

# Quadratic maximum-entropy serendipity shape functions for arbitrary planar polygons

N. Sukumar\*

*Department of Civil and Environmental Engineering, University of California  
One Shields Avenue, Davis, CA 95616, U.S.A.*

---

## Abstract

In this paper, we present the development of quadratic serendipity shape functions on planar convex and nonconvex polygons. Drawing on the work of Bompadre et al. [1] and Hormann and Sukumar [2], we adopt a relative entropy measure for signed (positive or negative) shape functions, with nodal *prior* weight functions that have the appropriate zero-set on the boundary of the polygon. We maximize the objective functional subject to the constraints for quadratic completeness proposed by Rand et al. [3]. Along an edge of a polygon, the approximation is identical to univariate Bernstein polynomials: the choice of the nodal prior weight function ensures that the shape functions satisfy a *weak* Kronecker-delta property on each edge. The shape functions are well-defined for arbitrary planar polygons without self-intersections. On using a modified numerical integration scheme, we show that the quadratic patch test is passed on polygonal meshes with convex and nonconvex elements. Numerical tests for the Poisson equation on self-similar trapezoidal meshes and quasiuniform polygonal meshes are presented, which reveal the sound accuracy of the method, and optimal rates of convergence in the  $L^2$  norm and the  $H^1$  seminorm are established.

*Key words:* maximum-entropy principle, relative entropy, Bernstein basis,

---

\*Corresponding author

Email address: [nsukumar@ucdavis.edu](mailto:nsukumar@ucdavis.edu) (N. Sukumar\*)

## 1. Introduction

Interpolating given discrete data with continuous functions in one or more variables is a fundamental problem in diverse fields of science and engineering. Barycentric coordinates, which were introduced by Möbius in 1827, still provide perhaps the most convenient way to linearly interpolate data prescribed at the vertices of a  $d$ -dimensional simplex in  $\mathbb{R}^d$ . In recent years, with an eye on computer graphics and finite element applications, barycentric coordinates on simplices have been extended to polygons and polyhedra. These coordinates (synonymous with shape functions in finite element methods) for polygons and polyhedra are referred to as *generalized barycentric coordinates*. On convex polygons, generalized barycentric coordinates are non-negative, they form a partition of unity, and have linear precision. Consider a planar convex polygon with  $n$  vertices ( $n$ -gon). Given the  $n$  vertices of a polygon, generalized barycentric coordinates are readily constructed using many different approaches, such as Wachspress coordinates [4], mean value coordinates [5], harmonic coordinates [6], and maximum-entropy (max-ent) coordinates [7] to name a few. On convex and nonconvex polygons, harmonic and maximum-entropy coordinates are linearly precise and strictly non-negative (convex approximation). Along with the advances of generalized barycentric coordinates in computer graphics, there has been a concomitant development of barycentric finite element and boundary element methods on polygons and polyhedra [7–18], with contributions in interpolation error estimates [19, 20], and applications in fracture modeling [21–23], topology optimization [24–26], and mesh generation [27–32].

The extension to higher-order approximations on polygonal elements has received limited attention. Milbrandt and Pick [11] present a construction for

transition elements on  $p$ -adaptive polygonal meshes. Recently, using generalized barycentric coordinates as a starting point, Rand et al. [3] developed quadratic serendipity finite elements on convex polygons. In their work, nodal shape functions are associated with vertices and with the mid-point of each boundary edge; in the spirit of 8-node (quadratic) serendipity isoparametric finite elements, there are no shape functions associated with interior nodes. In Reference [3], the criteria and constraints (reproducing conditions) that must be met by such shape functions are presented, and it is shown that the approximation delivers optimal rates of convergence in Sobolev norms. Within the family of entropy-based schemes, the construction of convex approximations with quadratic completeness has been pursued [33–35]; to retain the convexity property of the approximation, the quadratic reproducing conditions have to be modified, which complicates the formulation. Bompadre et al. [1] relaxed the convexity constraint and used a modified entropy measure to construct higher-order *local* max-ent approximation schemes.

In this paper, we adopt the constraints given in Rand et al. [3] and recast the objective functional introduced by Bompadre et al. [1] within the framework of relative entropy with *prior* weight functions [2, 36]. In particular, we use the nodal prior weight functions introduced in Hormann and Sukumar [2] for arbitrary planar polygons without self-intersections. For any point on a boundary edge of a polygonal element, only three nodal prior weight functions (nodes that lie on the edge) are non-zero. This ensures that the facet-reducing property [37] is also preserved for signed shape functions, and  $C^0$ -conforming approximations are realized on polygonal meshes. Our formulation provides serendipity shape functions with quadratic completeness on convex and nonconvex polygonal elements. The objective functional and the linear constraints are introduced in Section 2.1, and the max-ent variational formulation is presented in Section 2.2. The numerical algorithm

is presented in Section 2.3, and the expressions for the derivatives of the max-ent shape functions are derived in Section 2.4. Mesh generation using maximal-Poisson sampling is discussed in Section 3, and in Section 4 we tailor the cell-based meshfree numerical integration scheme proposed by Duan et al. [38] for quadratic polygonal elements. Numerical tests for Poisson problems are presented in Section 5. We show that the quadratic patch test is passed, and the quadratic polygonal finite element delivers optimal rates of convergence in Sobolev norms on self-similar trapezoidal meshes; on such meshes, quadratic serendipity isoparametric finite elements yield suboptimal convergence rates [39–41]. We close with some final remarks in Section 6.

## 2. Quadratic Maximum-Entropy Element

Consider a planar polygonal element  $\Omega_e \subset \mathbb{R}^2$  with  $n$  vertices (Fig. 1a). We do not allow polygons that self-intersect, but do permit weakly simple polygons in which three or more consecutive vertices are collinear. Let  $h_e$  denote the diameter of the polygon. To obtain shape functions that can reproduce bivariate quadratic functions, we assign an additional node to the mid-point of each edge. The vertex nodes are contained in the index set  $V = \{1, \dots, n\}$ , whereas the mid-edge nodes are contained in the index set  $E = \{n+1, \dots, 2n\}$ . We denote any point in  $\bar{\Omega}_e$  by  $\mathbf{x} \equiv (x, y)$  and the nodal coordinates by  $\mathbf{x}_a \equiv (x_a, y_a)$ . The coordinates of the vertex nodes are  $\mathbf{x}_a$  ( $a = 1, 2, \dots, n$ ), which are ordered in counter-clockwise orientation. The coordinates of the nodes in  $E$  are:  $\mathbf{x}_{n+a} = (\mathbf{x}_a + \mathbf{x}_b)/2$ , where  $a = 1, 2, \dots, n$  and  $b = \text{mod}(a, n)+1$  with  $\text{mod}$  denoting the modulo operation. The  $a$ -th edge  $\mathcal{E}_a := (a, a+1)$  has nodal connectivity  $(a, a+n, a+1)$  (Fig. 1b). Cyclic order is assumed for the edge-connectivity, i.e.,  $\mathcal{E}_n := (n, 1)$ . Given a real-valued function  $u(\mathbf{x}) : \Omega_e \rightarrow \mathbb{R}$ , the numerical approximation for  $u(\mathbf{x})$  within the

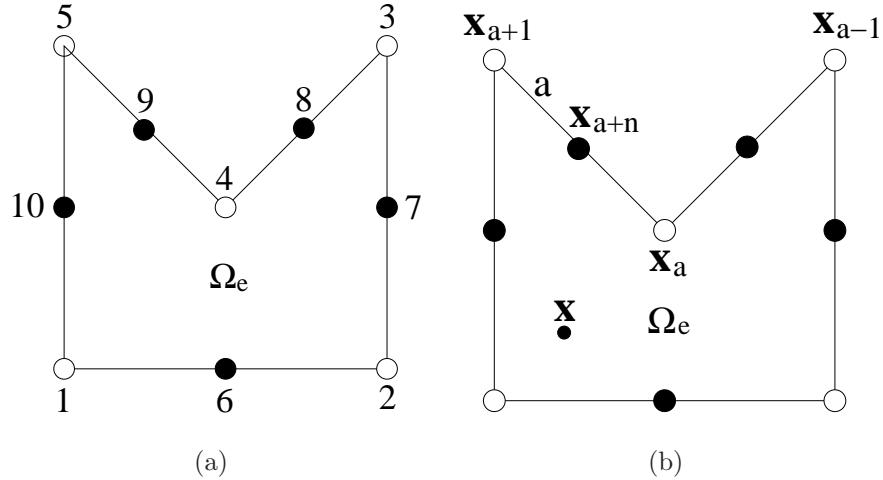


Fig. 1: Planar polygon  $\Omega_e$ . (a) Node numbering for vertex and mid-edge nodes; and (b) Nodal connectivity for edge  $a$  and point  $\mathbf{x} \in \Omega_e$  are depicted.

a polygonal element is written as:

$$u_e^h(\mathbf{x}) = \sum_{a=1}^{2n} \phi_a(\mathbf{x}) u_a, \quad (1)$$

where  $\phi_a(\mathbf{x})$  is the shape function associated with node  $a$ , and  $u_a$  are nodal coefficients. We first discuss the objective functional and the constraint equations, and then present the variational formulation and its numerical implementation to determine  $\phi_a(\mathbf{x})$  so that (1) has quadratic completeness.

### 2.1. Objective functional and linear constraints

In the maximum-entropy approach [42, 43], the unknown discrete probability distribution  $\mathbf{p}$  is determined by maximizing the Shannon entropy [44] or the negative of the relative entropy [45, 46] subject to constraints on  $\mathbf{p}$ . The maximum-entropy principle was used to construct strictly non-negative linearly complete basis functions [7, 36, 37].

For positive-negative probability distributions, Skilling [47] introduced an entropy measure based on relative entropy, which was adopted to construct signed meshfree basis functions [36]. Recently, Bompadre et al. [1] proposed a

modified entropy functional for constructing higher-order complete meshfree basis functions. Herein, to meet our objectives, we recast their formulation within the more general setting of relative entropy.

Let  $\{p_a^+(\mathbf{x})\}_{a=1}^{2n} : \Omega_e \rightarrow \mathbb{R}_+^{2n}$  and  $\{p_a^-(\mathbf{x})\}_{a=1}^{2n} : \Omega_e \rightarrow \mathbb{R}_+^{2n}$  be non-negative nodal functions, where  $\mathbb{R}_+^{2n}$  denotes the non-negative orthant (box in  $\mathbb{R}^{2n}$ ). We define  $\phi_a(\mathbf{x}) := p_a^+(\mathbf{x}) - p_a^-(\mathbf{x})$  [1] to be the shape function associated with node  $a$ , and set  $w_a(\mathbf{x})$  as the nodal prior weight function for both  $p_a^+(\mathbf{x})$  and  $p_a^-(\mathbf{x})$ . Then, for a fixed  $\mathbf{x} \in \Omega_e$ , we can write the negative of the relative (Shannon-Jaynes) entropy for the two distributions as [45]:

$$H(\mathbf{p}^+, \mathbf{p}^-) = - \sum_{a=1}^{2n} p_a^+(\mathbf{x}) \ln \left( \frac{p_a^+(\mathbf{x})}{w_a(\mathbf{x})} \right) - \sum_{a=1}^{2n} p_a^-(\mathbf{x}) \ln \left( \frac{p_a^-(\mathbf{x})}{w_a(\mathbf{x})} \right). \quad (2)$$

If a Gaussian function is chosen as the *prior*, i.e.,  $w_a(\mathbf{x}) = \exp(-\beta||\mathbf{x}_a - \mathbf{x}||^2)$ , then the above objective functional reduces to that proposed in Reference [1].

Now, we need to define the appropriate constraints, which will ensure quadratic completeness and  $C^0$ -conformity on each edge of the polygonal element. To this end, we follow the recent work of Rand et al. [3], where quadratic serendipity finite elements are developed. Let  $\{\lambda_a(\mathbf{x})\}_{a=1}^n$  be generalized barycentric coordinates for an  $n$ -gon. To construct a space of functions that can reproduce bivariate quadratic functions, Rand et al. [3] considered the constraints that stem from the pairwise products of generalized barycentric coordinates:

$$\sum_{a=1}^n \sum_{\tilde{a}=1}^n \mu_{a\tilde{a}} = 1, \quad (3a)$$

$$\sum_{a=1}^n \sum_{\tilde{a}=1}^n \mu_{a\tilde{a}} \left( \frac{\mathbf{x}_a + \mathbf{x}_{\tilde{a}}}{2} \right) = \mathbf{x}, \quad (3b)$$

$$\sum_{a=1}^n \sum_{\tilde{a}=1}^n \mu_{a\tilde{a}} \mathbf{x}_a \otimes \mathbf{x}_{\tilde{a}} = \mathbf{x} \otimes \mathbf{x}, \quad (3c)$$

where  $\{\mu_{a\tilde{a}}\} := \{\lambda_a \lambda_{\tilde{a}}\}_{a,\tilde{a}=1}^n$ . The set  $\{\mu_{a\tilde{a}}\}$  does not constitute a bases for the space of bivariate quadratic functions since it contains linearly dependent

functions. Note that all functions  $\mu_{a\tilde{a}}$  associated with nodes that are located in the interior of the polygon are zero on the boundary of the polygon. Motivated by this observation and the serendipity construction for finite elements, Rand et al. [3] rewrote (3) by retaining only functions associated with vertex nodes and mid-edge nodes of the polygon, which is a total of  $2n$  functions. The constraint equations so developed have a feasible solution [3]. Due to symmetry ( $\mu_{ab} = \mu_{ba}$ ), the constraint equations in (3) can be simplified, which we write in terms of the polygonal shape functions as:

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) = 1 \quad (4a)$$

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) \mathbf{x}_a = \mathbf{x} \quad (4b)$$

$$\sum_{a=1}^n \phi_a(\mathbf{x}) \mathbf{x}_a \otimes \mathbf{x}_a + \sum_{a=1}^n \phi_{a+n}(\mathbf{x}) \left( \frac{\mathbf{x}_a \otimes \mathbf{x}_b + \mathbf{x}_b \otimes \mathbf{x}_a}{2} \right) = \mathbf{x} \otimes \mathbf{x}, \quad (4c)$$

where  $b = \text{mod}(a, n) + 1$ . Note that the constraint equations in (4) are written in a form that slightly differs from that used in Reference [3].

