we then have

$$-\frac{1}{\tau} \int_\Omega \varphi_\ell |e^{\frac{\tau}{2}\Delta}u_\ell|^2 \, dx \geq -\frac{1}{\tau} \int_\Omega |e^{\frac{\tau}{2}\Delta}u_\ell|^2 \, dx$$

$$\geq -\frac{1}{\tau} \|e^{\frac{\tau}{2}\Delta}\|^2 \|u_\ell\|_2^2 \geq -\frac{1}{\tau} \|u_\ell\|_2^2 = -\frac{1}{\tau}$$

and thus we have $E^\tau(\varphi, u) \geq 0$.
(ii) Let $u, v \in L^2(\Omega, \mathbb{R}^k)$, direct calculation using the fact that $\|e^{\frac{\tau}{2}\Delta}u\|_2 \leq \|u\|_2$

yields

$$|E^\tau(\varphi, u) - E^\tau(\varphi, v)| = \sum_{\ell \in [k]} \frac{1}{\tau} \int_\Omega \varphi_\ell \left||e^{\frac{\tau}{2}\Delta} v_\ell|^2 - |e^{\frac{\tau}{2}\Delta} u_\ell|^2\right| \, dx$$

$$= \sum_{\ell \in [k]} \frac{1}{\tau} \int_\Omega \varphi_\ell \left|\left(e^{\frac{\tau}{2}\Delta}(u_\ell + v_\ell)\right)\left(e^{\frac{\tau}{2}\Delta}(u_\ell - v_\ell)\right)\right| \, dx$$

$$\leq \frac{1}{\tau} \sum_{\ell \in [k]} \|u_\ell + v_\ell\|_2 \|u_\ell - v_\ell\|_2$$

$$\leq \frac{2\sqrt{k}}{\tau} \|u - v\|_2,$$

implying the continuity in the L^2 topology.

(iii) This can be proved by a direct computation.

(iv) $\forall v \in L^2(\Omega, \mathbb{R}^k)$ with $v_i = 0$ $(i \neq \ell)$, direct computation yields

$$\left\langle \frac{\delta E^\tau(\varphi, u)}{\delta u_\ell}, v_\ell \right\rangle = \lim_{\epsilon \to 0} \frac{E^\tau(\varphi, u + \epsilon v) - E^\tau(\varphi, u)}{\epsilon}$$

$$= -\frac{2}{\tau} \int_\Omega (\varphi_\ell e^{\frac{\tau}{2}\Delta} u_\ell) e^{\frac{\tau}{2}\Delta} v_\ell \, dx$$

$$= -\frac{2}{\tau} \int_\Omega (e^{\frac{\tau}{2}\Delta}(\varphi_\ell e^{\frac{\tau}{2}\Delta} u_\ell)) v_\ell \, dx$$

$$= \left\langle -\frac{2}{\tau} e^{\frac{\tau}{2}\Delta}(\varphi_\ell e^{\frac{\tau}{2}\Delta} u_\ell), v_\ell \right\rangle.$$

Here, the second to the last equality is by the fact that the operator $e^{\frac{\tau}{2}\Delta}$ forms a semi-group, that is,

$$\left\langle e^{\frac{\tau}{2}\Delta} f, g \right\rangle = \left\langle f, e^{\frac{\tau}{2}\Delta} g \right\rangle. \qquad \qquad \square$$

In the follows, we first discuss the existence of the solution of the relaxed problem (2.7) for given φ.

THEOREM 2.3 (Existence of u). *For a given $\varphi \in K$ and $\tau > 0$, the problem* (2.7) *admits at least one solution $u \in B$.*

Proof. Denote

$$\tilde{B} = \{\tilde{u} = (\tilde{u}_1, \tilde{u}_2, \cdots, \tilde{u}_k) | u = (u_1, u_2, \cdots, u_k) \in L^2(\Omega, \mathbb{R}^k),$$

$$\tilde{u}_\ell = \frac{e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u_\ell)}{\|e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u_\ell)\|_2}, \|u_\ell\|_2 \leq 1\}$$

and $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$ as the closure of $\tilde{B}$ in $H^1(\Omega, \mathbb{R}^k)$. Here, we use the fact that $\tilde{u}_\ell$ is a smooth function and its H^1 norm is bounded. Because $H^1(\Omega, \mathbb{R}^k)$ is compactly embedded in $L^2(\Omega, \mathbb{R}^k)$ and $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$ is a closed and bounded subset in $H^1(\Omega, \mathbb{R}^k)$, $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$ is then compact in $L^2(\Omega, \mathbb{R}^k)$. In addition, because $E^\tau(\varphi, u)$ is continuous on $L^2(\Omega, \mathbb{R}^k)$, there exists at least one $u^\star \in cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$ such that

$$E^\tau(\varphi, u^\star) = \inf_{u \in cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}} E^\tau(\varphi, u).$$

Furthermore, it is straightforward to see $\|u^\star\|_2 = 1$ because the concavity of $E^\tau(\varphi, u)$ and convexity of $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$. In other words, the minimum value occurs at the extreme points of the set.

For any $v = (v_1, v_2, \cdots, v_k) \in B$ but not in $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$, denote

$$\hat{v}_\ell = \frac{e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell)}{\|e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell)\|_2},$$

we have

$$(2.8) \qquad -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell), v_\ell \right\rangle \geq -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell), \hat{v}_\ell \right\rangle$$

and thus

$$-\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell), v_\ell \right\rangle \geq -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell), v_\ell \right\rangle - 2\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}v_\ell), \hat{v}_\ell - v_\ell \right\rangle$$
$$(2.9) \qquad\qquad \geq -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}\hat{v}_\ell), \hat{v}_\ell \right\rangle.$$

The last inequality comes from the fact that the graph of the linearization of a concave functional locates above its graph. We then have $E^\tau(\varphi, v) \geq E^\tau(\varphi, \hat{v})$. This implies that the minimum value can always be attained in $cl(\tilde{B})_{H^1(\Omega, \mathbb{R}^k)}$. $\qquad\square$

The existence of φ can be argued as follows.

THEOREM 2.4 (Existence of φ). *Problem* (2.7) *admits at least one solution* $\varphi \in K$.

Proof. From Lemma 2.2(i), we have

$$\inf_{\varphi \in K} E^\tau(\varphi, u) \in [0, \frac{1}{\tau}].$$

Let $\{\varphi^n\}_{n=1}^\infty \in K$ be a minimizing sequence, from the weak$^*L^\infty(\Omega)$ sequential compactness of K, we have that there exists a subsequence, which we continue to denote by $\{\varphi^n\}_{n=1}^\infty$ and $\varphi^\star \in K$, such that $\varphi^n \overset{w^*-L^\infty}{\rightharpoonup} \varphi^\star$. Then, because of the fact that $|e^{\frac{\tau}{2}\Delta}u_\ell|^2 \in L^1(\Omega, \mathbb{R})$, we have

$$\sum_{\ell \in [k]} \int_\Omega \varphi_\ell^n |e^{\frac{\tau}{2}\Delta}u_\ell|^2 \, dx \to \sum_{\ell \in [k]} \int_\Omega \varphi_\ell^\star |e^{\frac{\tau}{2}\Delta}u_\ell|^2 \, dx$$

implying the infimum attains at $\varphi^\star$. $\qquad\square$

Furthermore, because of the form of the objective functional, one could arrive that at least one solution of φ gives partition functions. In other words, there exists at least one solution of φ whose entries are indicator functions of domains.

THEOREM 2.5. *Denote*

$$\tilde{K} = \{\varphi = (\varphi_1, \varphi_2, \cdots, \varphi_k) | \varphi_\ell : \Omega \to \{0, 1\} \text{ measurable and } \sum_{\ell \in [k]} \varphi_\ell = 1 \text{ a.e. } \Omega\}.$$

(2.7) *admits at least one solution in* $\tilde{K}$.

Proof. Assume φ is an optimal solution and not in $\tilde{K}$. We assume there exists an $\epsilon > 0$, a measurable set $A \subset \Omega$, and $i \neq j \in [k]$, such that $0 < |A| < |\Omega|$ and

$$\varphi_i(x), \varphi_j(x) \in (\epsilon, 1 - \epsilon) \quad \forall x \in A.$$

Considering an arbitrary $\tilde{A} \subset A$ with a positive measure,

$$\psi_m(x, t) = \varphi_m(x) + t(\delta_{m,i} - \delta_{m,j})\chi_{\tilde{A}}(x)$$

for $m = 1, 2, \cdots, k$. Then we have

$$\sum_m \psi_m(x, t) = 1 \ \text{ and } \ \psi_m(x, t) \geq 0$$

for $t \in (-\epsilon, \epsilon)$ so that $\psi_m(\cdot, t) \in K$. Then, we have

$$\frac{dE^\tau(\varphi, u)}{dt} = \frac{1}{\tau} \int_\Omega \chi_{\tilde{A}}(|e^{\frac{\tau}{2}\Delta}u_j|^2 - |e^{\frac{\tau}{2}\Delta}u_i|^2) \ dx.$$

Because φ is an optimal solution, the first order necessary condition gives that

$$\int_\Omega \chi_{\tilde{A}}(|e^{\frac{\tau}{2}\Delta}u_j|^2 - |e^{\frac{\tau}{2}\Delta}u_i|^2) \ dx = 0, \ \forall \tilde{A} \subset A \text{ with a positive measure}$$

implying that

$$|e^{\frac{\tau}{2}\Delta}u_j(x)|^2 = |e^{\frac{\tau}{2}\Delta}u_j(x)|^2, \ a.e. \text{ in } A.$$

This immediately leads us that $\varphi_i(x) = 1 \ \forall x \in A$ or $\varphi_j(x) = 1 \ \forall x \in A$ can give the same value of $E^\tau(\varphi, u)$.

If there exists some $\tilde{A} \subset A$ with positive measure that

$$\int_\Omega \chi_{\tilde{A}}(|e^{\frac{\tau}{2}\Delta}u_j|^2 - |e^{\frac{\tau}{2}\Delta}u_i|^2) \ dx \neq 0.$$

It then quickly implies that $\psi_m(x, 0)$ can not be an optimal solution which contradicts with the assumption.   □

Now, we focus on the situation on when $\varphi \in \tilde{K}$. For any entry in φ denoted by ϕ, when $\phi = \chi_A$ is the indicator function of a set $A \subset \Omega$, one can intuitively treat $\lambda_1^\tau(\phi)$ as an approximation to $\lambda_1(A)$. The following two lemmas show that as τ becomes small the minimizer u corresponding to $\lambda_1^\tau(\phi)$ becomes strongly localized on A and the approximation is exact in the limit that $\tau \to 0$.

LEMMA 2.6. *Given* $\phi = \chi_A$ *where* A *is a measurable set with positive measure and finite perimeter. Define*

$$u^\tau := \arg \min_{\substack{v \in L^2(\Omega, \mathbb{R}) \\ \|v\|_2 = 1}} - \int_\Omega \phi|e^{\frac{\tau}{2}\Delta}v|^2 \ dx.$$

Then, $u^\tau \geq 0, \ \forall x \in \Omega$ *(or* $u^\tau \leq 0, \ \forall x \in \Omega$*) and* $\lim_{\tau \to 0} \int_\Omega (1 - \phi)u^\tau \ dx = 0$.

Proof. Without loss of generality, we prove the non-negativity of u^τ. If not, we write $u^\tau = u_+^\tau - u_-^\tau$ where

$$u_+^\tau(x) = \max\{0, u^\tau(x)\}, \ \ u_-^\tau = u_+^\tau - u^\tau.$$

Writing $\tilde{u}^\tau = u_+^\tau + u_-^\tau$, we have $\|\tilde{u}^\tau\|_2 = 1$, $e^{\frac{\tau}{2}\Delta}u_+^\tau \geq 0$, and $e^{\frac{\tau}{2}\Delta}u_-^\tau \geq 0$. Thus,

$$|e^{\frac{\tau}{2}\Delta}u^\tau(x)| \leq |e^{\frac{\tau}{2}\Delta}\tilde{u}^\tau(x)|$$

and

$$-\int_\Omega \phi |e^{\frac{\tau}{2}\Delta} u^\tau|^2 \, dx \geq -\int_\Omega \phi |e^{\frac{\tau}{2}\Delta} \tilde{u}^\tau|^2 \, dx.$$

This implies the positivity of u^τ.

We then prove $\lim_{\tau \to 0} \int_\Omega (1-\phi) u^\tau \, dx = 0$.

Because $\|u^\tau\|_2 = 1$, we have

$$\|\phi e^{\frac{\tau}{2}\Delta} u^\tau\|_2 \leq 1 \quad \text{and} \quad \|e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau)\|_2 \leq 1$$

using the fact that $\|e^{\frac{\tau}{2}\Delta}\| \leq 1$.

Then, similar to (2.8) and (2.9), we have

$$-\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau), u^\tau \right\rangle \geq -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} \hat{u}^\tau), \hat{u}^\tau \right\rangle$$

where

$$\hat{u}^\tau = \frac{e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau)}{\|e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau)\|_2}.$$

Because u^τ is the optimal solution, the following equality holds

$$-\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau), u^\tau \right\rangle - 2\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau), \hat{u}^\tau - u^\tau \right\rangle = -\left\langle e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} \hat{u}^\tau), \hat{u}^\tau \right\rangle,$$

which gives

$$\left\langle \phi e^{\frac{\tau}{2}\Delta}(\hat{u}^\tau - u^\tau), e^{\frac{\tau}{2}\Delta}(\hat{u}^\tau - u^\tau) \right\rangle = 0.$$

That is,

$$e^{\frac{\tau}{2}\Delta} \hat{u}^\tau = e^{\frac{\tau}{2}\Delta} u^\tau \quad a.e. \text{ in } A$$

and

$$e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau) = e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} \hat{u}^\tau) = \nu \hat{u}^\tau \quad \text{in } \Omega$$

where

$$\nu = \|e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta} u^\tau)\|_2.$$

In [20], when the perimeter or surface area of the set A is finite and $\psi : \mathbb{R}^d \to \mathbb{R}$ is continuous with compact support, it was rigorously proved that

$$\lim_{\tau \to 0} \sqrt{\frac{2\pi}{\tau}} \int_{\mathbb{R}^d} (1-\phi) e^{\frac{\tau}{2}\Delta}(\phi\psi) \, dx = \int_\Gamma \psi \, d\Gamma$$

where Γ is the boundary of the set A. It follows that

$$\lim_{\tau \to 0} \int_\Omega (1-\phi) e^{\frac{\tau}{2}\Delta}(\phi\hat{\psi}) \, dx = \lim_{\tau \to 0} \int_\Omega (1-\phi) \hat{u}^\tau \, dx = 0$$

by denoting $\hat{\psi} = e^{\frac{\tau}{2}\Delta} \hat{u}^\tau$ and thus

$$\lim_{\tau \to 0} \int_\Omega (1-\phi) u^\tau \, dx = 0. \qquad \square$$

LEMMA 2.7. *Given $A \subset \Omega$ an open set with positive measure, $\lambda_1^\tau(\chi_A) \leq \lambda_1(A)$ and $\lim_{\tau \to 0} \lambda_1^\tau(\chi_A) = \lambda_1(A)$.*

Proof. We first prove the boundedness of $\lambda_1^\tau(\chi_A)$. As shown in the proof of Lemma 2.6, $\lambda_1^\tau(\chi_A)$ can be written by

$$\lambda_1^\tau(\chi_A) = \min_\gamma \frac{1-\gamma}{\tau}$$
$$\text{s.t. } e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}u) = \gamma u \text{ in } \Omega,$$
$$\|u\|_2 = 1.$$

Denote by $\tilde{u}$ the unit eigenfunction corresponding to the first eigenvalue $\lambda_1(A)$ of $-\Delta$ in A with Dirichlet boundary condition and extended by 0 in $\Omega \setminus A$;

$$-\Delta\tilde{u} = \lambda_1(A)\tilde{u} \text{ in } A,$$
$$\tilde{u} = 0 \text{ in } \Omega \setminus A.$$

Direct calculation yields $e^{\frac{\tau}{2}\Delta}\tilde{u} = e^{-\lambda_1(A)\tau/2}\tilde{u}$ which is still supported in A. Hence,

$$e^{\frac{\tau}{2}\Delta}(\phi e^{\frac{\tau}{2}\Delta}\tilde{u}) = e^{\frac{\tau}{2}\Delta}(e^{-\lambda_1(A)\tau/2}\tilde{u}) = e^{-\lambda_1(A)\tau}\tilde{u}.$$

That is, $\tilde{u}$ satisfies the constraint with $\gamma = e^{-\lambda_1(A)\tau}$. It implies that

$$\lambda_1^\tau(\chi_A) \leq \frac{1 - e^{-\lambda_1(A)\tau}}{\tau} \leq \lambda_1(A).$$

In addition, straightforward calculation yields,

$$\frac{d\lambda_1^\tau(\phi)}{d\tau} = -\frac{1}{\tau^2} + \frac{1}{\tau^2}\langle\phi e^{\frac{\tau}{2}\Delta}u, e^{\frac{\tau}{2}\Delta}u\rangle - \frac{1}{\tau}\langle\phi e^{\frac{\tau}{2}\Delta}u, e^{\frac{\tau}{2}\Delta}\Delta u\rangle$$
$$\leq -\frac{1}{\tau^2} + \frac{1}{\tau^2}\langle\phi e^{\frac{\tau}{2}\Delta}u, e^{\frac{\tau}{2}\Delta}(I - \tau\Delta)u\rangle$$
$$\leq -\frac{1}{\tau^2} + \frac{1}{\tau^2}\langle e^{\frac{\tau}{2}\Delta}u, e^{\frac{\tau}{2}\Delta}(I - \tau\Delta)u\rangle$$
$$= -\frac{1}{\tau^2} + \frac{1}{\tau^2}\langle u, e^{\tau\Delta}(I - \tau\Delta)u\rangle \leq -\frac{1}{\tau^2} + \frac{1}{\tau^2}\|e^{\tau\Delta}(I - \tau\Delta)\| \leq 0.$$

It is easy to check that the equality holds only when $\text{supp}(e^{\frac{\tau}{2}\Delta}u) \subset A$, $e^{\frac{\tau}{2}\Delta}(I - \tau\Delta)u \subset A$ and u is a constant function, which can not happen. Hence $\frac{d\lambda_1^\tau(\phi)}{d\tau} < 0$. Consequently, $\lim_{\tau\to 0}\lambda_1^\tau(\chi_A)$ exists and $\lim_{\tau\to 0}\lambda_1^\tau(\chi_A) \leq \lambda_1(A)$.

For the reverse inequality, for the sequence

$$u^\tau = \arg\min_{\substack{v\in L^2(\Omega,\mathbb{R}) \\ \|v\|_2=1}} \frac{1}{\tau} - \frac{1}{\tau}\int_\Omega \phi|e^{\frac{\tau}{2}\Delta}v|^2 \, dx,$$

Lemma 2.6 implies that there exists a $\tilde{u}$ such that $u^\tau \to \tilde{u}$ after possibly passing to a subsequence and $\int_\Omega(1-\phi)\tilde{u} \, dx = 0$. We then observe that $\tilde{u}$ is in the admissible set of the following minimization problem

$$\min_{\substack{v\in L^2(\Omega,\mathbb{R}) \\ \|v\|_2=1 \\ \int_\Omega(1-\phi)vdx=0}} \frac{1}{\tau} - \frac{1}{\tau}\int_\Omega |e^{\frac{\tau}{2}\Delta}v|^2 \, dx$$

which indicates that

$$\lim_{\tau\to 0}\lambda_1^\tau(\chi_A) \geq \lim_{\tau\to 0} \frac{1 - e^{-\lambda_1(A)\tau}}{\tau} = \lambda_1(A). \qquad \square$$
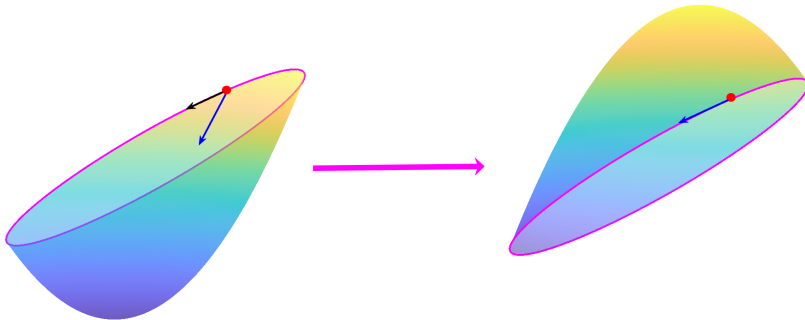
FIG. 1. *A diagram for the main idea on approximating a convex energy using a concave energy by keeping values on the constraint set. See Section 2.*

In Figure 1, we give a simple demonstration of the main idea on approximating a convex energy using a concave energy by keeping values on the constraint set. Considering the constraint set being a circle, for a convex energy as shown in the left, one could calculate the gradient direction which makes the iteration moves inside the convex hull of the constraint set, then an artificial projection to the circle is necessary which may make the energy either increasing or decreasing slowly. However, the optimization is essentially only on the boundary. If one can find a relaxation to a concave approximation as shown in the right, the problem can immediately be relaxed to minimizing a concave functional on a convex set, this can be done efficiently by linear sequential programming owing to the fact that the graph of a concave functional always locates under its linearization. Here, we assume the constraint set has a very important feature: the constraint set is exactly the extreme set of its convex hull. Many algorithms can be connected to this idea, for example, the threshold dynamics method [16] whose constraint set is $\{0,1\}^n$ which are the extreme points of its convex hull $[0,1]^n$. It has been applied into many interface related problems such as wetting dynamics [34, 35], image segmentation [19, 30, 33], foam bubbles [28] and surface reconstruction [27]. Some other related work on diffusion generated methods for target-valued harmonic maps can be referred to [22, 23, 32] where the constraint set is $n \times n$ orthogonal matrix group $O(n)$.