We now proceed to express the constraints in terms of the shifted nodal coordinates,  $\mathbf{x}_a - \mathbf{x}$ . On using the linear reproducing conditions, we can write the equality:  $\sum_a \phi_a(\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) = \sum_a \phi_a[\mathbf{x}_a \otimes \mathbf{x}_a - \mathbf{x} \otimes \mathbf{x}]$ , and hence the quadratic constraint in (4c) can be written as:

$$\begin{aligned} & \sum_{a=1}^{2n} \phi_a(\mathbf{x}) (\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) + \sum_{a=1}^n \phi_{a+n}(\mathbf{x}) \left( \frac{\mathbf{x}_a \otimes \mathbf{x}_b + \mathbf{x}_b \otimes \mathbf{x}_a}{2} \right) \\ & - \sum_{a=1}^n \phi_{a+n}(\mathbf{x}) \mathbf{x}_{a+n} \otimes \mathbf{x}_{a+n} = \mathbf{0}. \end{aligned}$$

Using  $\mathbf{x}_{a+n} = (\mathbf{x}_a + \mathbf{x}_b)/2$ , and after some algebraic simplifications, the above equation reduces to:

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) (\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) - \sum_{a=1}^n \phi_{a+n}(\mathbf{x}) \left[ \frac{(\mathbf{x}_b - \mathbf{x}_a) \otimes (\mathbf{x}_b - \mathbf{x}_a)}{4} \right] = \mathbf{0},$$

and therefore the constraints in (4) take the form:

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) = 1 \quad (5a)$$

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) (\mathbf{x}_a - \mathbf{x}) = \mathbf{0} \quad (5b)$$

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) [(\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) - \mathbf{d}_a] = \mathbf{0}, \quad (5c)$$

where

$$\mathbf{d}_1 = \dots = \mathbf{d}_n = \mathbf{0}, \quad \mathbf{d}_{a+n} = \frac{(\mathbf{x}_b - \mathbf{x}_a) \otimes (\mathbf{x}_b - \mathbf{x}_a)}{4} \quad (1 \leq a \leq n). \quad (5d)$$

The nodal coefficients  $\mathbf{d}_a$  in the above equation have an interpretation of so-called *nodal gaps* [35], which are used to ensure feasibility of reproducing  $\mathbf{x} \otimes \mathbf{x}$  in second-order convex approximation schemes. To aid in the derivation of the variational formulation, we define a nodal vector  $\mathbf{c}_a(\mathbf{x})$ :

$$\mathbf{c}_a(\mathbf{x}) := \begin{Bmatrix} 1 \\ \mathbf{x}_a - \mathbf{x} \\ (\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) \end{Bmatrix} \quad (1 \leq a \leq n), \quad (6a)$$

$$\mathbf{c}_a(\mathbf{x}) := \begin{Bmatrix} 1 \\ \mathbf{x}_a - \mathbf{x} \\ (\mathbf{x}_a - \mathbf{x}) \otimes (\mathbf{x}_a - \mathbf{x}) - \frac{(\mathbf{x}_b - \mathbf{x}_a) \otimes (\mathbf{x}_b - \mathbf{x}_a)}{4} \end{Bmatrix} \quad (a > n), \quad (6b)$$

where the last entry in  $\mathbf{c}_a$  is a symmetric  $2 \times 2$  matrix and hence only the entries in the upper triangle (Voigt notation) are stored. Now, the reproducing conditions in (4) can be written in compact form as:

$$\sum_{a=1}^{2n} \phi_a(\mathbf{x}) \mathbf{c}_a(\mathbf{x}) = \mathbf{q}, \quad (7)$$

where  $\mathbf{q} = \{1 \ \mathbf{0}\}^T$  is a constant  $(6 \times 1)$  vector.

The final ingredient is the choice for  $w_a(\mathbf{x})$ , which we construct such that when combined with the constraints in (7), it yields Bernstein polynomials [48] as the unique solution for the shape functions along each boundary edge. To this end, we select an edge weight function that vanishes along edge  $a$  and is strictly positive elsewhere in the plane. Two choices for the edge weight are [2]:

$$\rho_a(\mathbf{x}) = \|\mathbf{x}_a - \mathbf{x}\| + \|\mathbf{x}_{a+1} - \mathbf{x}\| - \|\mathbf{x}_{a+1} - \mathbf{x}_a\| \quad (8a)$$

$$\rho_a(\mathbf{x}) = (\mathbf{x}_a - \mathbf{x}) \cdot (\mathbf{x}_{a+1} - \mathbf{x}) + \|\mathbf{x}_a - \mathbf{x}\| \|\mathbf{x}_{a+1} - \mathbf{x}\|, \quad (8b)$$

where  $\mathbf{x}_{n+1} := \mathbf{x}_1$  and  $\rho_a(\mathbf{x}) \geq 0$  with  $\rho_a(\mathbf{x}) \equiv 0$  if  $\mathbf{x}$  lies on the edge  $a$  (see Fig. 1). Now, we define the nodal prior weight functions as [2]:

$$\begin{aligned} w_a(\mathbf{x}) &= \frac{\Pi_a(\mathbf{x})}{\sum_{b=1}^{2n} \Pi_b(\mathbf{x})}, \quad \Pi_a(\mathbf{x}) = \frac{1}{\rho_{a-1}(\mathbf{x}) \rho_a(\mathbf{x})} \quad (a = 1, \dots, n), \\ \Pi_a(\mathbf{x}) &= \frac{1}{\rho_a(\mathbf{x})} \quad (a = n+1, \dots, 2n), \end{aligned} \quad (9)$$

where  $\rho_0(\mathbf{x}) := \rho_n(\mathbf{x})$  and from (9), we note that on the  $a$ -th edge, only the *priors* for nodes  $a$ ,  $a+1$ , and  $a+n$  have non-zero contributions. The edge weight in (8a), which is based on the triangle inequality, is computationally more efficient and has slower growth than the one in (8b). In the numerical computations, the edge weight given in (8a) is used. In Fig. 2, two nodal prior weight functions for a square are shown. The gradient of (8a) is:

$$\nabla \rho_a(\mathbf{x}) = -\frac{\mathbf{x}_a - \mathbf{x}}{\|\mathbf{x}_a - \mathbf{x}\|} - \frac{\mathbf{x}_{a+1} - \mathbf{x}}{\|\mathbf{x}_{a+1} - \mathbf{x}\|}. \quad (10)$$

## 2.2. Variational formulation

On using (2) and (7), the max-ent variational formulation is: for a given  $\mathbf{x} \in \Omega_e$ , find  $p_a^+(\mathbf{x}), p_a^-(\mathbf{x}) \in \mathbb{R}_+$  ( $a = 1, \dots, 2n$ ) as the solution of the

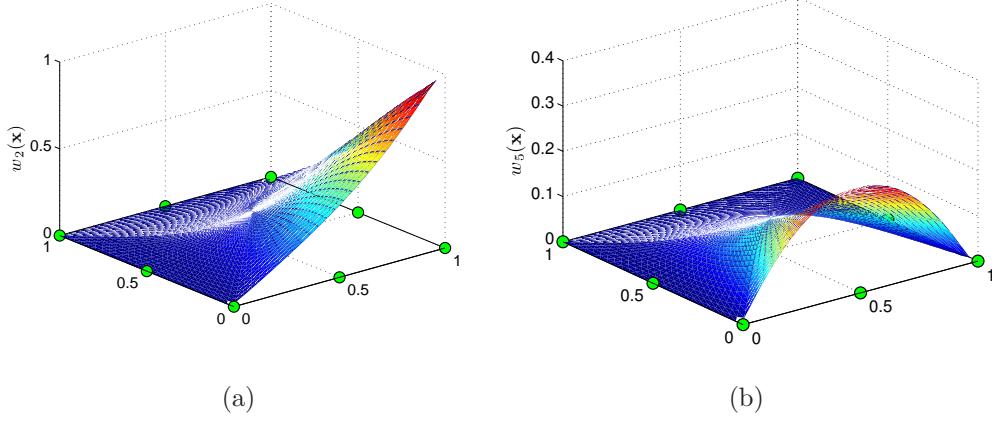


Fig. 2: Nodal prior weight functions for a square. (a)  $w_2(\mathbf{x})$ ; and (b)  $w_5(\mathbf{x})$ .

constrained (concave) optimization problem:

$$\max_{\mathbf{p}^+, \mathbf{p}^- \in \mathbb{R}_+^{2n}} \left[ - \sum_{a=1}^{2n} p_a^+(\mathbf{x}) \ln \left( \frac{p_a^+(\mathbf{x})}{w_a(\mathbf{x})} \right) - \sum_{a=1}^{2n} p_a^-(\mathbf{x}) \ln \left( \frac{p_a^-(\mathbf{x})}{w_a(\mathbf{x})} \right) \right], \quad (11a)$$

subject to the following linear constraints:

$$\sum_{a=1}^{2n} [p_a^+(\mathbf{x}) - p_a^-(\mathbf{x})] \mathbf{c}_a(\mathbf{x}) = \mathbf{q}. \quad (11b)$$

Equation (11) admits a unique solution since the objective functional is strictly concave in  $\Omega_e$  [49] and the linear constraints constitute a feasible set [3]. After solving for  $\mathbf{p}^+(\mathbf{x})$  and  $\mathbf{p}^-(\mathbf{x})$ ,  $\phi_a(\mathbf{x}) = p_a^+(\mathbf{x}) - p_a^-(\mathbf{x})$  yields the nodal shape functions.

The presentation of the variational formulation closely follows Bompadre et al. [1]. Let  $\boldsymbol{\lambda} \in \mathbb{R}^{\frac{(p+1)(p+2)}{2}}$  (here  $p = 2$ ) contain the six Lagrange multipliers associated with the linear constraints in (11b). Then, the Lagrangian is:

$$\begin{aligned} \mathcal{L}(\mathbf{p}^+, \mathbf{p}^-, \boldsymbol{\lambda}) = & - \sum_{a=1}^{2n} \left[ p_a^+(\mathbf{x}) \ln \left( \frac{p_a^+(\mathbf{x})}{w_a(\mathbf{x})} \right) + p_a^-(\mathbf{x}) \ln \left( \frac{p_a^-(\mathbf{x})}{w_a(\mathbf{x})} \right) \right] \\ & - \boldsymbol{\lambda} \cdot \left( \sum_{a=1}^{2n} [p_a^+(\mathbf{x}) - p_a^-(\mathbf{x})] \mathbf{c}_a(\mathbf{x}) - \mathbf{q} \right). \end{aligned} \quad (12)$$

The Karush-Kuhn-Tucker (KKT) first-order optimality conditions are:

$$-1 - \ln p_a^+(\mathbf{x}) + \ln w_a(\mathbf{x}) - \boldsymbol{\lambda} \cdot \mathbf{c}_a(\mathbf{x}) = 0,$$

$$-1 - \ln p_a^-(\mathbf{x}) + \ln w_a(\mathbf{x}) + \boldsymbol{\lambda} \cdot \mathbf{c}_a(\mathbf{x}) = 0,$$

and on noting that  $p_a^+, p_a^- : \Omega_e \times \mathbb{R}^6 \rightarrow \mathbb{R}_+$ , the above yields

$$p_a^+(\mathbf{x}, \boldsymbol{\lambda}) = w_a(\mathbf{x}) \exp(f_a^+(\mathbf{x}, \boldsymbol{\lambda})), \quad f_a^+(\mathbf{x}, \boldsymbol{\lambda}) = -1 - \boldsymbol{\lambda} \cdot \mathbf{c}_a(\mathbf{x}), \quad (13a)$$

$$p_a^-(\mathbf{x}, \boldsymbol{\lambda}) = w_a(\mathbf{x}) \exp(f_a^-(\mathbf{x}, \boldsymbol{\lambda})), \quad f_a^-(\mathbf{x}, \boldsymbol{\lambda}) = -1 + \boldsymbol{\lambda} \cdot \mathbf{c}_a(\mathbf{x}). \quad (13b)$$

The Lagrangian dual function  $f : \mathbb{R}^6 \rightarrow \mathbb{R}$  is:

$$f(\boldsymbol{\lambda}) = \max_{\mathbf{p}^+, \mathbf{p}^- \in \mathbb{R}_{+}^{2n}} \mathcal{L}(\mathbf{p}^+, \mathbf{p}^-, \boldsymbol{\lambda}).$$

On substituting (12) and (13) in the above equation and simplifying, we obtain

$$\begin{aligned} f(\boldsymbol{\lambda}) := Z(\mathbf{x}, \boldsymbol{\lambda}) &= \lambda_0 + \sum_{a=1}^{2n} w_a(\mathbf{x}) [\exp(f_a^+(\mathbf{x}, \boldsymbol{\lambda})) + \exp(f_a^-(\mathbf{x}, \boldsymbol{\lambda}))] \\ &= \lambda_0 + \sum_{a=1}^{2n} [p_a^+(\mathbf{x}, \boldsymbol{\lambda}) + p_a^-(\mathbf{x}, \boldsymbol{\lambda})], \end{aligned} \quad (14)$$

where  $Z(\mathbf{x}, \boldsymbol{\lambda}) : \Omega_e \times \mathbb{R}^6 \rightarrow \mathbb{R}$  is the *partition function*, and its explicit dependence on  $\mathbf{x}$  is indicated. Hence, the dual problem is:

$$\boldsymbol{\lambda}^*(\mathbf{x}) = \underset{\boldsymbol{\lambda} \in \mathbb{R}^6}{\operatorname{argmin}} Z(\mathbf{x}, \boldsymbol{\lambda}), \quad (15)$$

where  $\boldsymbol{\lambda}^*(\mathbf{x})$  is the converged solution for the Lagrange multiplier vector.