3. Derivation of the algorithm. In this section, for any fixed k, we focus on the numerical method for the following minimization problem

$$\min_{\substack{\varphi \in K \\ u \in B}} E^\tau(\varphi, u) = \frac{k}{\tau} - \sum_{\ell \in [k]} \frac{1}{\tau} \int_\Omega \varphi_\ell |e^{\frac{\tau}{2}\Delta} u_\ell|^2 \; dx.$$

We simply use an alternating direction method of minimization to minimize the energy functional with respect to $u = (u_1, u_2, \cdots, u_k)$ and $\varphi = (\varphi_1, \varphi_2, \cdots, \varphi_k)$. To be specific, start with an initial guess u^0, we compute the sequence

$$\varphi^0, u^1, \varphi^1, u^2, \varphi^2, \cdots, \varphi^n, u^n, \cdots$$

by

(3.1) $$\varphi^n = \min_{\varphi \in K} E^\tau(\varphi, u^n)$$

(3.2) $$u^{n+1} = \min_{u \in B} E^\tau(\varphi^n, u)$$

To solve (3.1), it's easy to see that the problem is linear in φ_ℓ and it can be solved in a pointwise manner by a simple comparison among k values at any $x \in \Omega$. That is, for each x,

$$(3.3) \qquad \varphi_\ell^{n+1}(x) = \begin{cases} 1 & \text{if } \ell = \min\{\arg\max_{i \in [k]} |e^{\frac{\tau}{2}\Delta} u_\ell^n(x)|^2\}, \\ 0 & \text{otherwise.} \end{cases}$$

We then quickly have the following lemma by a direct calculation.

LEMMA 3.1. *The φ^{n+1} computed using (3.3) satisfies*

$$E_k^\tau(\varphi^{n+1}, u^n) \leq E_k^\tau(\varphi, u^n), \quad \forall \varphi \in K.$$

To solve (3.2), we note that the functional is quadratic and strictly concave with respect to u_ℓ. Furthermore, we note that u_ℓ are individual to each other, hence (3.2) can be further relaxed to update u_ℓ independently,

$$(3.4) \qquad u_\ell^{n+1} = \min_{\substack{u_\ell \in L^2(\Omega,\mathbb{R}) \\ \|u_\ell\|_2 \leq 1}} \frac{1}{\tau} - \int_\Omega \frac{1}{\tau} \varphi_\ell^n |e^{\frac{\tau}{2}\Delta} u_\ell|^2 \ dx.$$

The following lemma shows the equivalence between problem (3.2) and problem (3.4).

LEMMA 3.2.

$$\min_{u \in B} E_k^\tau(\varphi^n, u) = \min_{\substack{u \in L^2(\Omega,\mathbb{R}^k) \\ \|u_\ell\|_2 \leq 1}} E_k^\tau(\varphi^n, u).$$

Proof. If $c = \|u_\ell\|_2 < 1$ for some $\ell \in [k]$, write $\tilde{u} = (u_1, u_2, \cdots, u_\ell/c, \cdots, u_k)$, we have

$$E_k^\tau(\varphi^n, \tilde{u}) - E_k^\tau(\varphi^n, u) = -\frac{1 - c^2}{c^2} \frac{1}{\tau} \int_\Omega \varphi_\ell |e^{\frac{\tau}{2}\Delta} u_\ell|^2 \ dx \leq 0$$

where the equality holds only when

$$\int_\Omega \varphi_\ell |e^{\frac{\tau}{2}\Delta} u_\ell|^2 \ dx = 0,$$

which is impossible for a minimizer. Hence, the minimizer for

$$\min_{\substack{u \in L^2(\Omega,\mathbb{R}^k) \\ \|u_\ell\|_2 \leq 1}} E_k^\tau(\varphi^n, u)$$

can be attained in B.   □

Then, problem (3.4) can be solved by the sequential linear programming approach. That is, we consider

$$(3.5) \qquad u_\ell^{n+1} = \arg \min_{\substack{u \in L^2(\Omega,\mathbb{R}^k) \\ \|u_\ell\|_2 \leq 1}} L_\tau^{u_\ell^n}(u)$$

where

$$L_\tau^{u_\ell^n}(u) = E_k^\tau(\varphi^n, u^n) + \sum_{\ell \in [k]} \left\langle u_\ell - u_\ell^n, -\frac{2}{\tau} e^{\frac{\tau}{2}\Delta}(\varphi_\ell^n e^{\frac{\tau}{2}\Delta} u_\ell^n) \right\rangle.$$

The following lemma shows that the minimization problem (3.5) can then be done by a simple projection step.

LEMMA 3.3. *The minimum value of problem* (3.5) *attains at*

$$u_\ell^{n+1} = \frac{e^{\frac{\tau}{2}\Delta}(\varphi_\ell^n e^{\frac{\tau}{2}\Delta}u_\ell^n)}{\|e^{\frac{\tau}{2}\Delta}(\varphi_\ell^n e^{\frac{\tau}{2}\Delta}u_\ell^n)\|_2}.$$

Proof. By rewriting (3.5) and dropping constant terms with respect to u, problem (3.5) is equivalent to

$$u_\ell^{n+1} = \arg \min_{\substack{u_\ell \in L^2(\Omega,\mathbb{R}^k) \\ \|u_\ell\|_2 \leq 1}} \left\langle u_\ell, -\frac{2}{\tau} e^{\frac{\tau}{2}\Delta}(\varphi_\ell^n e^{\frac{\tau}{2}\Delta}u_\ell^n) \right\rangle.$$

This is a direct consequence of the constraint $\|u_\ell\|_2 = 1$ and a simple projection. □

When updating u, one can simply iterate one step to find a solution giving a smaller value in problem (3.4) or iterate to a stationary solution of u for fixed φ^n before updating φ^{n+1}. These are summarized into the following two algorithms (*i.e.*, Algorithm 3.1 and Algorithm 3.2), respectively.

Algorithm 3.1 An iterative method for approximating the solution of problem (2.7).

Input: Let Ω be a given domain, $\tau > 0$, *tol* > 0, and $u^0 \in B$.
Output: φ^n: representing the partition
Set $s = 1$
while $\|\varphi^{s+1} - \varphi^s\|_2 \geq tol$ **do**
    1. Diffusion Step. Compute $u_\ell^* = e^{\frac{\tau}{2}\Delta}u_\ell^s$.
    2. Update φ. Update φ by:

$$\varphi_\ell^s(x) = \begin{cases} 1 & \text{if } \ell = \min\{\arg\max_{i \in [k]} |u_\ell^*(x)|^2\}, \\ 0 & \text{otherwise.} \end{cases}$$

    3. Update u. Update u by:

$$u_\ell^{s+1} = \frac{e^{\frac{\tau}{2}\Delta}(\varphi_\ell^s e^{\frac{\tau}{2}\Delta}u_\ell^s)}{\|e^{\frac{\tau}{2}\Delta}(\varphi_\ell^s e^{\frac{\tau}{2}\Delta}u_\ell^s)\|_2}$$

Set $s = s + 1$
end while

Intuitively, for Algorithms 3.1 and 3.2, a relatively large τ may make the algorithm be insensitive to initial guesses and a relatively small τ could increase the accuracy of the method, especially in u. Based on these observations, we propose adaptive in time algorithms corresponding to Algorithms 3.1 and 3.2 in Algorithms 3.3 and 3.4.

Based on the derivation of above algorithms, we have the following theorem to guarantee that the updating sequence decreases the energy functional (*i.e.*, unconditionally stable) and converges in finite steps.

THEOREM 3.4. *For any given $\tau > 0$, considering the sequence (φ^s, u^s) ($s = 0, 1, 2, \cdots$) generated from Algorithms 3.1 or 3.2, we have*

$$E^\tau(\varphi^{s+1}, u^{s+1}) \leq E^\tau(\varphi^s, u^s).$$

Proof. From Lemma 3.1, we have

$$E^\tau(\varphi^{s+1}, u^s) \leq E^\tau(\varphi^s, u^s).$$

Algorithm 3.2 An iterative method for approximating the solution of problem (2.7).

Input: Let Ω be a given domain, $\tau > 0$, $tol > 0$, and $u^0 \in B$.

Output: φ^n: representing the partition

Set $s = 0$

while $\|\varphi^{s+1} - \varphi^s\|_2 \geq tol$ **do**

   1. Diffusion Step. Compute $u_\ell^* = e^{\frac{\tau}{2}\Delta} u_\ell^s$.

   2. Update φ. Update φ by:

$$\varphi_\ell^s(x) = \begin{cases} 1 & \text{if } \ell = \min\{\arg\max_{i \in [k]} |u_\ell^*(x)|^2\}, \\ 0 & \text{otherwise.} \end{cases}$$

   3. Update u. Set $\hat{u}^{s,0} = u^s$ and $m = 0$,

   while $\|\hat{u}^{s,m+1} - \hat{u}^{s,m}\| \geq tol$ **do**

$$\hat{u}_\ell^{s,m+1} = \frac{e^{\frac{\tau}{2}\Delta}(\varphi_\ell^s e^{\frac{\tau}{2}\Delta}\hat{u}_\ell^{s,m})}{\|e^{\frac{\tau}{2}\Delta}(\varphi_\ell^s e^{\frac{\tau}{2}\Delta}\hat{u}_\ell^{s,m})\|_2}.$$

      Set $m = m + 1$.

   end while

   Set $u^{s+1} = \hat{u}^{s,m}$.

   Set $s = s + 1$.

end while

Algorithm 3.3 An adaptive in time algorithm for Algorithm 3.1.

Input: Let Ω be a given domain, $\tau > 0$, $tol > 0$, and $u^0 \in B$.

Output: φ^n: representing the partition

Set $r = 0$

while $\|\varphi^{r+1} - \varphi^r\|_2 \geq tol$ **do**

   Run Algorithm 3.1 and output φ^r and u^r.

   Set $r = r + 1$ and $\tau \leftarrow \tau/2$.

end while

Algorithm 3.4 An adaptive in time algorithm for Algorithm 3.2.

Input: Let Ω be a given domain, $\tau > 0$, $tol > 0$, and $u^0 \in B$.

Output: φ^n: representing the partition

Set $r = 0$

while $\|\varphi^{r+1} - \varphi^r\|_2 \geq tol$ **do**

   Run Algorithm 3.2 and output φ^r and u^r.

   Set $r = r + 1$ and $\tau \leftarrow \tau/2$.

end while

Combining Lemma 3.3 and the concavity of $E^\tau(\varphi, u)$ with respect to u yields

$$E^\tau(\varphi^{s+1}, u^s) \geq L_\tau^{u^n}(\varphi^{s+1}, u^{s+1}) \geq E^\tau(\varphi^{s+1}, u^{s+1})$$

where the last inequality comes from the fact that the linearization of a concave functional always locates above the functional. Thus we have

$$E^\tau(\varphi^{s+1}, u^{s+1}) \leq E^\tau(\varphi^s, u^s).$$

The equality holds only when $u^{s+1} = u^s$ and $\varphi^{s+1} = \varphi^s$. Because the energy functional has a low bound, the algorithm converges to a stationary solution in finite steps. □

4. Implementation and discussion on boundary conditions. In this section, we discuss the implementation of the algorithm on flat tori and with Dirichlet boundary conditions on arbitrary domains.

Note that in the algorithm, the only thing we need to compute is $e^{\frac{\tau}{2}\Delta}u$ for a given u. In the follows, we discuss it into two cases.

1. **Periodic boundary conditions:** In the case where we consider Ω as flat tori $([-\pi, \pi]^d)$, we write $e^{\frac{\tau}{2}\Delta}u = G_{\tau/2} * u$ where

$$G_{\tau/2} = \frac{1}{(2\pi\tau)^{d/2}} \exp(-\frac{|x|^2}{2\tau}).$$

Because of the periodic boundary condition, we simply compute it by

$$e^{\frac{\tau}{2}\Delta}u = \mathcal{F}^{-1}(\exp(-|\xi|^2\tau/2)\mathcal{F}(u))$$

where $\mathcal{F}$ is the Fourier transform, ξ is the spectral variable, and $\mathcal{F}^{-1}$ is the inverse Fourier transform. These can be done efficiently by using the fast Fourier transform (FFT). Note that $\exp(-|\xi|^2\tau/2)$ here is the Fourier transform of $\frac{1}{(\pi\tau)^{d/2}} \exp(-\frac{|x|^2}{2\tau})$.

2. **Dirichlet boundary conditions:** In the case where we consider Ω as arbitrary domains with Dirichlet boundary conditions, we consider an extension of Ω to $\tilde{\Omega}$ with $\tilde{\Omega}$ being relatively large and square (See Figure 2 for a diagram). The values of u and φ are extended from Ω to $\tilde{\Omega}$ simply by assigning 0 in $\tilde{\Omega} \setminus \Omega$. In each iteration, we only update φ in the domain Ω which can be simply done by introducing an indicator function of Ω in the computational domain $\tilde{\Omega}$. To be more precise, we use ψ to denote the indicator function of the domain and in the update of φ_ℓ^n, we simply set $\varphi_\ell^n = \varphi_\ell^n\psi$.

   We note that this does not break the energy decaying property of the algorithm. The proposed algorithm does not need a special discretization for the specific domain, one could compute the Dirichlet k-partition in a fixed computational domain for arbitrary shapes by only introducing one auxiliary indicator function ψ.

5. Numerical experiments. In this Section, we demonstrate the diffusion generated method in Algorithms 3.1-3.4 on flat tori and arbitrary domains. All methods were implemented in MATLAB and results reported below were obtained on a laptop with a 2.7GHz Intel Core i5 processor and 8GB of RAM.

If there is no other statement, for all computational results, we set the computational domain as $[-\pi, \pi]^d$ and use a random initial guess for k-partition (*i.e.* u^0) as follows.

Step1. Generate k random points (*seeds*), x_ℓ ($\ell \in [k]$), in the computational domain.

Step2. Compute the Voronoi cell around each seed, A_ℓ ($\ell \in [k]$), and denote

$$u_\ell^0 = \frac{\chi_{A_\ell}}{(\int_\Omega \chi_{A_\ell} \, dx)^{1/2}}.$$

Remark 5.1. If we consider the cases of Dirichlet boundary conditions in arbitrary domains, we restrict k random seeds in the specific domain instead of the whole computational domain.
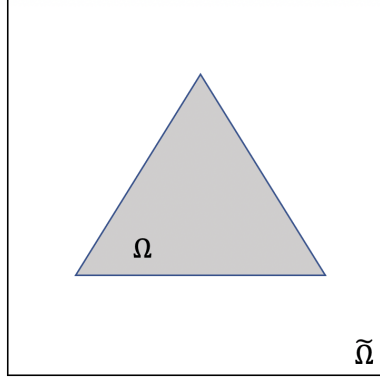
FIG. 2. *A diagram for the extension of a computation domain from Ω to $\tilde{\Omega}$.*

5.1. Accuracy on the computation of the first eigenvalue and eigenfunction. In the first experiment, we check the computation accuracy of the proposed algorithm for computing the first eigenvalue and eigenfunction when φ is fixed as $\varphi = \chi_A$ with $A = [-\pi/2, \pi/2]^2 \subset [-\pi, \pi]^2$. We then solve

$$u = \arg\min_{v \in B} E^\tau(\chi_A, v)$$

by the Step 3 in Algorithm 3.2 associated with an adaptive in time technique. To make no confusion, we write the scheme in the follows.

Algorithm 5.1 A scheme to approximate the first eigenvalue and eigenfunction of a fixed domain with Dirichlet boundary conditions.

   Input: Let $A \subset \Omega$ be a given domain, $\tau > 0$, $tol > 0$, and $u^0 \in B$.
   Output: u^n: approximating the first eigenfunction,
$\frac{1}{\tau} - \frac{1}{\tau} \int_\Omega \chi_A |e^{\frac{\tau}{2}\Delta} u^n|^2 \, dx$: approximating the first eigenvalue.
Set $s = 0$
while $\tau \geq tol$ **do**
    Set $\hat{u}^{s,0} = u^s$ and $m = 0$,
    while $\|u^{s,m+1} - u^{s,m}\|_2 \geq tol$ **do**

$$\hat{u}^{s,m+1} = \frac{e^{\frac{\tau}{2}\Delta}(\chi_A e^{\frac{\tau}{2}\Delta}\hat{u}^{s,m})}{\|e^{\frac{\tau}{2}\Delta}(\chi_A e^{\frac{\tau}{2}\Delta}\hat{u}^{s,m})\|_2}.$$

      Set m = m+1.
    end while
    Set $\tau \leftarrow \frac{\tau}{2}$.
    Set $u^{s+1} = \hat{u}^{s,m}$.
    Set $s = s + 1$.
end while

The exact solution of the first eigenvalue and the corresponding eigenfunction

(with unit $L^2(\Omega)$ norm) in A with the zero Dirichlet boundary condition is written by

$$(5.1) \qquad \lambda_1(A) = 2, \quad u_1^A(x,y) = \begin{cases} \frac{2}{\pi} \cos x \cos y & \text{if } (x,y) \in [-\pi/2, \pi/2]^2, \\ 0 & \text{otherwise.} \end{cases}$$

We first check the accuracy of the new approximation

$$\frac{1}{\tau} - \frac{1}{\tau} \int_\Omega \chi_A |e^{\frac{\tau}{2}\Delta} u|^2 \ dx$$

to the first eigenvalue when $u = u_1^A$. Table 1 lists the approximate eigenvalue computed with different τ. One can observe that the computed eigenvalue converges to the exact value 2 as τ goes to 0 from below. This verifies the motivation of the new proposed approximation and is consistent with Lemma 2.7. We note that this fact is also similar to those in [5, 24].

TABLE 1
Approximate eigenvalues computed on Ω with different $\tau = 2^{-4}$- 2^{-15}. See Section 5.1.

τ	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}
Approximate λ_1	1.8801	1.9388	1.9691	1.9845	1.9922	1.9961
τ	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Approximate λ_1	1.9980	1.9990	1.9995	1.9998	1.9999	1.9999

We then apply Algorithm 5.1 onto the computational domain $[-\pi, \pi]^2$ discretized by 64×64, 128×128, 256×256, 512×512 and 1024×1024 uniform grid points. Figure 3 shows the approximated solution of the eigenfunction and the difference between the approximate solution and the exact solution (5.1), computed on 512×512 discretized mesh. We observe that the support of the approximate eigenfunction is almost in $[-\pi/2, \pi/2]^2$ and is consistent with the exact solution. Table 2 lists the eigenvalues computed on different discretization with different values of *tol*. It is clear that the eigenvalue converges to the exact value 2 with a finer mesh and a smaller τ.
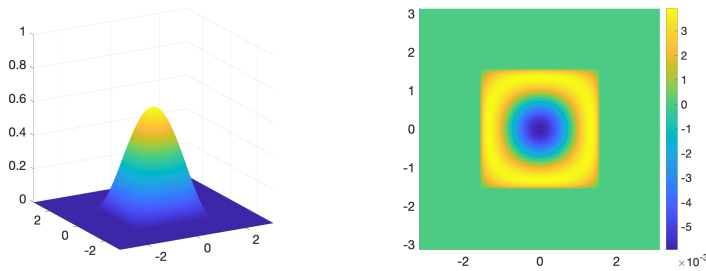

FIG. 3. *Left: the approximate eigenfunction in $[-\pi/2, \pi/2]^2$. Right: The error compared to the exact solution* (5.1). *See Section 5.1.*

To check the convergence of the relaxation to the exact eigenvalue for arbitrary domains in a same computational domain (*i.e.*, $[-\pi, \pi]^2$), we apply Algorithm 5.1 to consider the approximate eigenvalues in the following several domains: 1. a rotated square domain, 2. a rectangle domain, 3. an equilateral triangle domain, 4. a disk domain, and 5. a three quarter disk domain as displayed in Figure 4. For the domain

The approximate eigenvalue computed on Ω with different discretization and different tol = 10^{-3}-10^{-7}. See Section 5.1.