### 2.3. Numerical algorithm

The dual problem posed in (15) is an unconstrained convex optimization problem, and is solved using Newton's method. The main steps in Newton's method to compute the shape functions  $\phi_a(\mathbf{x})$  for any  $\mathbf{x} \in \Omega_e$  are as follows:

1. Inputs: Vertices of polygon  $\Omega_e$  in counter-clockwise orientation, and a point  $\mathbf{x} \in \Omega_e$ . Rescale diameter of the polygon to  $\mathcal{O}(1)$  for the computations. Also functions to compute the nodal prior weight function  $w_a(\mathbf{x})$  and its gradient are available.
2. Set the nodal (vertex and mid-edge) coordinates  $\mathbf{x}_a$  ( $a = 1, \dots, 2n$ ), and use (6) to store  $\mathbf{c}_a$ .
3. Start with iteration counter  $k = 0$ , and the initial guess  $\boldsymbol{\lambda}^0 = \mathbf{0}$ . Let  $\epsilon$  be the convergence tolerance, where  $\epsilon$  in the range  $10^{-5}$  to  $10^{-15}$  is suitable.
4. Compute  $\mathbf{g}^k = \nabla_{\boldsymbol{\lambda}} Z^k$  and  $\mathbf{H}^k = \nabla_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} Z^k$ , which are the gradient and Hessian of the *partition function*  $Z(\mathbf{x}, \boldsymbol{\lambda})$ , respectively. The gradient  $\mathbf{g}^k = -\sum_{a=1}^{2n} \phi_a^k \mathbf{c}_a + \mathbf{q}$ , and the Hessian  $\mathbf{H}^k = \sum_{a=1}^{2n} [p_a^{+k} + p_a^{-k}] \mathbf{c}_a \otimes \mathbf{c}_a$ .
5. Determine Newton search direction  $\Delta \boldsymbol{\lambda}^k = -(\mathbf{H}^k)^{-1} \mathbf{g}^k$ .
6. Update:  $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha \Delta \boldsymbol{\lambda}^k$ , where  $\alpha$  is the step size. If  $\|\mathbf{g}^k\| > 10^{-4}$ , then  $\alpha$  is determined using a line search algorithm [50]; otherwise  $\alpha$  is set to unity.
7. Check convergence: if  $\|\mathbf{g}^{k+1}\| > \epsilon$ , then increment the iteration counter  $k$  and goto 3, else continue.
8. Set  $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^{k+1}$  and compute  $p_a^{+*}(\mathbf{x})$  and  $p_a^{-*}(\mathbf{x})$  using (13) and then the shape function  $\phi_a^*(\mathbf{x}) = p_a^{+*}(\mathbf{x}) - p_a^{-*}(\mathbf{x})$  ( $a = 1, \dots, 2n$ ).

The unconstrained convex optimization problem is efficiently solved using Newton's method with line search; only 3 to 7 iterations are needed to obtain an accuracy of  $10^{-15}$  on planar polygons. For a Gaussian nodal prior weight function, the implicit function theorem is invoked to prove the smoothness of non-negative [37] and signed shape functions [1]. Following the analysis of Sukumar and Wets [49], the continuity of signed shape functions for a  $C^k(\Omega_e)$  ( $k \geq 0$ ) nodal prior weight function is readily established.

## 2.4. Computation of the gradient of the shape functions

As indicated in Section 2.3, once the Lagrange multiplier vector is determined, the shape function  $\phi_a^*(\mathbf{x})$  is:

$$\begin{aligned}\phi_a^*(\mathbf{x}) &= p_a^{+*}(\mathbf{x}) - p_a^{-*}(\mathbf{x}) \\ &= w_a(\mathbf{x}) \exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - w_a(\mathbf{x}) \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)),\end{aligned}\tag{16}$$

where (13) is used. Therefore, the gradient of  $\phi_a^*(\mathbf{x})$  is:

$$\begin{aligned}\nabla \phi_a^*(\mathbf{x}) &= \left\{ \exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)) \right\} \nabla w_a(\mathbf{x}) \\ &\quad + w_a(\mathbf{x}) \left\{ D[\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*))] - D[\exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \right\},\end{aligned}\tag{17}$$

where  $D[\cdot]$  is the total derivative of its argument,  $\nabla w_a(\mathbf{x})$  is known and  $D[\exp(f_a^{\pm*}(\mathbf{x}, \boldsymbol{\lambda}^*))]$  needs to be determined. We can write

$$\begin{aligned}D[\exp(f_a^{\pm*}(\mathbf{x}, \boldsymbol{\lambda}^*))] &= \exp(f_a^{\pm*}(\mathbf{x}, \boldsymbol{\lambda}^*)) \nabla(f_a^{\pm*}(\mathbf{x}, \boldsymbol{\lambda}^*)) \\ &= \exp(f_a^{\pm*}(\mathbf{x}, \boldsymbol{\lambda}^*)) [\mp \mathbf{c}_a(\mathbf{x}) \cdot \nabla \boldsymbol{\lambda}^* \mp \boldsymbol{\lambda}^* \cdot \nabla \mathbf{c}_a(\mathbf{x})],\end{aligned}$$

where (13) is used to arrive at the second equality. Therefore,

$$\begin{aligned}w_a(\mathbf{x}) \left\{ D[\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*))] - D[\exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \right\} \\ = -[p_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*) + p_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)] [\mathbf{c}_a(\mathbf{x}) \cdot \nabla \boldsymbol{\lambda}^* + \boldsymbol{\lambda}^* \cdot \nabla \mathbf{c}_a(\mathbf{x})].\end{aligned}\tag{18}$$

In the above equation, the matrix-representation for  $\nabla \mathbf{c}_a(\mathbf{x})$  is obtained from (6):

$$\mathbf{D}_a(\mathbf{x}) := \nabla \mathbf{c}_a(\mathbf{x}) = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ -2(x_a - x) & 0 \\ 0 & -2(y_a - y) \\ -(y_a - y) & -(x_a - x) \end{bmatrix},\tag{19}$$

and hence we need to determine  $\nabla \boldsymbol{\lambda}^*$  in (18) to complete the derivation. To this end, using (14) and letting  $\mathbf{r} := \nabla_{\boldsymbol{\lambda}} Z(\mathbf{x}, \boldsymbol{\lambda})$ , we have

$$\begin{aligned}\mathbf{r}^* &= \mathbf{q} - \sum_{a=1}^{2n} [w_a(\mathbf{x}) \exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - w_a(\mathbf{x}) \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \mathbf{c}_a(\mathbf{x}) \\ &= \mathbf{q} - \sum_{a=1}^{2n} [p_a^{+*}(\mathbf{x}) - p_a^{-*}(\mathbf{x})] \mathbf{c}_a(\mathbf{x}) = \mathbf{0},\end{aligned}\quad (20)$$

since the constraint (7) is satisfied at convergence. Therefore,

$$D\mathbf{r}^* = \nabla_{\mathbf{x}}\mathbf{r}^* + (\nabla_{\boldsymbol{\lambda}}\mathbf{r})^* \cdot \nabla \boldsymbol{\lambda}^* = \mathbf{0}. \quad (21)$$

Let  $\mathbf{H}^* = (\nabla_{\boldsymbol{\lambda}}\mathbf{r})^*$  denote the Hessian of the partition function. Applying the operator  $\nabla_{\boldsymbol{\lambda}}$  to (20) leads to

$$\mathbf{H}^* = \sum_{a=1}^{2n} [p_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*) + p_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)] \mathbf{c}_a \otimes \mathbf{c}_a. \quad (22)$$

Note that for any non-zero  $\mathbf{u} \in \mathbb{R}^6$ , it follows from (22) that  $\mathbf{u} \cdot \mathbf{H} \cdot \mathbf{u} \geq 0$  (positive semi-definite), since  $p_a^+(\mathbf{x}, \boldsymbol{\lambda})$  and  $p_a^-(\mathbf{x}, \boldsymbol{\lambda})$  are both non-negative. Furthermore, in  $\Omega_e$  (open set),  $p_a^+ + p_a^-$  is positive from (13), and strict convexity of the partition function is realized [1]. However, unlike Reference [1] where strict convexity is also met on  $\partial\Omega_e$  with a Gaussian *prior*, here it is not so since if  $\mathbf{x}$  lies on a boundary edge, only three nodal prior weight functions  $w_a(\mathbf{x})$  are non-zero, and  $\mathbf{H}$  is singular.

Now, on applying  $\nabla_{\mathbf{x}}$  to (20), we obtain

$$\begin{aligned}\nabla_{\mathbf{x}}\mathbf{r}^* &= - \sum_{a=1}^{2n} [\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \mathbf{c}_a(\mathbf{x}) \otimes \nabla w_a(\mathbf{x}) \\ &\quad + \sum_{a=1}^{2n} \mathbf{c}_a(\mathbf{x}) \otimes w_a(\mathbf{x}) [\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) + \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \boldsymbol{\lambda}^* \cdot \mathbf{D}_a(\mathbf{x}) \\ &\quad - \sum_{a=1}^{2n} w_a(\mathbf{x}) [\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \mathbf{D}_a(\mathbf{x}),\end{aligned}$$

and denoting  $\mathbf{A}^* := \nabla_{\mathbf{x}} \mathbf{r}^*$ , we have

$$\begin{aligned}\mathbf{A}^* := \nabla_{\mathbf{x}} \mathbf{r}^* &= - \sum_{a=1}^{2n} [\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \mathbf{c}_a(\mathbf{x}) \otimes \nabla w_a(\mathbf{x}) \\ &\quad + \sum_{a=1}^{2n} \mathbf{c}_a(\mathbf{x}) [p_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*) + p_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)] \otimes (\boldsymbol{\lambda}^* \cdot \mathbf{D}_a(\mathbf{x})) \\ &\quad - \sum_{a=1}^{2n} \phi_a^*(\mathbf{x}, \boldsymbol{\lambda}^*) \mathbf{D}_a(\mathbf{x}).\end{aligned}\tag{23}$$

On substituting (22) and (23) in (21), we obtain

$$\nabla \boldsymbol{\lambda}^* = -(\mathbf{H}^*)^{-1} \cdot \mathbf{A}^*. \tag{24}$$

Now, using (18), (19), and (24), we can write (17) as

$$\begin{aligned}\nabla \phi_a^*(\mathbf{x}) &= [\exp(f_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*)) - \exp(f_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*))] \nabla w_a(\mathbf{x}) + \\ &\quad [p_a^{+*}(\mathbf{x}, \boldsymbol{\lambda}^*) + p_a^{-*}(\mathbf{x}, \boldsymbol{\lambda}^*)] [\mathbf{c}_a(\mathbf{x}) \cdot (\mathbf{H}^*)^{-1} \cdot \mathbf{A}^* - \boldsymbol{\lambda}^* \cdot \mathbf{D}_a(\mathbf{x})]\end{aligned}\tag{25}$$

which is the final result for the gradient of  $\phi_a^*(\mathbf{x})$ . It is readily verified that this expression for  $\nabla \phi_a^*(\mathbf{x})$  satisfies:  $\sum_{a=1}^{2n} [\mathbf{c}_a(\mathbf{x}) \otimes \nabla \phi_a^*(\mathbf{x}) + \phi_a^*(\mathbf{x}) \mathbf{D}_a] = \mathbf{0}$ , which is the gradient of (7).

### 3. Voronoi Mesh Generation Based on Maximal Poisson-Sampling

Compared to mesh generation for triangular meshes, polygonal mesh generation is still in its infancy. Among the Voronoi mesh generation techniques, the contributions of Du and Gunzburger [51], Yip et al. [27], Sieger et al. [28], Talischi et al. [29], and the recent work of Ebeida et al. [30–32] are significant. To generate Voronoi meshes with uniform random sampling, it is desirable to insert points without bias within a domain. This leads to a randomly closed-packed structure (RCP), and is referred to as maximal Poisson-disk sampling (MPS) or dart-throwing in computer graphics. Poisson-disk sampling is a process that distributes uniform random points in a domain in

$\mathbb{R}^d$  with no two points being closer than a distance  $\rho$ . Recently, Ebeida and co-workers [32] proposed an efficient algorithm, which is provably maximal, bias-free unlike centroidal Voronoi tessellation [51, 52]), has a finite stopping-time, and is more efficient than previous algorithms. An application to uniform Voronoi mesh-generation is also described [30].

We have developed a Voronoi mesh generator based on the MPS algorithm presented in Reference [31]. The built-in Matlab function `voronoin` and Matlab functions provided in PolyTop [29] for building the mesh-connectivity are used to create the polygonal meshes. The essentials of the algorithm are now described for the unit square. Given a  $\rho$ , the unit square is divided into square (base) cells of side length  $s_0 = \rho/\sqrt{2}$  ( $s_0 = \rho/\sqrt{d}$  in  $\mathbb{R}^d$ ), so that at most only one point is assigned to each base cell. All these base cells are at level 0 and are either empty or have one point during any iteration. Let  $|\mathcal{C}|$  denote the number of cells that are active (empty). At the beginning of each iteration, we throw  $0.5|\mathcal{C}|$  number of darts uniformly on the grid, which results in some *hits*. Those base cells that are *hit* are assigned the point, and they are now deemed inactive. For the next iteration, we refine the background grid by subdividing all uncovered (active) cells into four subcells (we discard subcells that are completely covered by the disks). Now, we are at level 1, and repeat the foregoing process. By retaining only squares of the same size at any iteration, it allows the generation of an unbiased candidate point in constant time and leads to efficient memory usage and run-time. A sample Voronoi mesh is illustrated in Fig. 3. For the point-distribution (Voronoi generators) in a unit square, the essential ingredients in the algorithm (active cells, inactive cells, and circles) at every iteration are shown in Fig. 4.