Resolution tol	64×64	128×128	256×256	512×512	1024×1024
10^{-3}	1.7725	1.8514	1.8863	1.9023	1.9100
10^{-4}	1.7938	1.8881	1.9342	1.9543	1.9631
10^{-5}	1.7970	1.8945	1.9463	1.9727	1.9852
10^{-6}	1.7972	1.8949	1.9472	1.9746	1.9890
10^{-7}	1.7972	1.8950	1.9473	1.9749	1.9896

FIG. 4. *Five general domains to check the convergence of approximate eigenvalue as $\tau \to 0$. See Section 5.1.*

TABLE 3
The approximate eigenvalue for different domains. See Section 5.1.

Domain τ	rotated square	rectangle	triangle	disk	3/4 disk
10^{-3}	1.9285	4.6713	5.0156	2.2657	4.3111
10^{-4}	1.9737	4.8693	5.2237	2.3195	4.5068
10^{-5}	1.9877	4.9274	5.2872	2.3360	4.5667
10^{-6}	1.9910	4.9384	5.3008	2.3397	4.5791
10^{-7}	1.9915	4.9397	5.3025	2.3402	4.5806
Reference solution	2	5	16/3	2.3438	4.6182

where the exact solution is not available, we set the reference solution by the solution computed from finite element method with very fine meshes. In all experiments, we use 1024×1024 uniform grids to discretize the computational domain. In Table 3, we list the approximate value obtained by different *tol* and observe that for general domains, the approximate values converge to the exact values as $\tau \to 0$ from below, which is consistent with the fact we proved in Lemma 2.7 and also shows that the relaxation and approximation is insensitive to the domain of consideration.

5.2. Verification on energy decaying and comparisons among Algorithms 3.1-3.4. In this section, we perform a careful study on the energy decaying properties for Algorithms 3.1-3.4. For this study, we simply choose $k = 3$ with a random initial guess. Consider the computational domain discretized by 256×256 grid points and use $\tau = 1/4$ for Algorithms 3.1-3.2 with the random initial guess as shown in the middle of Figure 5. One can observe that Algorithm 3.1 takes 66 iterations to converge while Algorithm 3.2 only takes 29 iterations. This is consistent with the fact that Algorithm 3.2 always finds the stationary solution for u^{n+1} when φ^n is given while Algorithm 3.1 only iterates one step along the descent direction.

However, what is interesting, both algorithms converge to the same stationary solution and Algorithm 3.1 only takes 0.27 *seconds* CPU time while Algorithm 3.2 takes 2.52 *seconds* CPU time. One can understand this by that even Algorithm 3.2 only takes 29 iterations, in each iteration, it takes many more steps for u to converge.
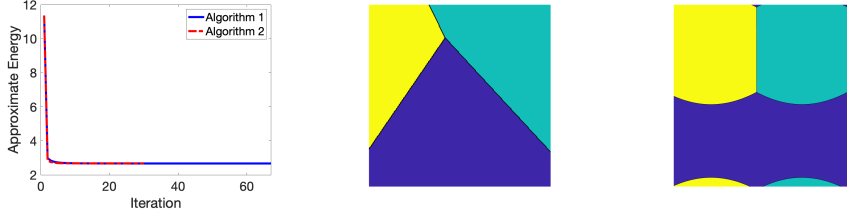

FIG. 5. *Left: The energy curve with respect to the number of iterations for Algorithms 3.1-3.2. Middle: An initial guess for* 3*-partition. Right: The converged solution (same for Algorithm 3.1 and Algorithm 3.2). See Section 5.2.*

Furthermore, we check the energy decaying properties of Algorithms 3.3-3.4 with an initial $\tau = 1/4$ and $tol_\tau = 10^{-4}$. Because adaptive in time techniques are used, we compute the approximate energy in two ways: 1. using the adaptive τ and 2. using a fixed relatively small $\tau = 10^{-4}$. Figure 6 list the energy decaying curve of Algorithms 3.3 and 3.4. If we use a fixed τ to calculate the energy, the approximate energy is monotonically decaying (See the left in Figure 6). If we use varying τ to compute the approximate energy, one can observe that in the iteration of each τ, the approximate energy is decaying but the energy jumps to a large value at the iteration when τ is halved. This is also consistent with the observation from Tables 1 and 2 that the energy converges to the exact value from below (See the middle in Figure 6). In the right of Figure 6, we plot the change between two iterations of φ, one can see that as τ decreases, the partition becomes stationary. In particular, we observe that an initial $\tau = 1/4$ can already efficiently find the stationary solution and decreasing τ only refines the solution a bit but computes the approximate energy more accurately.
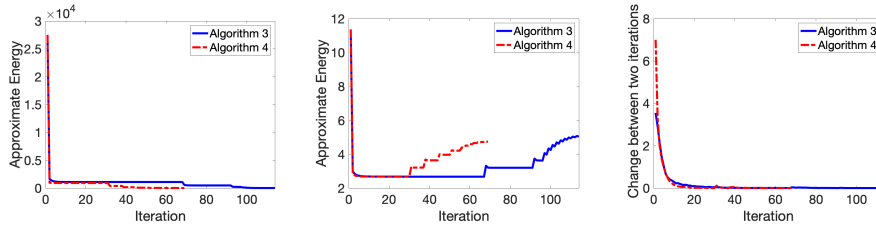

FIG. 6. *Left: Energy decaying curves with respect to iterations when using $\tau = 10^{-4}$ to calculate the approximate energy, Middle: Energy decaying curves with respect to iterations when using the adaptive values of τ to calculate the approximate energy, Right: Change in φ with respect to iterations. See Section 5.2.*

5.3. Acceleration by adaptive in time techniques. In this section, we check the advantage of adaptive in time techniques through two experiments from a same initialization: 1.) adaptive in time for τ changes in $[1/4, 1/8, 1/16, 1/32, 1/64]$, 2.) fix $\tau = 1/64$ without adaptive in time. In Figure 7, we list the snapshots at different iterations for both experiments and observe that both converge to the same solution. In the first row, snapshots at iteration $67, 77, 85, 88, 90$ correspond to stationary solutions at different $\tau = 1/4, 1/8, 1/16, 1/32$, and $1/64$. One can observe that in the

sense of partition, using large τ can achieve almost same result as that obtained from small τ, but it accelerate the convergence dramatically. For instance, using a large time step $\tau = 1/4$ gives a similar solution after 10 iterations while using $\tau = 1/64$ requires about 80 iterations.
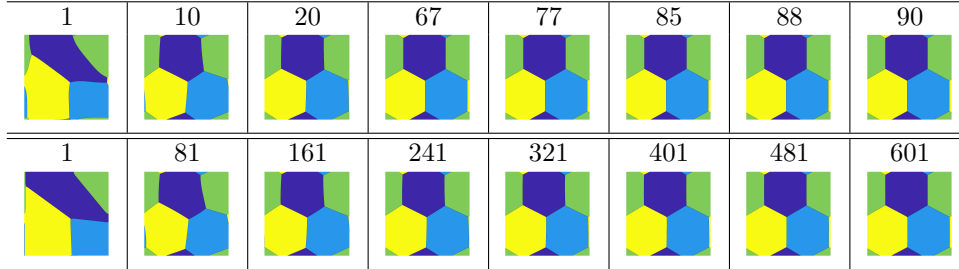

FIG. 7. *First row: snapshots at different iterations in adaptive in time for τ changing in $[1/4, 1/8, 1/16, 1/32, 1/128]$, second row: snapshots at different iterations for a fixed $\tau = 1/64$. See Section 5.3.*

5.4. k-partition for a 2-dimensional periodic domain. In this section, we simply apply Algorithm 3.3 on the calculation of k-partitions for periodic domains in 2-dimensional spaces. In Figure 8, we list the solution of k-partition for $k = 4 - 12$, $14 - 16$, 18, 20, $23 - 25$, 28, 30 and 36. In all results, we discretize the domain with 256^2 grid points, set an initial $\tau = 1/4$ and $tol_\tau = 1/128$, and start with random initial guesses. From our experimental observation, all experiments converge in fewer than about 2-3 hundreds steps and take about 2-75 *seconds* CPU time in average. Here, the average CPU time is the average CPU time of 10 experiments with individually independent random initial guesses for a fixed k. More precisely, the 4-partition case only takes 2 seconds and even the 36-partition computation only takes 75 *seconds*. All reported results are consistent with the results in [31]. Besides, we observe that for most k, especially when k is large, we get hexagon structures. This tessellation for Dirichlet partition is also consistent with the conjecture proposed in [7]. For k = 5,7,10, irregular structures are observed and periodic extensions are plotted in Figure 9.

5.5. k-partition for 2-dimensional arbitrary domains.. In this section, we compute the Dirichlet k-partition in arbitrary domains. We treat the domain as a subset of the computational domain $[-\pi, \pi]^2$ and discretize the computational domain by 512^2 uniform grid points.

In Figure 10, we list k-partitions in an equilateral triangle domain for $k = 2 - 10$, 12, 13, 15, 21, 28, 36, and 45. For $k = 2 - 10$, the results are consistent with the results presented in [4, 10]. For $k > 10$, we select regular structures we observe to list in Figure 10. In particular, when $k = \frac{n(n+1)}{2}, n = 1, 2, \cdots$, one can see very regular structures with hexagon tessellations in the interior layer of the partition (for example, $k = 10, 15, 21, 28, 36$, and 45). For all reported k in Figure 10, the average CPU time for each computation is less than hundred *seconds* starting with random initial guesses. For small k (*e.g.* $k < 10$), the computation only takes about 2 *seconds*.