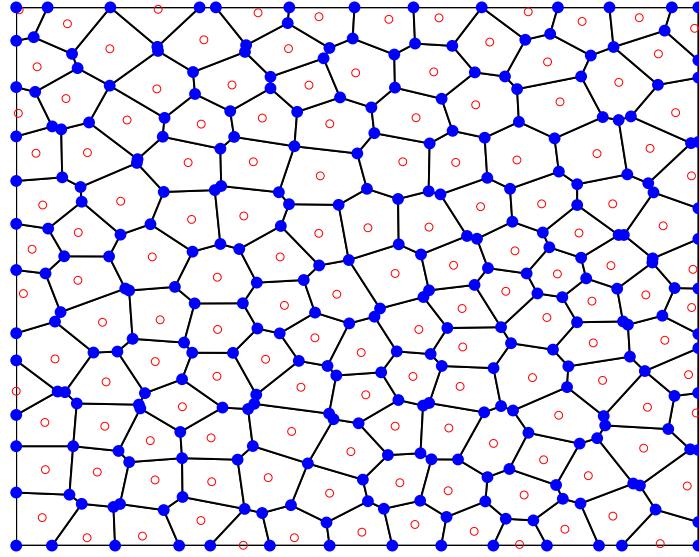


Fig. 3: Voronoi mesh using maximal Poisson-sampling algorithm. Nodes are indicated by filled circles and the seed points are shown as open circles.

#### 4. Numerical integration of the weak form integrals

As in meshfree methods [53], since polygonal shape functions are non-polynomial, accurate and efficient numerical integration is also a pertinent issue in polygonal finite element methods. Use of standard polynomial-precise quadrature on triangles [54] proves to be insufficient to pass the patch test to machine precision [7]. To satisfy the patch test in nodally-integrated meshfree methods, a *smoothed or corrected strain operator* [55, 56] has been proposed. Even though the patch test is met, nodal integration requires additional stabilization to prevent the presence of spurious modes [57]. The corrected strain operator is also used in the smoothed finite element method on polygonal meshes [58, 59]. Most of the aforementioned studies have focused on a modified strain operator for linearly complete meshfree approximations. Recently, Duan et al. [38] proposed a cell-based integration scheme that is suitable for quadratically complete meshfree methods; herein, we tailor this

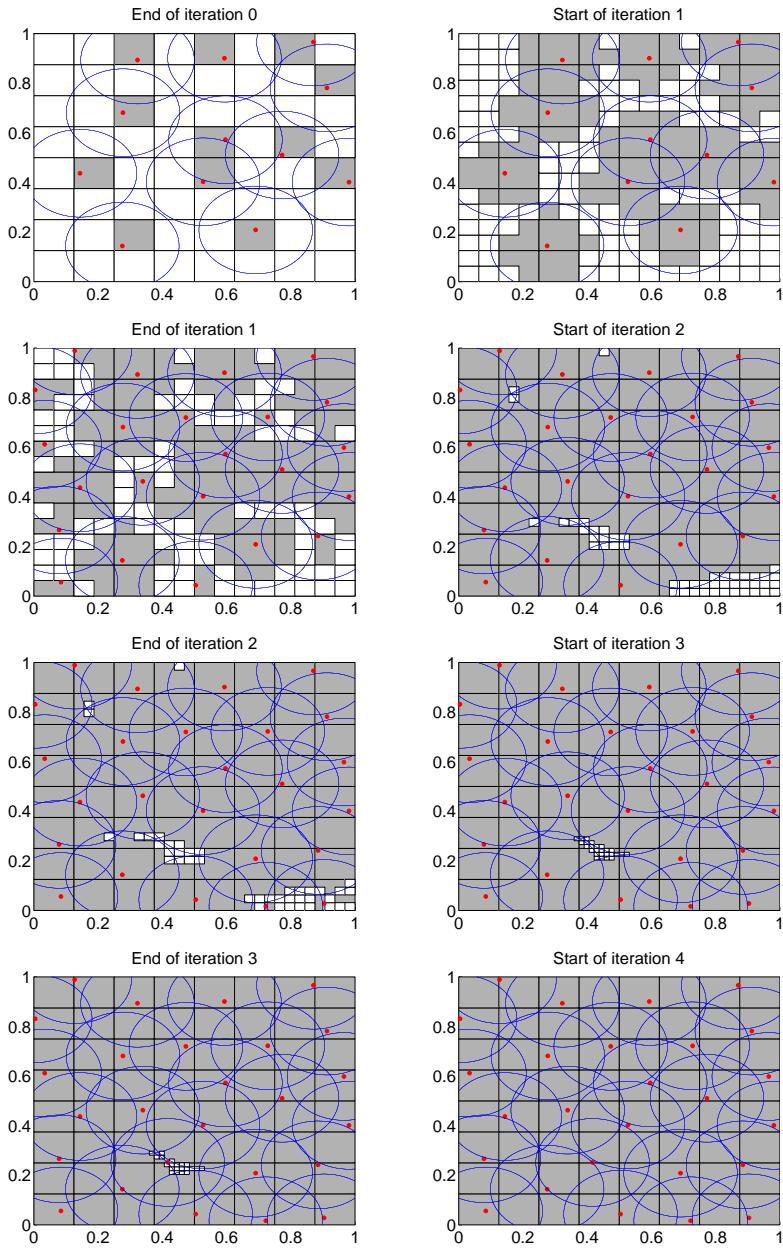


Fig. 4: Maximal Poisson-sampling algorithm [31]. Light cells are active and dark (grey) cells are inactive. At the beginning of iteration 4, the sampling is maximal since there aren't any active cells left. At any iteration, all active cells are always at the same *level*. This simplifies the data structures, reduces the memory requirements, and eases the numerical implementation.

integration scheme for the quadratic serendipity finite element method to ensure that the patch test is satisfied.

Consider the strong form for the Poisson equation:

$$-\nabla^2 u(\mathbf{x}) = f(\mathbf{x}) \text{ in } \Omega = (0, 1)^2, \quad (26a)$$

$$u(\mathbf{x}) = g(\mathbf{x}) \text{ on } \partial\Omega, \quad (26b)$$

whose weak form is:

$$a(u, w) = \ell(w) \quad \forall w \in H_0^1, \quad a(u, w) = \int_{\Omega} \nabla u \cdot \nabla w d\Omega, \quad \ell(w) = \int_{\Omega} f w d\Omega. \quad (27)$$

For a Galerkin method with quadratic max-ent shape functions, we have the following linear system:

$$\mathbf{Kd} = \mathbf{f}, \quad \mathbf{K}_{ab} = \int_{\Omega} \nabla \phi_a \cdot \nabla \phi_b d\Omega, \quad \mathbf{f}_a = \int_{\Omega} f \phi_a d\Omega, \quad (28)$$

where  $\mathbf{K}$  and  $\mathbf{f}$  are formed via element-level assembly procedures.

In deriving the weak form in (27), we multiply (26a) by test functions  $\phi_a$  and integrate over the domain. On invoking the divergence theorem, we obtain

$$-\int_{\Omega} \phi_a u_{,ii} d\Omega = -\int_{\partial\Omega} \phi_a u_{,i} n_i dS + \int_{\Omega} \phi_{a,i} u_{,i} d\Omega + \int_{\Omega} \phi_a f d\Omega,$$

or,

$$\int_{\Omega} \phi_{a,i} u_{,i} d\Omega = \int_{\partial\Omega} \phi_a u_{,i} n_i dS - \int_{\Omega} \phi_a u_{,ii} d\Omega - \int_{\Omega} \phi_a f d\Omega.$$

For the quadratic patch test,  $f = 0$ , and  $u_{,1}$  and  $u_{,2}$  are affine functions. Hence, the above equation becomes

$$\int_{\Omega} \phi_{a,i} \{1 \ x_1 \ x_2\}^T d\Omega = \int_{\partial\Omega} \phi_a \{1 \ x_1 \ x_2\}^T n_i dS - \int_{\Omega} \phi_a \{0 \ \delta_{1i} \ \delta_{2i}\}^T d\Omega \quad (i = 1, 2),$$

and therefore we can write the conditions (integration constraints) that must

be met by the shape function derivatives as:

$$\int_{\Omega} \phi_{a,1} d\Omega = \int_{\partial\Omega} \phi_a n_1 dS, \quad (29a)$$

$$\int_{\Omega} \phi_{a,1} x_1 d\Omega = \int_{\partial\Omega} \phi_a x_1 n_1 dS - \int_{\Omega} \phi_a d\Omega, \quad (29b)$$

$$\int_{\Omega} \phi_{a,1} x_2 d\Omega = \int_{\partial\Omega} \phi_a x_2 n_1 dS, \quad (29c)$$

which applies to  $\phi_{a,1}$ , and similarly,

$$\int_{\Omega} \phi_{a,2} d\Omega = \int_{\partial\Omega} \phi_a n_2 dS, \quad (29d)$$

$$\int_{\Omega} \phi_{a,2} x_1 d\Omega = \int_{\partial\Omega} \phi_a x_1 n_2 dS, \quad (29e)$$

$$\int_{\Omega} \phi_{a,2} x_2 d\Omega = \int_{\partial\Omega} \phi_a x_2 n_2 dS - \int_{\Omega} \phi_a d\Omega \quad (29f)$$

must be met by  $\phi_{a,2}$ .

To obtain corrected shape function derivatives that satisfy (29), we divide each  $n$ -gon into  $n$  triangles, and use a 3-point quadrature rule [54] in the interior of each subtriangle and 2-point Gauss quadrature along each edge of the subtriangle (Fig. 5). The 3-point integration rule has the weight  $A_e^\Delta/3$  for each integration point, where  $A_e^\Delta$  is the area of the subtriangle. Hence, the discrete form of (29) over each subtriangle is:

$$\frac{A_e^\Delta}{3} \begin{bmatrix} 1 & 1 & 1 \\ x_1^\ell & x_1^m & x_1^n \\ x_2^\ell & x_2^m & x_2^n \end{bmatrix} \begin{Bmatrix} \phi_{a,i}(\mathbf{x}_\ell) \\ \phi_{a,i}(\mathbf{x}_m) \\ \phi_{a,i}(\mathbf{x}_n) \end{Bmatrix} = \begin{Bmatrix} f_i(\mathbf{x}_\ell) \\ f_i(\mathbf{x}_m) \\ f_i(\mathbf{x}_n) \end{Bmatrix}, \quad (30)$$

where  $\{\mathbf{x}_\ell, \mathbf{x}_m, \mathbf{x}_n\}$  are the three integration points (with equal weights) within the subtriangle, and this choice ensures that the matrix on the left-hand side is invertible [38]. The use of numerical quadrature to compute the integrals on the right-hand side of (29) yields  $f_i(\cdot)$  in (30). Once these corrected derivatives are evaluated at the  $3n$  points of each polygon, they are used to compute the element stiffness matrix, which is assembled to

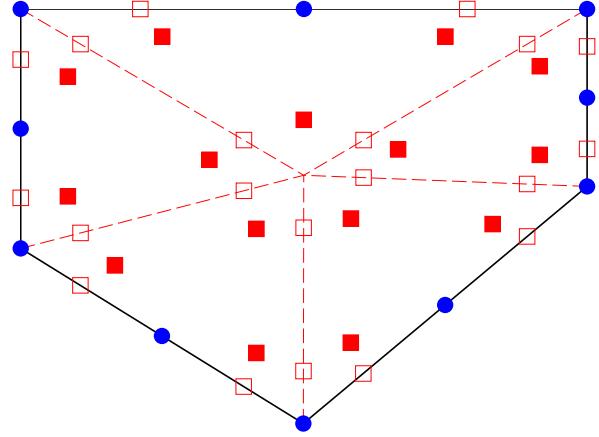


Fig. 5: Numerical integration scheme in a polygonal element. Nodes of the element are shown as filled circles. The filled squares are integration points within each subtriangle, and the open squares are Gauss points on the edges.

form the stiffness matrix  $\mathbf{K}$  in (28). For a Poisson problem with a non-zero  $f(\mathbf{x})$ , a standard 3-point integration rule [54] within each subtriangle is used to compute the element force vector, which is assembled to form the force vector  $\mathbf{f}$  in (28).

## 5. Numerical Examples

First, the linear independence (unisolvency of the degrees of freedom) of the shape functions and the  $C^0$ -conformity of the max-ent approximation are established. Then, shape function plots on convex and nonconvex polygons are illustrated to reveal their properties. Finally, Galerkin solutions for two Poisson problems are presented to demonstrate the accuracy and convergence of the method. In particular, we demonstrate that the numerical integration scheme presented in Section 5 passes the quadratic patch test, and then show that optimal rates of convergence ( $L^2$  norm and  $H^1$  seminorm) of the max-ent approximation are obtained on uniform refinements of a mesh

consisting of self-similar trapezoids. On such meshes, it is well-known that quadratic serendipity isoparametric finite elements yield suboptimal rates of convergence [39–41]. The maximum-entropy code for Galerkin computations is developed in **Matlab** Version 7.14 (R2012a).

### 5.1. Linear independence of shape functions

As indicated in Section 2.1, for a point  $\mathbf{x}$  that lies on an edge  $a$  that contains nodes  $(a, a+n, a+1)$  (see Fig. 1), the prior weight functions given in (9) are such that only  $w_a(\mathbf{x})$ ,  $w_{a+n}(\mathbf{x})$  and  $w_{a+1}(\mathbf{x})$  are non-zero. It follows from (16) and the constraints in (7) that a unique solution is obtained with  $\phi_a(\mathbf{x})$ ,  $\phi_{a+n}(\mathbf{x})$  and  $\phi_{a+1}(\mathbf{x})$  as the only non-zero shape functions. If the  $a$ -th edge  $(\mathbf{x}_a, \mathbf{x}_{a+1})$  is parametrized by  $t \in [0, 1]$ , then

$$\phi_a(t) = (1-t)^2, \quad \phi_{a+n}(t) = 2t(1-t), \quad \phi_{a+1}(t) = t^2, \quad (31)$$

which are quadratic Bernstein basis functions [48]. Due to the choice of the *prior*, the shape functions herein retain the *facet-reducing property* of non-negative linearly complete max-ent shape functions [37].