Figure 11 lists the Dirichlet k-partitions in a square domain, a pentagon domain, a regular hexagon domain, a disk domain, a three-fold domain, and a five-fold domain for $k = 2 - 10$. All results agree with the computational results in [10] and [4] for the reported cases. However, all average computational time is less than
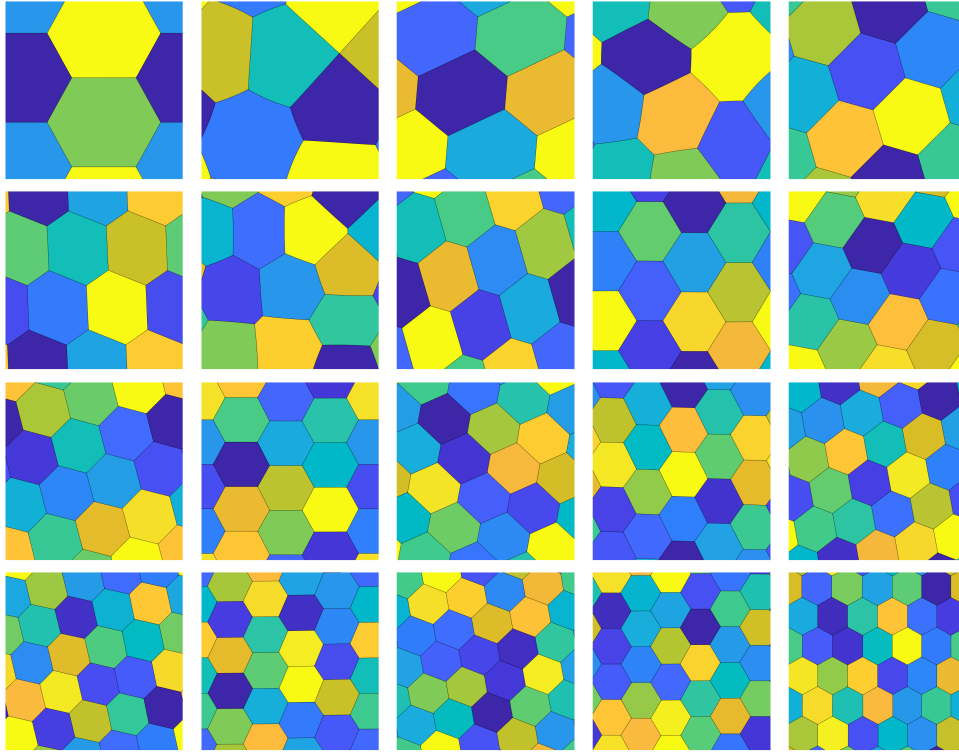
k=4	k=5	k=6	k=7	k=8	k=9	k=10	k=11	k=12	k =14
5.06	8.95	10.48	18.44	16.81	20.56	24.67	28.77	32.81	42.20
k=15	k = 16	k=18	k = 20	k =23	k = 24	k=25	k=28	k = 30	k=36
46.72	51.81	62.33	73.01	89.94	95.88	101.91	120.10	132.47	327.67

FIG. 8. *k-partitions in a periodic domain for k = 4 − 12, 14 − 16, 18, 20, 23 − 25, 28, 30 and 36 together with corresponding approximate eigenvalues. See Section 5.4.*
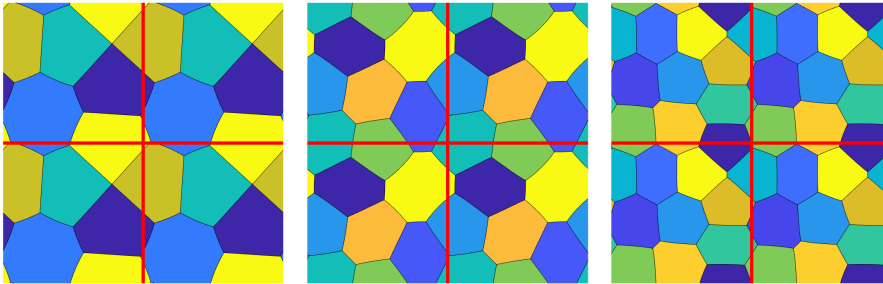
FIG. 9. *Periodic extensions of k-partitions with k = 5, 7, 10. See Section 5.4.*

30 *seconds* starting with random initial guesses. In particular, the computational time of the level set based method proposed in [10] for the five-fold star cases for $k = 3 − 10$ are $23, 27, 31, 35, 40, 44, 49$ and 54 *minutes* respectively with a 100^2 discretization of the domain. However, we find the same results with random initial guesses (with the computational domain $[−\pi, \pi]^2$ discretized by 512^2 grid points) only in $5.8, 9.2, 9.5, 5.4, 18.4, 15.8, 20.8, 19.1$ *seconds*, respectively. It achieves more
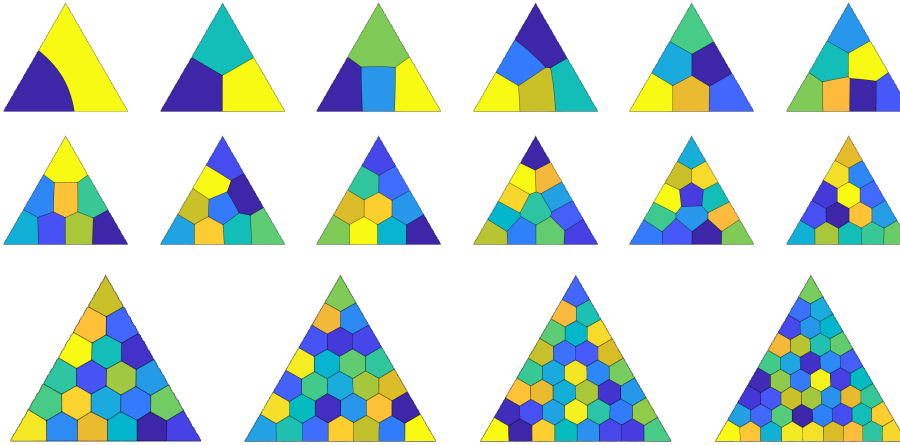
k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9
20.45	35.72	61.16	87.71	113.76	148.74	184.59	220.78
k=10	k=12	k=13	k=15	k=21	k=28	k=36	k=45
256.33	342.54	385.97	473.18	768.24	1142.01	1592.94	3379.16

FIG. 10. *k-partitions in a triangle domain for $k = 2 - 10$, 12, 13, 15, 21, 28, 36, and 45. The table lists all approximate eigenvalues. See Section 5.5.*

than 100 times acceleration. The corresponding approximate values are listed in the table in Figure 11.

5.6. k-partition for a 3-dimensional periodic domain. In this section, we show the efficiency of the proposed algorithm for a 3-dimensional periodic domain. In all experiments, we discretize the periodic computational domain $[-\pi, \pi]^3$ by 128^3 uniform grid points and simply fix $\tau = \pi/16$.

For $k = 4$ and initialization using a random tessellation, we obtain a partition of a 3D flat torus by four identical rhombic dodecahedron structures as displayed in Figure 12 with the approximate eigenvalue 6.96. The result agrees with those reported in [31, 10]. The CPU time for this experiment without parallel computing is only 60 *seconds* and the CPU time reported in [10] for the same case with a 100^3 uniform discretization is 588 *minutes*.

For $k = 8$, we obtain the well-known Weaire–Phelan structure which is a structure representing a foam of equal-sized shapes as shown in the second and third row of Figure 13. Among eight partitions, two regions have the first type of shape which consists of 12 pentagonal faces while the other six regions are the second type of shape which has 2 hexagonal faces surrounded by 12 pentagonal faces. The overall packing is shown in the first row of Figure 13. The approximate eigenvalue is 21.97. The CPU time for this computation from a random initialization with a 128^3 uniform discretization is only 374 *seconds* while the time reported in [10] is 1246 *minutes* for a 100^3 discretization.

For $k = 16$, we obtain the well-known Kelvin Structure which consists of 16 exactly same shapes as shown in Figure 14 with the approximate eigenvalue 69.65. This shape is usually called truncated octahedron which is a space-filling convex polyhedron with 6 square faces and 8 hexagonal faces. The CPU time for this computation from a random initialization with 128^3 uniform discretization is only 656 *seconds*.
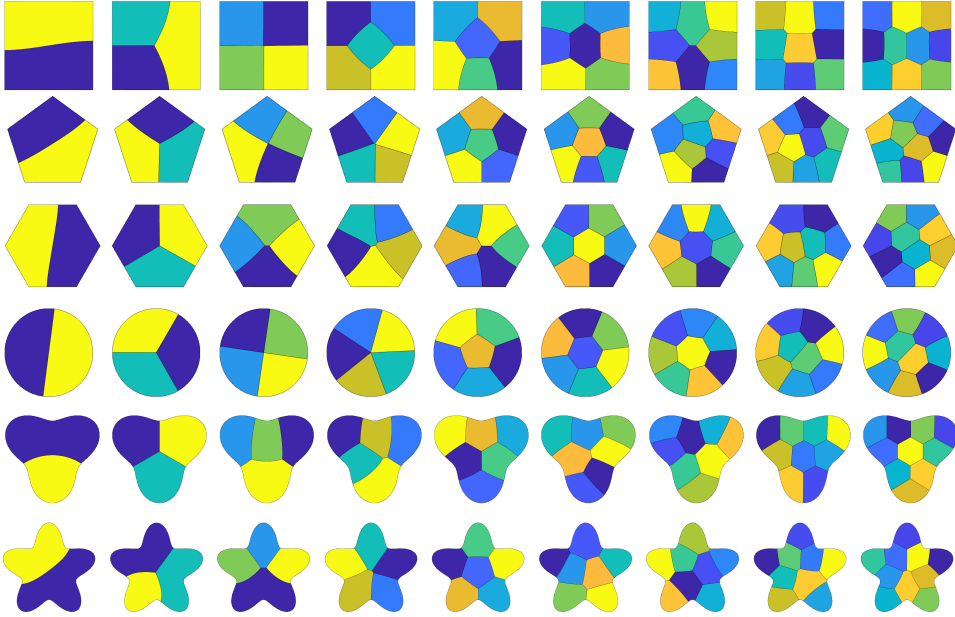
Approximate eigenvalues	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k =10
Square	13.91	26.90	41.69	64.35	87.31	112.78	139.03	172.64	200.69
Petagon	11.84	23.25	38.08	54.67	74.08	98.30	123.78	151.43	179.79
Hexgon	9.29	18.33	35.52	52.78	71.70	89.89	115.59	142.47	169.15
Disk	9.17	18.33	20.48	45.27	61.69	79.72	100.26	124.67	148.40
Three-fold star	9.29	16.28	29.40	44.12	58.81	79.05	100.08	121.51	143.37
Five-fold star	10.69	20.39	31.62	43.17	58.09	78.46	98.62	120.13	142.30

Fig. 11. *Computed k-partition in a square domain, a pentagon domain, a regular hexagon domain, a disk domain, a three-fold star domain, and a five-fold star domain for $k = 2-10$ together with corresponding approximate eigenvalues. See Section 5.5.*