To establish linear independence of the shape functions, we need to show that if  $\sum_{a=1}^{2n} \phi_a(\mathbf{x}) u_a = 0$  for all  $\mathbf{x} \in \bar{\Omega}_e$ , then it implies that  $u_a \equiv 0$  for all  $a$ .

**Proof.** We start with

$$\sum_{b=1}^{2n} \phi_b(\mathbf{x}) u_b = \sum_{b \in V} \phi_b(\mathbf{x}) u_b + \sum_{b \in E} \phi_b(\mathbf{x}) u_b = 0, \quad (32)$$

where  $V$  and  $E$  are the vertex and mid-edge nodal index sets, respectively, which are defined in Section 2. Using (31), we observe that for  $a \in V$ , the shape functions satisfy  $\phi_b(\mathbf{x}_a) = 0$  for  $b \neq a$  and  $\phi_a(\mathbf{x}_a) = 1$ . Choosing  $\mathbf{x} = \mathbf{x}_a$  in (32) leads to  $u_a = 0$  for all  $a \in V$ , and (32) becomes

$$\sum_{b \in E} \phi_b(\mathbf{x}) u_b = 0. \quad (33)$$

Now, for  $a, b \in E$ ,  $\phi_b(\mathbf{x}_a) = 0$  for  $b \neq a$  and  $\phi_a(\mathbf{x}_a) = 1/2$  from (31). Choosing  $\mathbf{x} = \mathbf{x}_a$  in (33) results in  $u_a = 0$  for all  $a \in E$ , and therefore  $u_a \equiv 0$  for all  $a \in V \cup E$ .  $\square$

### 5.2. $C^0$ approximation on polygonal meshes

As noted earlier, due to the choice of the *priors*, only  $\phi_a(\mathbf{x})$ ,  $\phi_{a+n}(\mathbf{x})$ , and  $\phi_{a+1}(\mathbf{x})$  are non-zero along edge  $a$ , and we recover the facet-reducing property of convex approximations [37]. An edge (three nodes on an edge) is common to two elements in a polygonal mesh, and due to the facet-reducing property of the shape functions, the approximation is  $C^0$ -continuous on any edge, and therefore the max-ent approximation is conforming on polygonal meshes.

### 5.3. Plots of shape functions

For the purpose of plotting the shape functions, each  $n$ -gon is divided into  $n$  triangles, and a tensor-product  $m \times m$  grid is used on each triangle. The polygonal elements that we consider are shown in Fig. 6. The nodal prior weight functions are illustrated in Fig. 2 for two nodes in the unit square (Fig. 6a). The average number of iterations needed in Newton's algorithm to compute the shape functions are listed in Table 1. The shape function plots are presented in Fig. 7. For the shape function plots, we use  $m = 25$  and a tolerance of  $\epsilon = 10^{-10}$  in the Newton iterations. Use of a uniform or Gaussian *prior* leads to non-conformity on polygonal meshes. To illustrate this, a nodal shape function is plotted in Fig. 7a for a Gaussian *prior* ( $\beta = 1$ ). Note that  $\phi_2(\mathbf{x})$  is non-zero on edges that do not contain node 2, which would prevent conformity on a polygonal mesh. In Figures 7b and 7c, the edge-based nodal prior weight function defined in (8a) and (9) is used, and the facet-reducing property of the shape functions is observed. In addition, the minimum value of the shape functions is  $\mathcal{O}(10^{-3})$ , whereas

with quadratic serendipity isoparametric elements, the corresponding minimum value is  $-0.037$ . In all the plots that appear hereafter, the nodal prior weight function defined via (8a) and (9) is adopted in the numerical computations. Shape function plots for the pentagon, a saw-tooth (nonconvex) polygon, and a  $L$ -shaped (nonconvex) polygon appear in Figures 7d–7f, Figures 7g–7i, and Figures 7j–7l, respectively. We observe that along the edge of a polygon, the maximum-entropy shape functions are identical to Bernstein basis functions given in (31). In Fig. 8, the derivatives of the shape functions are depicted ( $m = 50$  is used).

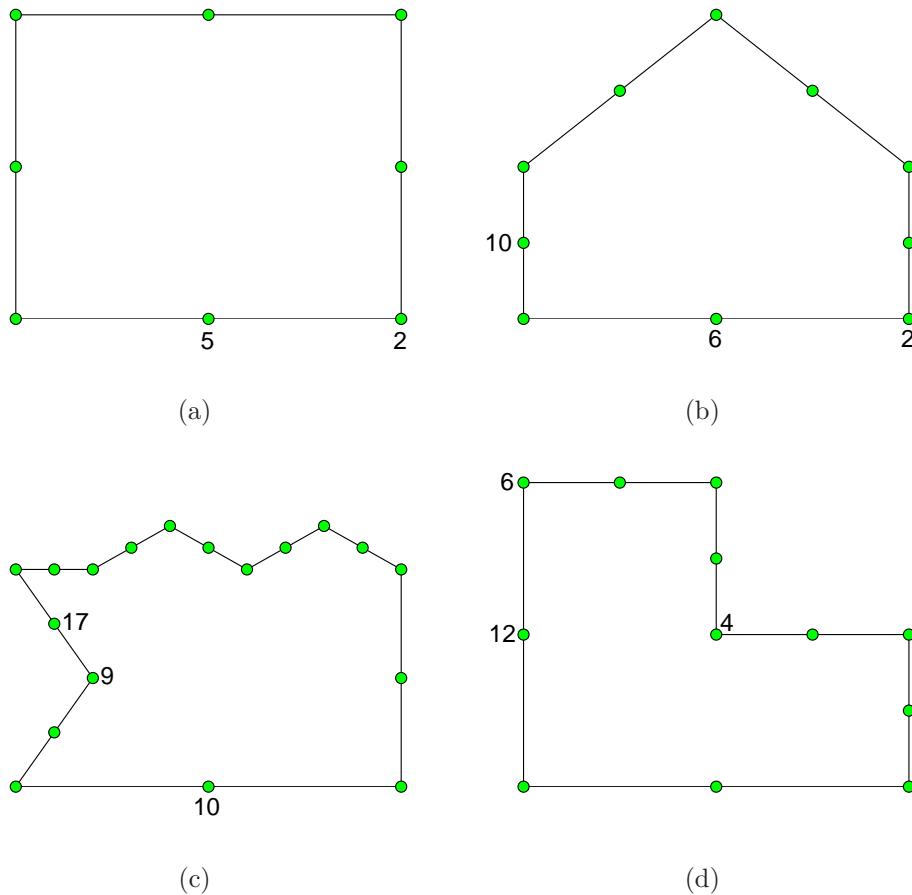
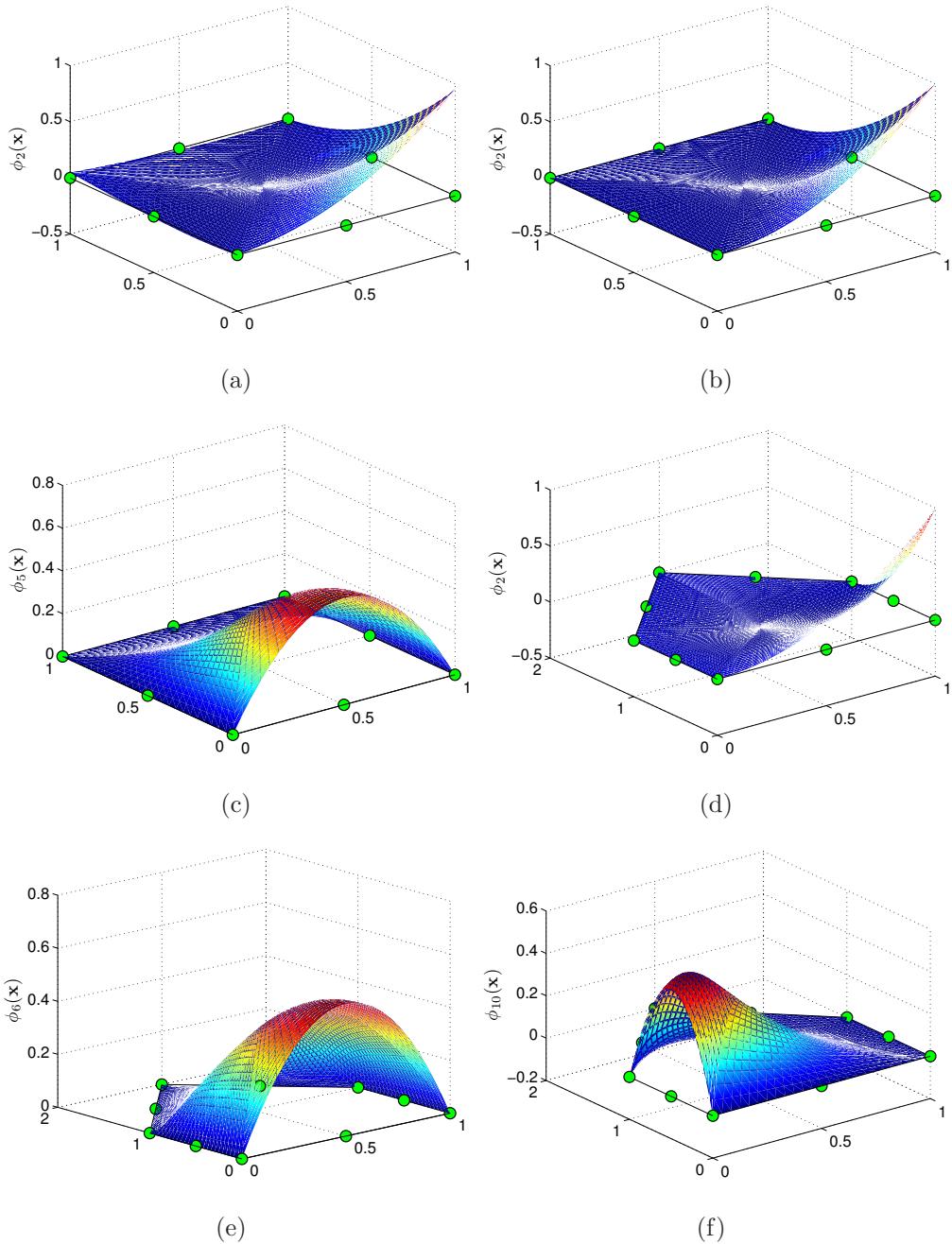


Fig. 6: Quadratic serendipity polygonal elements. (a) square; (b) pentagon; (c) saw-tooth; and (d)  $L$ -shaped.



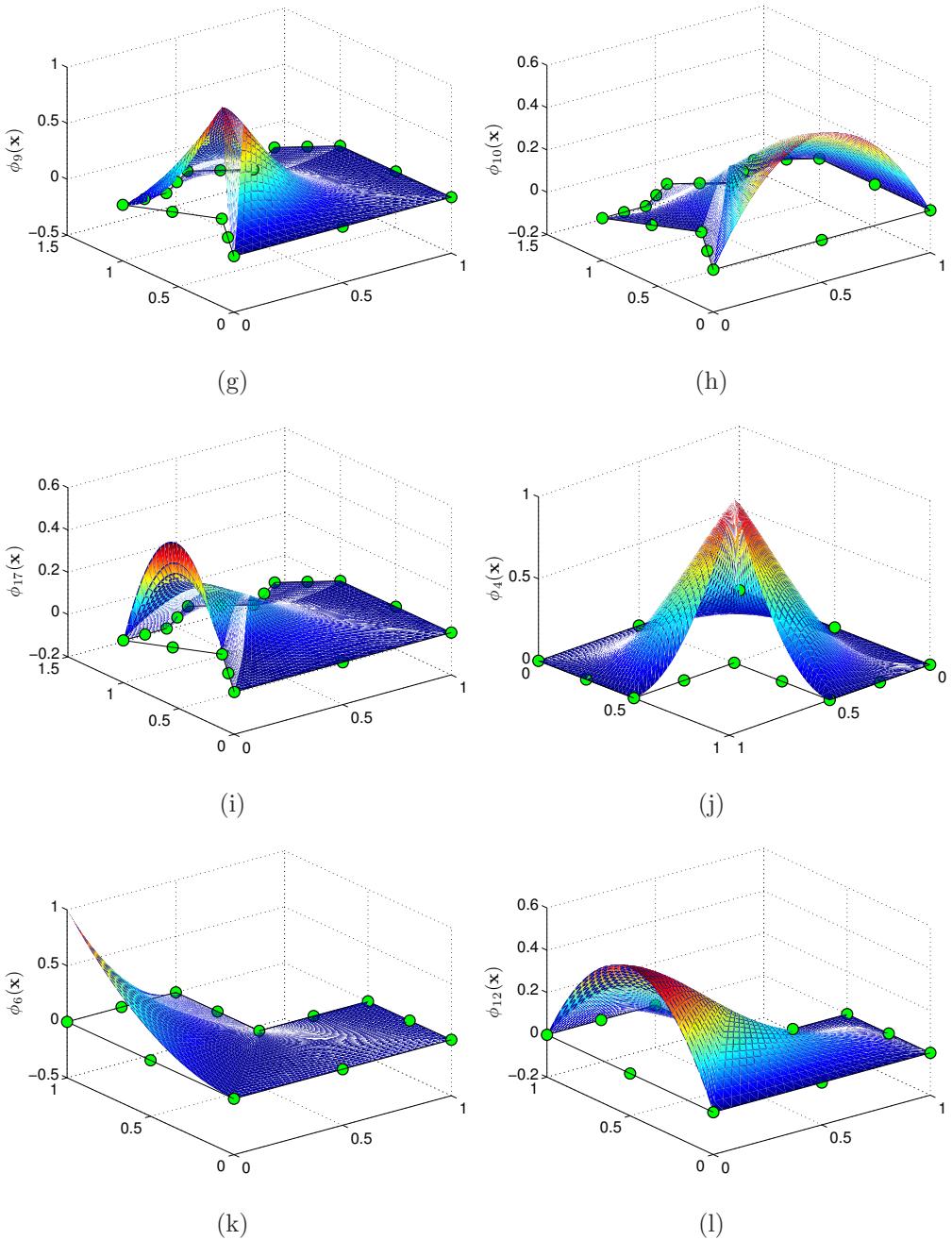


Fig. 7: Plots of max-ent shape functions. Square: (a)  $\phi_2(\mathbf{x})$  (Gaussian prior), (b)  $\phi_2(\mathbf{x})$  (edge-based nodal prior), (c)  $\phi_5(\mathbf{x})$  (edge-based nodal prior). Pentagon: (d)  $\phi_2(\mathbf{x})$ , (e)  $\phi_6(\mathbf{x})$ , (f)  $\phi_{10}(\mathbf{x})$ . Nonconvex element: (g)  $\phi_9(\mathbf{x})$ , (h)  $\phi_{10}(\mathbf{x})$ , (i)  $\phi_{17}(\mathbf{x})$ . L-shaped element: (j)  $\phi_4(\mathbf{x})$ , (k)  $\phi_6(\mathbf{x})$ , (l)  $\phi_{12}(\mathbf{x})$ .

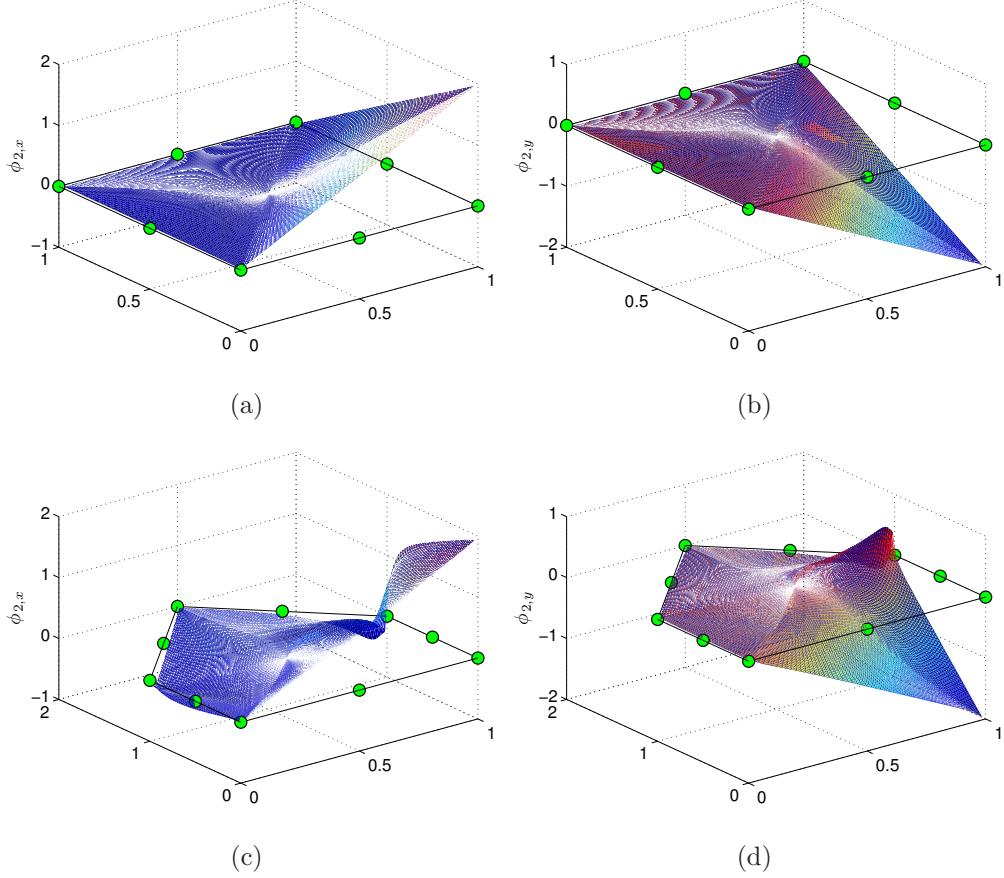


Fig. 8: Derivatives of max-ent shape functions. Square: (a)  $\partial\phi_2(\mathbf{x})/\partial x$  and (b)  $\partial\phi_2(\mathbf{x})/\partial y$ . Pentagon: (c)  $\partial\phi_2(\mathbf{x})/\partial x$  and (d)  $\partial\phi_2(\mathbf{x})/\partial y$ .

The maximum-entropy shape functions are also well-defined on weakly simple polygons. To illustrate this, we consider the polygons shown in Figures 9a, 9d, and 9g. In Fig. 9a, three consecutive vertices are collinear on the bottom edge, whereas in Fig. 9d, five consecutive vertices are collinear on the bottom edge. In Fig. 9g, three consecutive vertices are collinear on both the left and right edges. A few shape function plots are depicted in Fig. 9, which reveal that the shape functions are well-behaved on edges with collinear vertices.

Finally, we approximate the bivariate polynomial  $u(\mathbf{x}) = 1 - 2x + 6y + x^2 - 3xy + 4y^2$  on the L-shaped polygon. The error in  $u$  is presented in Fig. 10,

Table 1: Efficiency of Newton's method for shape function computations.

Element	Tolerance	Average number of iterations
Square	$10^{-5}$	3.5
	$10^{-10}$	4.5
	$10^{-15}$	5.1
Pentagon	$10^{-5}$	4.1
	$10^{-10}$	5.1
	$10^{-15}$	5.8
Saw-tooth	$10^{-5}$	4.1
	$10^{-10}$	5.1
	$10^{-15}$	5.8
<i>L</i> -shaped	$10^{-5}$	2.6
	$10^{-10}$	3.2
	$10^{-15}$	3.6

which verifies the quadratic completeness of the max-ent approximation.

#### 5.4. Quadratic patch test

Consider the Laplace equation:  $\nabla^2 u(\mathbf{x}) = 0$  in  $\Omega = (0, 1)^2$ , with essential boundary condition  $g(\mathbf{x}) = 1 - x + 5y - 2xy - 4x^2 + 4y^2$  imposed on  $\partial\Omega$ . The exact solution is:  $u(\mathbf{x}) = g(\mathbf{x})$ . For the patch test, ukufive meshes are considered (see Fig. 11), and the numerical integration scheme presented in Section 4 is used. For the patch test and for the convergence studies (Section 5.5), a tolerance of  $\epsilon = 10^{-14}$  is used in the Newton iterations to evaluate the shape functions. In Table 2, the relative error in the  $L^2$  norm and the  $H^1$  seminorm are presented. We observe that the patch test is passed. The results reported in Table 2 are significantly more accurate than those realized in a meshfree method based on moving least squares approximants [38].

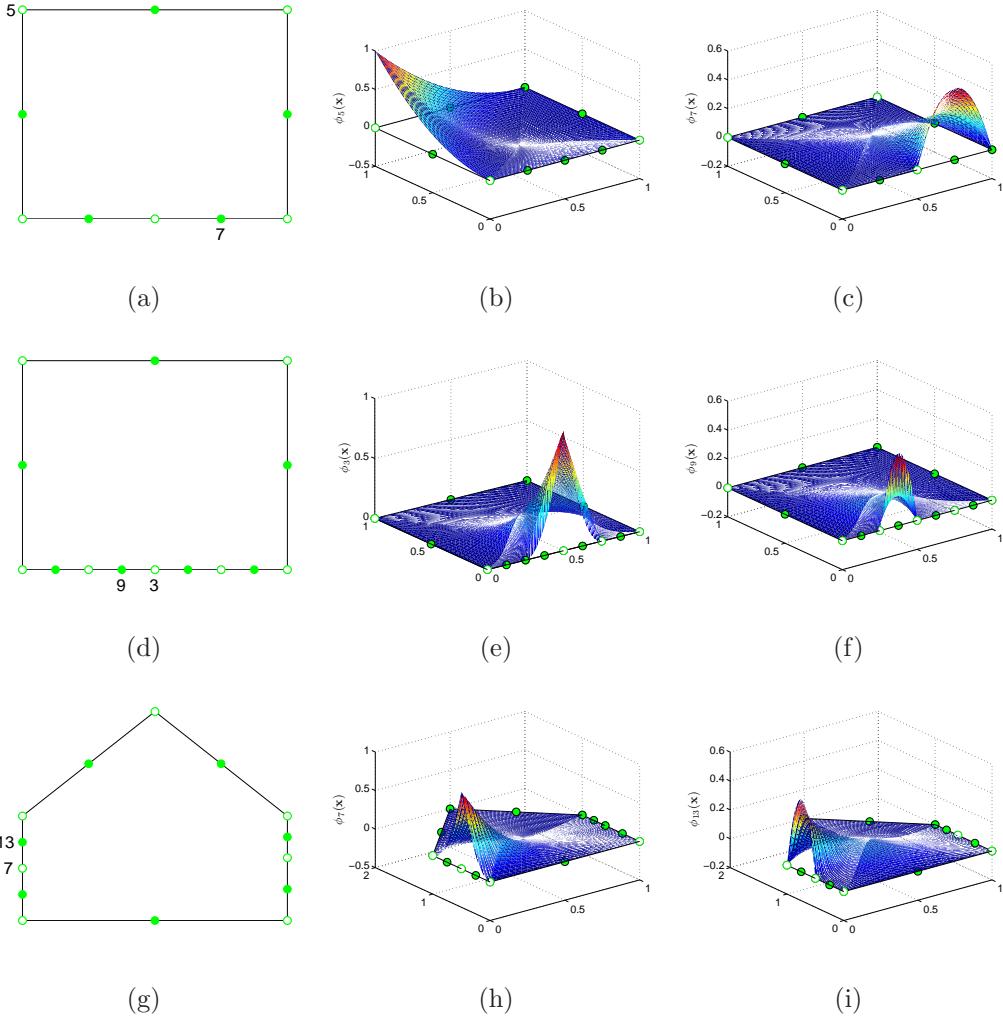


Fig. 9: Plots of max-ent shape functions on weakly simple polygons. The vertices of the polygon are shown as open circles. (a)–(c) Square with three collinear vertices on the bottom edge.  $\phi_5(\mathbf{x})$  and  $\phi_7(\mathbf{x})$  are shown; (d)–(f) Square with five collinear vertices on the bottom edge.  $\phi_3(\mathbf{x})$  and  $\phi_9(\mathbf{x})$  are shown; and (g)–(i) Pentagon with three collinear vertices on the left and right edges.  $\phi_7(\mathbf{x})$  and  $\phi_{13}(\mathbf{x})$  are shown.

### 5.5. Rate of convergence for Poisson problem

On quadrilateral meshes, 9-node biquadratic isoparametric finite elements deliver optimal convergence rates of 3 and 2 in the  $L^2$  norm and the  $H^1$

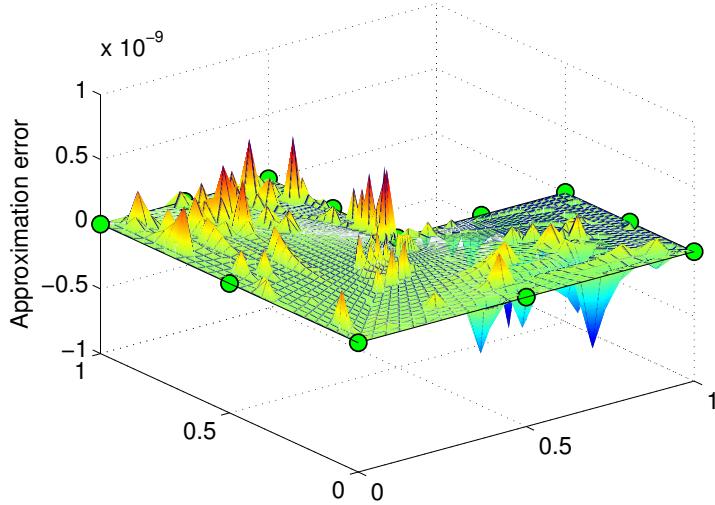


Fig. 10: Approximation error ( $u(\mathbf{x}) - u^h(\mathbf{x})$ ) over the polygon for the bivariate polynomial  $u(\mathbf{x}) = 1 - 2x + 6y + x^2 - 3xy + 4y^2$ .

Table 2: Relative error norms in the quadratic patch test.