Fig. 12. *Left: A periodic extension on the 4-partition in a 3-dimensional flat torus. Middle: The rhombic dodecahedron structure. Right: The xy-view of the rhombic dodecahedron structure. The approximate eigenvalue is 6.96. The CPU time is 60 seconds. See Section 5.6.*

5.7. k-partition in arbitrary 3-dimensional domains. In this section, we consider the k-partition in several 3-dimensional domains to show the performance of the proposed method in 3-dimensional arbitrary domains with Dirichlet boundary conditions. In all following experiments, we set the computational domain as $[-\pi, \pi]^3$
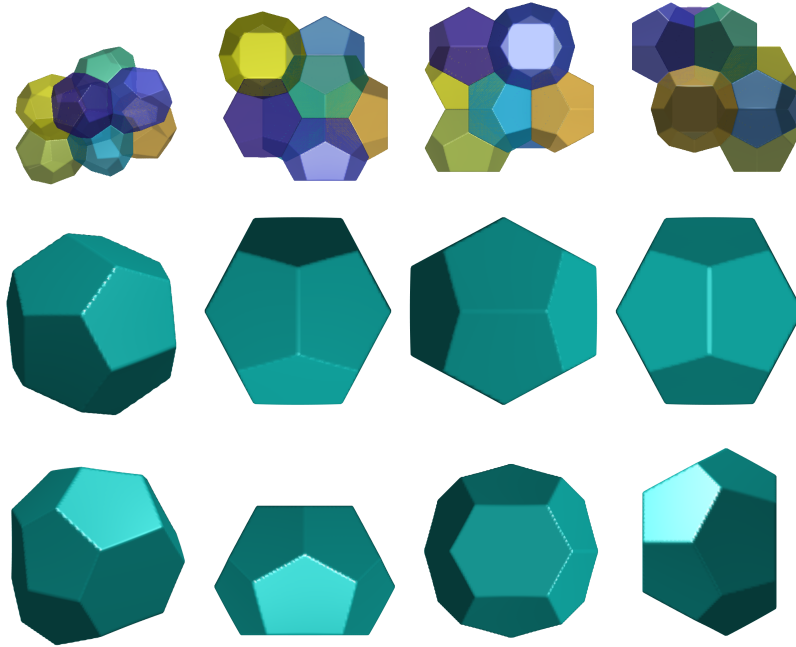
FIG. 13. *First row: The 8-partition for a periodic cube with three different views. Second and third row: Two types of shapes. The shape in the second row consists of 12 pentagonal faces and the shape in the third row has 2 hexagonal faces surrounded by 12 pentagonal faces. The approximate eigenvalue is* 21.97. *The CPU time is 374 seconds. See Section* 5.6.
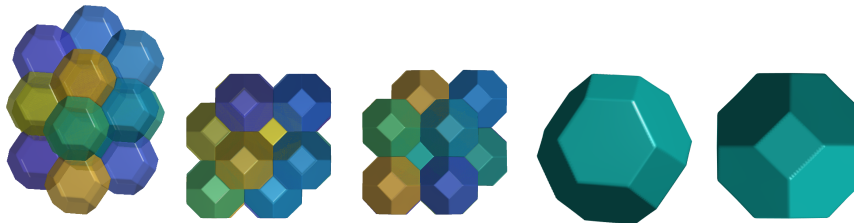

FIG. 14. *The 16-partition for a periodic cube with different views and the shape of the truncated octahedron. Left to right: the 16-partition for a periodic cube, the xy-view of the structure (same as the yz-view), the xz-view of the structure, the truncated octahedron, and the xy-view of the truncated octahedron (same as the xz- and yz- views). The approximate eigenvalue is* 69.65. *The CPU time for this computation from a random initialization with* 128^3 *uniform discretization is only 656 seconds. See Section* 5.6.

discretized by 128^3 uniform grid points and $\tau = \pi/16$. We consider the following three domains: $[-\pi/2, \pi/2]^3 \subset [-\pi, \pi]^3$, a ball centered at the origin with radius $\pi/2$, and a regular tetrahedron centered at the origin with radius of circumsphere $3\pi/4$. We mainly compare the results with the results computed from the method in [5] and reported by Bogosel[1].

In Figure 15, we list the k-partitions in a cube for $k = 3 - 6$, 8, 9 and 14 with some dissections to expose the interior shapes. All results agree with those reported

---

[1]http://www.cmap.polytechnique.fr/~beniamin.bogosel/eig_part3D.html

by Bogosel. The approximate eigenvalues are $23.23, 33.62, 48.02, 62.20, 95.41, 117.07$, and 244.02. The CPU time to obtain these results from random initial guesses are 64, 41, 157, 108, 57, 233, 368 *seconds*, respectively.
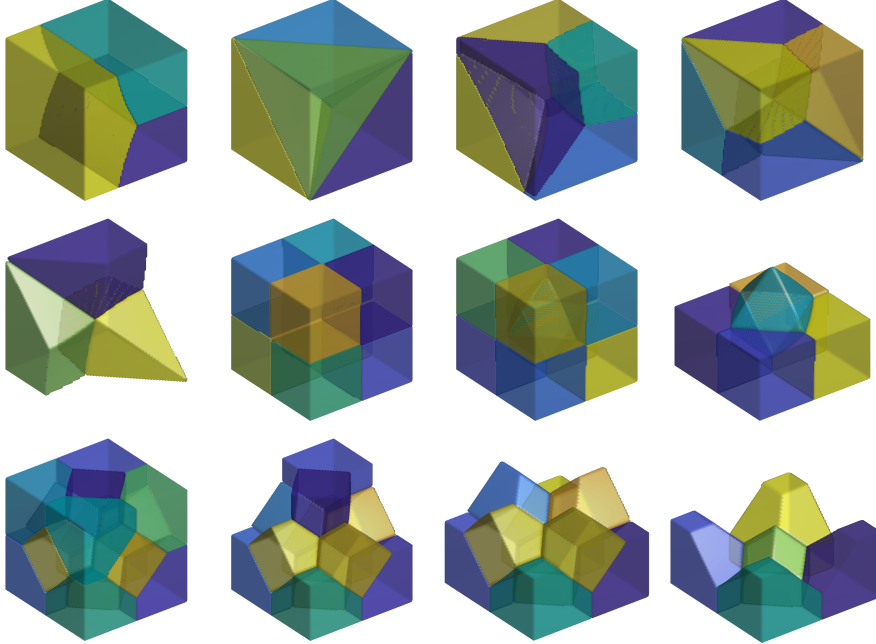

k=3	k=4	k=5	k=6	k=8	k=9	k=14
23.23	33.62	48.02	62.20	95.41	117.07	244.02

FIG. 15. *First row:* $3-6$*-partitions, second row: a dissection of* 6*-partition,* 8*-partition,* 9*-partition, and a dissection of* 9*-partition, third row:* 14*-partition and dissections of the* 14*-partition. The table lists the approximate eigenvalues for different* k*. The CPU time to obtain these results from random initial guesses are* 64*,* 41*,* 157*,* 108*,* 57*,* 233*,* 368 *seconds, respectively. See Section* 5.7*.*

Figure 16 lists the optimal k-partition in a ball with $k = 3, 4, 6, 12, 13$, and 15. The approximate eigenvalues are $31.68, 49.07, 92.95, 285.36, 320.59$, and 405.72, respectively. For $k = 3, 4, 6, 12$, and 13, all results agree with the results reported by Bogosel. The 13-partition is very regular and composed of one interior region and 12 regions that are on the boundary. Interestingly, the interior bubble is very similar to a regular dodecahedron as shown in Figure 16. Furthermore, we observe that the 15-partition is also very regular and is composed of one interior region and 14 regions on the boundary. The interior shape is very similar to the truncated hexagonal trapezohedron that appears in the Weaire–Phelan structure similar to the second shape showed in Figure 13. The shapes on the boundary consist of twelve rounded truncated pentagonal trapezohedron and two rounded truncated hexagonal trapezohedron as shown in Figure 16. These results are also similar to the optimal structure of foam bubbles in the sense of minimizing the total surface area reported in [28]. The CPU time for $k = 3, 4, 6, 12, 13$, and 15 are $20, 41, 193, 296, 143$, and 388 *seconds*, respectively.

When the domain is a tetrahedron, we list the results for $k = 2, 4, 10$, and 20 in Figure 17. For $k = 2, 4, 10$, and 20, we recover the results in 14, 14, 97, and 244 *seconds* with approximate eigenvalues $24.02, 58.71, 254.21$, and 776.74, respectively.
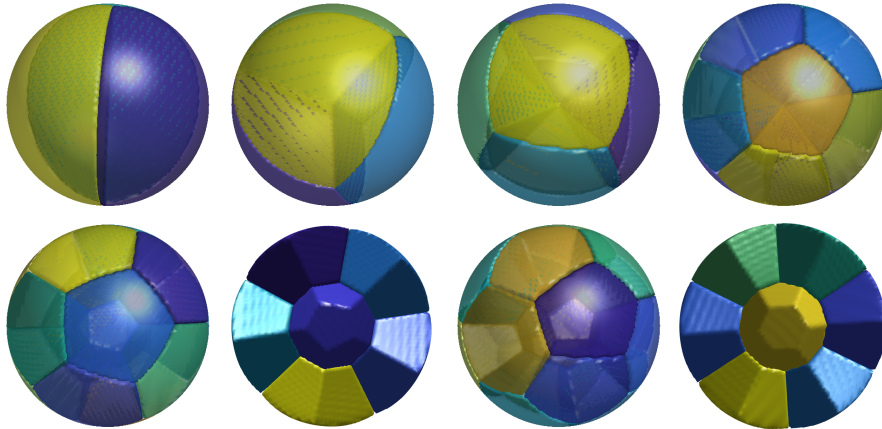
k=3	k=4	k=6	k=12	k=13	k=15
31.68	49.07	92.95	285.36	320.59	405.72