Mesh	$\frac{\ u - u^h\ _{2,\Omega}}{\ u\ _{2,\Omega}}$	$\frac{ u - u^h _{1,\Omega}}{ u _{1,\Omega}}$
11a	$2.0 \times 10^{-15}$	$2.7 \times 10^{-14}$
11b	$2.6 \times 10^{-15}$	$2.5 \times 10^{-14}$
11c	$2.6 \times 10^{-15}$	$1.7 \times 10^{-14}$
11d	$2.9 \times 10^{-15}$	$3.8 \times 10^{-14}$
11e	$4.0 \times 10^{-15}$	$1.0 \times 10^{-13}$

seminorm, respectively. However, the rates of convergence for the quadratic serendipity isoparametric element are suboptimal on meshes with non-affinely mapped quadrilateral elements [39, 40]. Arnold and Awanou [41] provide a general definition for the space of serendipity finite elements in two- and higher-dimensions. The inability to represent arbitrary bivariate functions on general quadrilateral meshes, stiffening due to element distortion, and its poor performance for constrained media problems are some of the deficiencies of the serendipity element [60]. Lee and Bathe [61] perform a detailed study

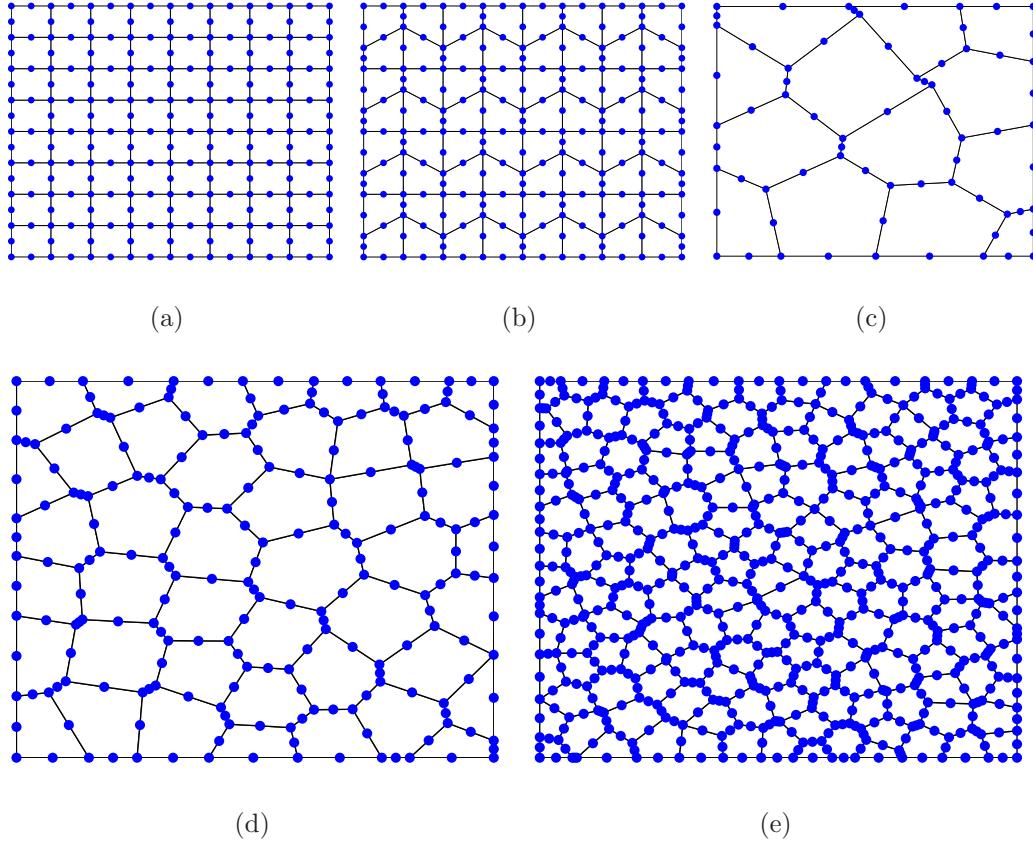


Fig. 11: Patch test. (a) Uniform mesh; (b) Self-similar trapezoidal mesh; (c) Polygonal mesh with nonconvex elements; and (d), (e) Convex polygonal meshes generated using maximal Poisson-sampling algorithm.

on the effects of element distortion using quadratic (Lagrangian and serendipity) isoparametric quadrilateral finite elements. The issue of mesh-distortion in isoparametric finite elements with potential remedies is discussed by Rajendran [62].

We adopt the self-similar trapezoidal meshes used in References [3, 40] to establish the rate of convergence of the quadratic max-ent approximation. Consider the Poisson problem in (26) with  $f(\mathbf{x})$  and  $g(\mathbf{x})$  chosen so that the exact solution is:  $u(\mathbf{x}) = x^3 + 5y^2 - 10y^3 + y^4$  [40]. The meshes considered in the convergence study are partitions of the domain into  $m \times m$  congruent

Table 3: Relative error norms for Poisson problem on trapezoidal meshes.

$m$	Number of nodes	$\frac{\ u - u^h\ _{2,\Omega}}{\ u\ _{2,\Omega}}$	Rate	$\frac{ u - u^h _{1,\Omega}}{ u _{1,\Omega}}$	Rate
2	21	$6.1 \times 10^{-2}$	—	$1.1 \times 10^{-1}$	—
4	65	$6.8 \times 10^{-3}$	3.16	$2.6 \times 10^{-2}$	2.02
8	225	$8.0 \times 10^{-4}$	3.10	$6.4 \times 10^{-3}$	2.03
16	833	$9.6 \times 10^{-5}$	3.05	$1.6 \times 10^{-3}$	2.02
32	3201	$1.2 \times 10^{-5}$	3.03	$4.0 \times 10^{-4}$	2.01
64	12545	$1.5 \times 10^{-6}$	3.01	$9.9 \times 10^{-5}$	2.00

trapezoids that are similar to the trapezoid with vertices at  $(0, 0)$ ,  $(1/2, 0)$ ,  $(1/2, 2/3)$ , and  $(0, 1/3)$  [40]. The meshes for  $m = 2, 4, 8, 16$  are depicted in Fig. 12. The relative error in the  $L^2$  norm and the  $H^1$  seminorm are listed in Table 3, and the convergence plots are illustrated in Fig. 13. The convergence rates of the max-ent polygonal finite element method are 3 and 2 in  $L^2$  and  $H^1$  Sobolev norms, respectively, which are in agreement with theory for the Poisson equation using second-order complete approximations.

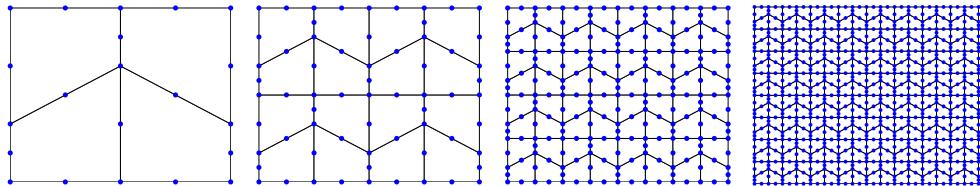


Fig. 12: Self-similar trapezoidal meshes for  $m = 2, 4, 8, 16$ .

Now, we perform the convergence study on polygonal meshes, and use five meshes that are constructed using  $\rho_k = \sqrt{2}/2^{2+k}$  ( $k = 1, 2, \dots, 5$ ) in the maximal Poisson-sampling algorithm. The meshes are depicted in Fig. 14. The relative error norms are listed in Table 4, and the convergence plots are presented in Fig. 15. We obtain numerical rates of convergence of 3 and 2 in the  $L^2(\Omega)$  norm and the  $H^1(\Omega)$  seminorm, respectively, which are in

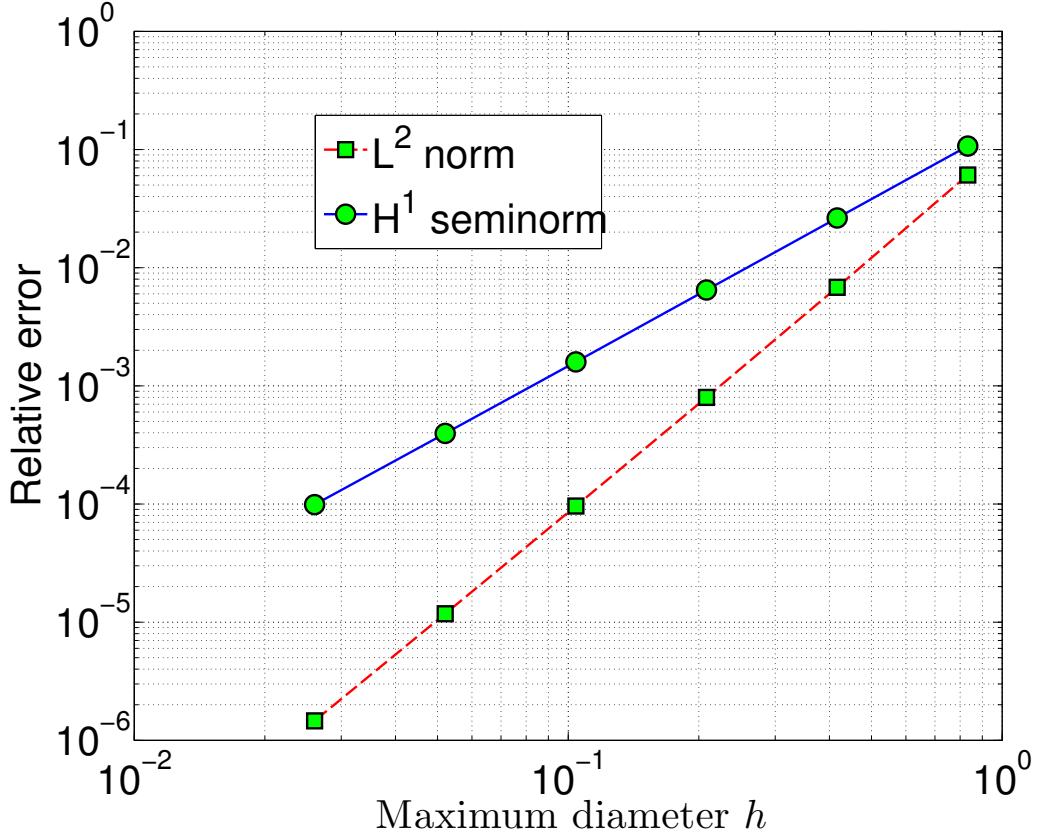


Fig. 13: Rates of convergence for Poisson problem on trapezoidal meshes.

agreement with theory.

## 6. Concluding Remarks

In this paper, we have provided a construction for  $C^0$  quadratic serendipity shape functions on planar (convex and nonconvex) polygons. To this end, we determined the shape functions via the maximization of a concave functional subject to linear constraints. We recast the objective functional of Bompade et al. [1] in terms of the relative entropy with nodal *prior* weight functions, and used the linear constraints for quadratic completeness proposed by Rand et al. [3]. The nodal prior weight functions were selected with the appropriate zero-set on the boundary edges [2] so that a

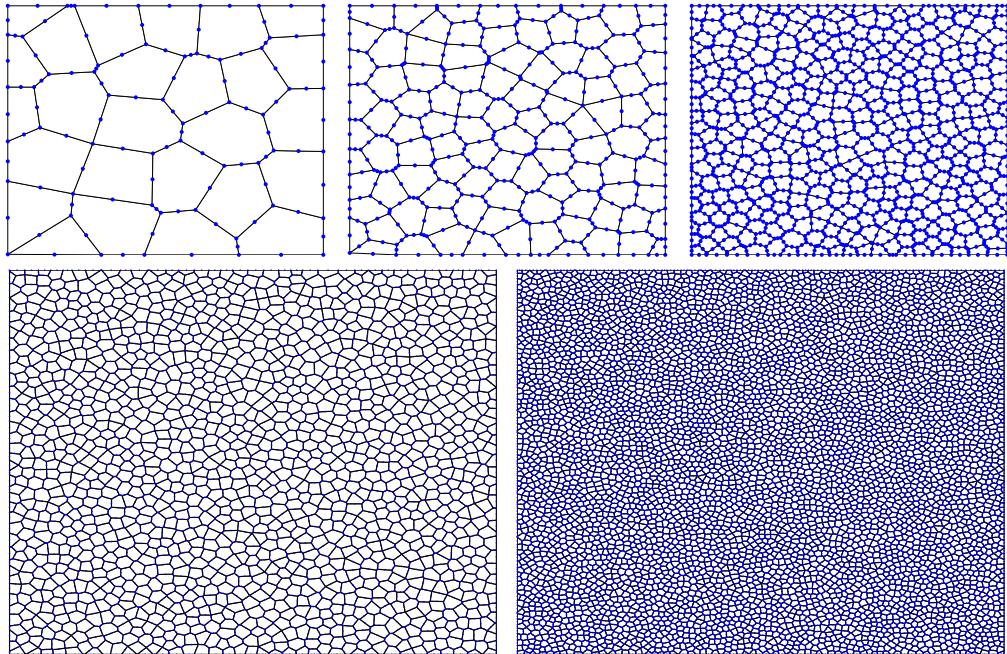


Fig. 14: Polygonal meshes used for the Poisson problem.

Table 4: Relative error norms for Poisson problem on polygonal meshes.

Number of nodes	$\frac{\ u - u^h\ _{2,\Omega}}{\ u\ _{2,\Omega}}$	Rate	$\frac{ u - u^h _{1,\Omega}}{ u _{1,\Omega}}$	Rate
127	$7.1 \times 10^{-3}$	–	$4.3 \times 10^{-2}$	–
485	$8.0 \times 10^{-4}$	3.17	$1.0 \times 10^{-2}$	2.05
1809	$1.1 \times 10^{-4}$	3.00	$2.7 \times 10^{-3}$	2.02
7100	$1.3 \times 10^{-5}$	3.03	$6.7 \times 10^{-4}$	2.07
27684	$1.9 \times 10^{-6}$	3.01	$1.7 \times 10^{-4}$	2.07

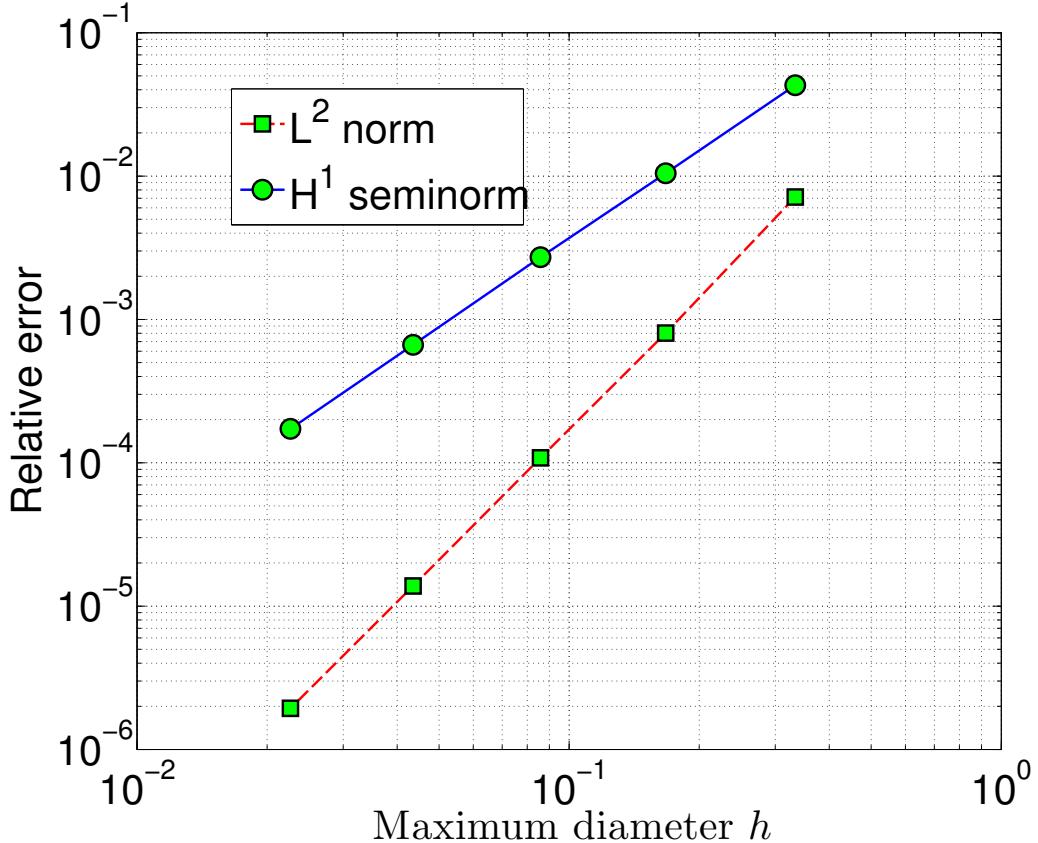


Fig. 15: Rates of convergence for Poisson problem on polygonal meshes.

unique solution (Bernstein basis functions) was realized on each edge. The dual (unconstrained convex minimization) problem was solved by Newton's method.

First, linear independence of the shape functions and conformity of the max-ent approximation were established. Plots of the shape functions and the approximation of a bivariate polynomial were presented to affirm the reproducing properties of the shape functions. Applications of the max-ent shape functions in a Galerkin method were presented. To realize high-accuracy and efficiency in the numerical integration, we tailored the corrected integration scheme proposed by Duan et al. [38] for polygonal elements. The quadratic patch test was passed to near machine-precision accuracy, and for

a Poisson problem on polygonal meshes as well as on quadrilateral meshes comprised of congruent trapezoids, the method delivered optimal rates of convergence of 3 and 2 in the  $L^2(\Omega)$  norm and the  $H^1(\Omega)$  seminorm, respectively. On self-similar trapezoidal meshes, it is well-established that quadratic serendipity isoparametric finite elements yield suboptimal rates of convergence [40]. The max-ent formulation is straightforward and holds promise for further extensions to higher-order complete shape functions on two- and three-dimensional elements. While retaining the advantage of requiring fewer degrees of freedom than tensor-product Lagrange shape functions, it can yield optimal convergence rates in Sobolev norms and provide accurate solutions even on distorted meshes, thus overcoming some of the deficiencies of serendipity isoparametric finite elements.

## 7. Acknowledgment

The author thanks Seyed Mousavi for his assistance in the **Matlab** implementation. Helpful discussions with Adrian Rosolen, Joseph Bishop, Scott Mitchell and Mohamed Ebeida are also gratefully acknowledged.

## References

- [1] A. Bompadre, L. E. Perotti, C. J. Cyron, M. Ortiz, Convergent meshfree approximation schemes of arbitrary order and smoothness, *Computer Methods in Applied Mechanics and Engineering* 221-222 (2012) 83–103.
- [2] K. Hormann, N. Sukumar, Maximum entropy coordinates for arbitrary polytopes, *Computer Graphics Forum* 27 (5) (2008) 1513–1520.
- [3] A. Rand, A. Gillette, C. Bajaj, Quadratic serendipity finite elements on polygons using generalized barycentric coordinates, *Mathematics of Computation*, in press, 2013.

- [4] E. L. Wachspress, A Rational Finite Element Basis, Vol. 114 of Mathematics in Science and Engineering, Academic Press, 1975.
- [5] M. S. Floater, Mean value coordinates, Computer Aided Geometric Design 20 (1) (2003) 19–27.
- [6] P. Joshi, M. Meyer, T. DeRose, B. Green, T. Sanocki, Harmonic coordinates for character articulation, ACM Transactions on Graphics 26 (3) (2007) 71/1–71/9.
- [7] N. Sukumar, Construction of polygonal interpolants: a maximum entropy approach, International Journal for Numerical Methods in Engineering 61 (12) (2004) 2159–2181.
- [8] G. Dasgupta, Interpolants within convex polygons: Wachspress' shape functions, Journal of Aerospace Engineering 16 (1) (2003) 1–8.
- [9] N. Sukumar, A. Tabarraei, Conforming polygonal finite elements, International Journal for Numerical Methods in Engineering 61 (12) (2004) 2045–2066.
- [10] N. Sukumar, E. A. Malsch, Recent advances in the construction of polygonal finite element interpolants, Archives of Computational Methods in Engineering 13 (1) (2006) 129–163.
- [11] P. Milbradt, T. Pick, Polytope finite elements, International Journal for Numerical Methods in Engineering 73 (12) (2007) 1811–1835.
- [12] M. Wicke, M. Botsch, M. Gross, A finite element method on convex polyhedra, Computer Graphics Forum 26 (3) (2007) 355–364.
- [13] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, M. Gross, Polyhedral finite elements using harmonic basis functions, Computer Graphics Forum 27 (5) (2008) 1521–1529.

- [14] S. Weïer, Residual error estimates for BEM-based FEM on polygonal meshes, *Numerische Mathematik* 118 (2011) 765–788.
- [15] S. Rjasanow, S. Weïer, Higher order BEM-based FEM on polygonal meshes, *SIAM Journal of Numerical Analysis* 50 (5) (2012) 2357–2378.
- [16] M. Kraus, P. Steinmann, Finite element formulations for 3D convex polyhedra in nonlinear continuum mechanics, *Computer Assisted Methods in Engineering and Science* 19 (2012) 121–134.
- [17] J. E. Bishop, A displacement-based finite-element formulation for general polyhedra using harmonic shape functions, in review, 2013.
- [18] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of virtual element methods, *Mathematical Models and Methods in Applied Sciences* 23 (2013) 199.
- [19] A. Rand, A. Gillette, C. Bajaj, Error estimates for generalized barycentric interpolation, *Advances in Computational Mathematics* 37 (3) (2012) 417–439.
- [20] A. Rand, A. Gillette, C. Bajaj, Interpolation error estimates for mean value coordinates over convex polygons, *Advances in Computational Mathematics*, DOI: 10.1007/s10444-012-9282-z, 2013.
- [21] A. Tabarraei, N. Sukumar, Extended finite element method on polygonal and quadtree meshes, *Computer Methods in Applied Mechanics and Engineering* 197 (5) (2008) 425–438.
- [22] J. E. Bishop, Simulating the pervasive fracture of materials and structures using randomly closed packed Voronoi tessellations, *Computational Mechanics* 44 (4) (2009) 455–471.

- [23] E. T. Ooi, C. M. Song, F. Tin-Loi, Z. J. Yang, Polygon scaled boundary finite elements for crack propagation modeling, *International Journal for Numerical Methods in Engineering* 91 (3) (2012) 319–342.
- [24] C. Talischi, G. H. Paulino, C. H. Le, Honeycomb Wachspress finite elements for structural topology optimization, *Journal of Structural and Multidisciplinary Optimization* 37 (6) (2009) 569–583.
- [25] C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes, Polygonal finite elements for topology optimization: A unifying paradigm, *International Journal for Numerical Methods in Engineering* 82 (6) (2010) 671–698.
- [26] C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes, PolyTop: A Matlab implementation of a general topology framework using unstructured polygonal finite element meshes, *Journal of Structural and Multidisciplinary Optimization* 45 (3) (2012) 329–357.
- [27] M. Yip, J. Mohle, J. E. Bolander, Automated modeling of three-dimensional structural components using irregular lattices, *Journal of Computed-Aided Civil and Infrastructure Engineering* 20 (6) (2005) 393–407.
- [28] D. Sieger, P. Alliez, M. Botsch, Optimizing Voronoi diagrams for polygonal finite elements, *Proceedings of the 19th International Meshing Roundtable*, 2010, pp. 335–350.
- [29] C. Talischi, G. H. Paulino, A. Pereira, I. F. M. Menezes, PolyMesher: A general-purpose mesh generator for polygonal elements written in Matlab, *Journal of Structural and Multidisciplinary Optimization* 45 (3) (2012) 309–328.

- [30] M. S. Ebeida, S. A. Mitchell, Uniform random Voronoi meshes, in: Proceedings of the 20th International Meshing Roundtable, Paris, France, 2011, pp. 258–275.
- [31] M. S. Ebeida, A. A. Davidson, A. Patney, P. M. Knupp, S. A. Mitchell, J. D. Owens, Efficient maximal Poisson-disk sampling, in: ACM Transactions on Graphics, Vol. 30, 2011, p. 49.
- [32] M. S. Ebeida, S. A. Mitchell, A. Patney, A. A. Davidson, J. D. Owens, A simple algorithm for maximal Poisson-disk sampling in high dimensions, Computer Graphics Forum 31 (2) (2012) 785–794.
- [33] C. J. Cyron, M. Arroyo, M. Ortiz, Smooth, second order, non-negative meshfree approximants selected by maximum entropy, International Journal for Numerical Methods in Engineering 79 (13) (2009) 1605–2632.
- [34] D. González, E. Cueto, M. Doblaré, A higher order method based on local maximum entropy approximation, International Journal for Numerical Methods in Engineering 83 (6) (2010) 741–764.
- [35] A. Rosolen, D. Millán, M. Arroyo, Second-order convex *maximum entropy* approximants with applications to high-order PDE, International Journal for Numerical Methods in Engineering 94 (2) (2013) 150–182.
- [36] N. Sukumar, R. W. Wright, Overview and construction of meshfree basis functions: From moving least squares to entropy approximants, International Journal for Numerical Methods in Engineering 70 (2) (2007) 181–205.
- [37] M. Arroyo, M. Ortiz, Local *maximum-entropy* approximation schemes: a seamless bridge between finite elements and meshfree methods, Inter-

- national Journal for Numerical Methods in Engineering 65 (13) (2006) 2167–2202.
- [38] Q. Duan, X. Li, H. Zhang, T. Belytschko, Second-order accurate derivatives and integration schemes for meshfree methods, International Journal for Numerical Methods in Engineering 92 (2012) 399–424.
  - [39] D. N. Arnold, D. Boffi, R. S. Falk, L. Gastaldi, Finite element approximation on quadrilateral meshes, Communications in Numerical Methods in Engineering 17 (11) (2001) 805–812.
  - [40] D. N. Arnold, D. Boffi, R. S. Falk, Approximation by quadrilateral finite elements, Mathematics of Computation 71 (239) (2002) 909–922.
  - [41] D. N. Arnold, G. Awanou, The serendipity family of finite elements, Foundations of Computational Mathematics 11 (3) (2011) 337–344.
  - [42] E. T. Jaynes, Information theory and statistical mechanics, Physical Review 106 (4) (1957) 620–630.
  - [43] E. T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, Cambridge, UK, 2003.
  - [44] C. E. Shannon, A mathematical theory of communication, The Bell Systems Technical Journal 27 (1948) 379–423.
  - [45] S. Kullback, R. A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1) (1951) 79–86.
  - [46] J. E. Shore, R. W. Johnson, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, IEEE Transactions on Information Theory 26 (1) (1980) 26–36.

- [47] J. Skilling, The axioms of maximum entropy, in: G. J. Erickson, C. R. Smith (Eds.), Maximum-Entropy and Bayesian Methods in Science and Engineering, Vol. 1: Foundations, Dordrecht, The Netherlands, 1988, pp. 173–187.
- [48] R. T. Farouki, The Bernstein polynomial basis: a centennial retrospective, *Computer Aided Geometric Design* 29 (2012) 379–419.
- [49] N. Sukumar, R. J.-B. Wets, Deriving the continuity of maximum-entropy basis functions via variational analysis, *SIAM Journal of Optimization* 18 (3) (2007) 914–925.
- [50] R. L. Burden, J. D. Faires, Numerical Analysis, eight Edition, Brooks/Cole, Belmont, CA, 2004.
- [51] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, *SIAM Review* 41 (1999) 637–676.
- [52] Y. Liu, W. P. Wang, B. Levy, F. Sun, D. M. Yan, L. Lu, C. L. Yang, On centroidal Voronoi tessellation—energy smoothness and fast computations, *ACM Transactions on Graphics* 28 (4) (2009) 101.
- [53] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, *Computer Methods in Applied Mechanics and Engineering* 139 (1996) 3–47.
- [54] D. A. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, *International Journal for Numerical Methods in Engineering* 21 (1985) 1129–1148.
- [55] Y. Krongauz, T. Belytschko, Consistent pseudo-derivatives in meshless methods, *Computer Methods in Applied Mechanics and Engineering* 146 (2–3) (1997) 371–386.

- [56] J. S. Chen, C. T. Wu, S. Yoon, Y. You, A stabilized conforming nodal integration for Galerkin meshfree methods, International Journal for Numerical Methods in Engineering 50 (2001) 435–466.
- [57] M. A. Puso, J. S. Chen, E. Zywicz, W. Elmer, Meshfree and finite element nodal integration methods, International Journal for Numerical Methods in Engineering 74 (3) (2008) 416–446.
- [58] G. R. Liu, K. Y. Dai, T. T. Nguyen, A smoothed finite element for mechanics problems, Computational Mechanics 39 (2007) 859–877.
- [59] K. Y. Dai, G. R. Liu, T. T. Nguyen, A  $n$ -sided polygonal smoothed finite element method ( $n$ SFEM) for solid mechanics, Finite Elements in Analysis and Design 43 (2007) 847–860.
- [60] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, sixth Edition, Elsevier Butterworth-Heinemann, Oxford, UK, 2005.
- [61] N.-S. Lee, K.-J. Bathe, Effects of element distortions on the performance of isoparametric elements, International Journal for Numerical Methods in Engineering 36 (1993) 3553–3576.
- [62] S. Rajendran, A technique to develop mesh-distortion immune finite elements, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1044–1063.