FIG. 16. *The optimal k-partition in a ball with k = 3, 4, 6, 12, 13, and 15 and dissections of 13-partition and 15-partition to expose the interior shapes. The table lists the approximate eigenvalues. The CPU time for k = 3, 4, 6, 12, 13, and 15 are 20, 41, 193, 296, 143, and 388 seconds, respectively. See Section 5.7.*
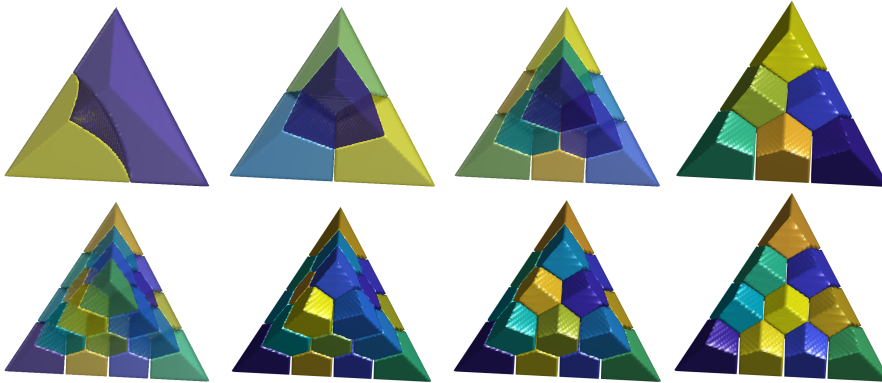

FIG. 17. *Left to right, top to bottom: The optimal k-partition (obtained by the proposed method) in a tetrahedron with k = 2, 4, 10, the dissection of 10-partition, the 20-partition, three dissections of the 20-partition. The CPU time for k = 2, 4, 10, and 20 are 14, 14, 97, and 244 seconds and the approximate eigenvalues are 24.02, 58.71, 254.21, and 776.74, respectively. See Section 5.7.*

6. Conclusion and discussions. In this paper, we proposed a new relaxation of Dirichlet k-partition problems in arbitrary domains and derived a novel algorithm for computing Dirichlet $k-$partitions. The algorithm is very efficient and insensitive to domains. We theoretically proved the monotonically decaying property of the approximate energy. Numerical results show that the proposed method can achieve more than hundreds of times acceleration.

To our knowledge, this is the first paper on relaxing the Dirichlet k-partition via using concave functionals and auxiliary indicator functions. A rigorous proof of the convergence of the new approximation as $\tau \to 0$ is needed for the theoretical guarantee of the new approximation. Besides, in this work, we compute the convolution using

FFT by an extension of the domain of interest. Because the values out of the domain are all 0, one can also implement the algorithm by fast Multipole methods or Non-uniform fast Fourier transform based approaches [18, 29] to further accelerate the algorithm. These are out of the scope of this work and will be reported elsewhere.

REFERENCES

[1] W. BAO, *Ground states and dynamics of multicomponent Bose–Einstein condensates*, Multiscale Modeling & Simulation, 2 (2004), pp. 210–236, https://doi.org/10.1137/030600209.

[2] W. BAO AND Q. DU, *Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow*, SIAM Journal on Scientific Computing, 25 (2004), pp. 1674–1697, https://doi.org/10.1137/s1064827503422956.

[3] B. BOGOSEL, *Efficient algorithm for optimizing spectral partitions*, Applied Mathematics and Computation, 333 (2018), pp. 61–75, https://doi.org/10.1016/j.amc.2018.03.087.

[4] B. BOGOSEL AND B. VELICHKOV, *A multiphase shape optimization problem for eigenvalues: Qualitative study and numerical results*, SIAM Journal on Numerical Analysis, 54 (2016), pp. 210–241, https://doi.org/10.1137/140976406.

[5] B. BOURDIN, D. BUCUR, AND E. OUDET, *Optimal Partitions for Eigenvalues*, SIAM Journal on Scientific Computing, 31 (2010), pp. 4100–4114, https://doi.org/10.1137/090747087.

[6] D. BUCUR, G. BUTAZZO, AND A. HENROT, *Existence results for some optimal partition problems*, Adv. Math. Sci. Appl., 8 (1998), pp. 571–579.

[7] L. A. CAFFERELLI AND F. H. LIN, *An Optimal Partition Problem for Eigenvalues*, J. Sci. Comp., 31 (2007), pp. 5–18, https://doi.org/10.1007/s10915-006-9114-8.

[8] S.-M. CHANG, C.-S. LIN, T.-C. LIN, AND W.-W. LIN, *Segregated nodal domains of two-dimensional multispecies Bose–Einstein condensates*, Physica D: Nonlinear Phenomena, 196 (2004), pp. 341–361, https://doi.org/10.1016/j.physd.2004.06.002.

[9] Q. CHENG AND J. SHEN, *Global constraints preserving scalar auxiliary variable schemes for gradient flows*, SIAM Journal on Scientific Computing, 42 (2020), pp. A2489–A2513, https://doi.org/10.1137/19m1306221.

[10] K. CHU AND S. LEUNG, *A level set method for the dirichlet k-partition problem*, Journal of Scientific Computing, 86 (2021), https://doi.org/10.1007/s10915-020-01368-w.

[11] M. CONTI, S. TERRACINI, AND G. VERZINI, *Nehari's problem and competing species systems*, Annales de l'IHP Analyse Nonlinéaire, 19 (2002), pp. 871–888, https://doi.org/10.1016/s0294-1449(02)00104-x.

[12] M. CONTI, S. TERRACINI, AND G. VERZINI, *An optimal partition problem related to nonlinear eigenvalues*, Journal of Functional Analysis, 198 (2003), pp. 160–196, https://doi.org/10.1016/s0022-1236(02)00105-2.

[13] O. CYBULSKI, V. BABIN, AND R. HOLYST, *Minimization of the Renyi entropy production in the space-partitioning process*, Physical Review E, 71 (2005), p. 46130, https://doi.org/10.1103/physreve.71.046130.

[14] O. CYBULSKI AND R. HOLYST, *Three-dimensional space partition based on the first Laplacian eigenvalues in cells*, Physical Review E, 77 (2008), p. 56101, https://doi.org/10.1103/physreve.77.056101.

[15] Q. DU AND F. LIN, *Numerical approximations of a norm-preserving gradient flow and applications to an optimal partition problem*, Nonlinearity, 22 (2008), pp. 67–83, https://doi.org/10.1088/0951-7715/22/1/005.

[16] S. ESEDOGLU AND F. OTTO, *Threshold dynamics for networks with arbitrary surface tensions*, Communications on Pure and Applied Mathematics, 68 (2015), pp. 808–864, https://doi.org/10.1002/cpa.21527.

[17] B. HELFFER, *On Spectral Minimal Partitions: A Survey*, Milan J. Math., 78 (2010), pp. 575–590, https://doi.org/10.1007/s00032-010-0129-0.

[18] S. JIANG, D. WANG, AND X.-P. WANG, *An efficient boundary integral scheme for the MBO threshold dynamics method via the NUFFT*, Journal of Scientific Computing, 74 (2017), pp. 474–490, https://doi.org/10.1007/s10915-017-0448-1.

[19] J. MA, D. WANG, X.-P. WANG, AND X. YANG, *A characteristic function-based algorithm for geodesic active contours*, SIAM Journal on Imaging Sciences, 14 (2021), pp. 1184–1205,

https://doi.org/10.1137/20m1382817.

[20] M. Miranda Jr, D. Pallara, F. Paronetto, and M. Preunkert, *Short-time heat flow and functions of bounded variation in $\mathbb{R}^n$*, in Annales de la Faculté des sciences de Toulouse: Mathématiques, vol. 16, 2007, pp. 125–145.

[21] B. Osting and T. H. Reeb, *Consistency of dirichlet partitions*, SIAM Journal on Mathematical Analysis, 49 (2017), pp. 4251–4274, https://doi.org/10.1137/16m1098309.

[22] B. Osting and D. Wang, *A diffusion generated method for orthogonal matrix-valued fields*, Mathematics of Computation, 89 (2019), pp. 515–550, https://doi.org/10.1090/mcom/3473.

[23] B. Osting and D. Wang, *Diffusion generated methods for denoising target-valued images*, Inverse Problems & Imaging, 14 (2020), pp. 205–232, https://doi.org/10.3934/ipi.2020010.

[24] B. Osting, C. D. White, and E. Oudet, *Minimal Dirichlet energy partitions for graphs*, SIAM J. Scientific Computing, 36 (2014), pp. A1635–A1651, https://doi.org/10.1137/130934568.

[25] J. Shen, J. Xu, and J. Yang, *The scalar auxiliary variable (SAV) approach for gradient flows*, Journal of Computational Physics, 353 (2018), pp. 407–416, https://doi.org/10.1016/j.jcp.2017.10.021.

[26] J. Shen, J. Xu, and J. Yang, *A new class of efficient and robust energy stable schemes for gradient flows*, SIAM Review, 61 (2019), pp. 474–506, https://doi.org/10.1137/17m1150153.

[27] D. Wang, *An efficient iterative method for reconstructing surface from point clouds*, Journal of Scientific Computing, 87 (2021), https://doi.org/10.1007/s10915-021-01457-4.

[28] D. Wang, A. Cherkaev, and B. Osting, *Dynamics and stationary configurations of heterogeneous foams*, PLOS ONE, 14 (2019), p. e0215836, https://doi.org/10.1371/journal.pone.0215836.

[29] D. Wang, S. Jiang, and X.-P. Wang, *An efficient boundary integral scheme for the threshold dynamics method II: Applications to wetting dynamics*, Journal of Scientific Computing, 81 (2019), pp. 1860–1881, https://doi.org/10.1007/s10915-019-01067-1.

[30] D. Wang, H. Li, X. Wei, and X.-P. Wang, *An efficient iterative thresholding method for image segmentation*, Journal of Computational Physics, 350 (2017), pp. 657–667, https://doi.org/10.1016/j.jcp.2017.08.020.

[31] D. Wang and B. Osting, *A diffusion generated method for computing dirichlet partitions*, Journal of Computational and Applied Mathematics, 351 (2019), pp. 302–316, https://doi.org/10.1016/j.cam.2018.11.015.

[32] D. Wang, B. Osting, and X.-P. Wang, *Interface dynamics for an allen–cahn-type equation governing a matrix-valued field*, Multiscale Modeling & Simulation, 17 (2019), pp. 1252–1273, https://doi.org/10.1137/19m1250595.

[33] D. Wang and X.-P. Wang, *The iterative convolution-thresholding method (ictm) for image segmentation*, arXiv preprint arXiv:1904.10917, (2019).

[34] D. Wang, X.-P. Wang, and X. Xu, *An improved threshold dynamics method for wetting dynamics*, Journal of Computational Physics, 392 (2019), pp. 291–310, https://doi.org/10.1016/j.jcp.2019.04.037.

[35] X. Xu, D. Wang, and X.-P. Wang, *An efficient threshold dynamics method for wetting on rough surfaces*, Journal of Computational Physics, 330 (2017), pp. 510–528, https://doi.org/10.1016/j.jcp.2016.11.008.