Recent Advances in the Construction of Polygonal Finite Element Interpolants

N. Sukumar

Department of Civil and Environmental Engineering
University of California
One Shields Avenue
Davis, CA 95616. U.S.A.
nsukumar@ucdavis.edu

E.A. Malsch

Institute of Applied Mechanics
Technische Universität Braunschweig
Spielmannstraße 11, 38023 Braunschweig, Germany

Summary

This paper is an overview of recent developments in the construction of finite element interpolants, which are C^0-conforming on polygonal domains. In 1975, Wachspress proposed a general method for constructing finite element shape functions on convex polygons. Only recently has renewed interest in such interpolants surfaced in various disciplines including: geometric modeling, computer graphics, and finite element computations. This survey focuses specifically on polygonal shape functions that satisfy the properties of barycentric coordinates: (a) form a partition of unity, and are non-negative; (b) interpolate nodal data (Kronecker-delta property), (c) are linearly complete or satisfy linear precision, and (d) are smooth within the domain. We compare and contrast the construction and properties of various polygonal interpolants—Wachspress basis functions, mean value coordinates, metric coordinate method, natural neighbor-based coordinates, and maximum entropy shape functions. Numerical integration of the Galerkin weak form on polygonal domains is discussed, and the performance of these polygonal interpolants on the patch test is studied.

1 INTRODUCTION

The three-node triangle and the four-node quadrilateral, or quadrangle, are basic element shapes used in most two-dimensional finite element methods. General polygonal elements, or n-gons, with more than four vertices are not employed in any conventional finite element computations. The removal of this restriction has the potential to lead to new developments in mesh generation and the finite element method. Polygonal finite elements provide greater flexibility for the meshing of complex geometries, are potentially useful for the modeling of polycrystalline materials [18], can serve as transition elements in finite element meshes [14], and are well-suited for material design [13]. Furthermore, such elements can prove to be useful for satisfying incompressibility in constrained media problems that arise in solid and fluid mechanics.

In 1975, Wachspress [55] used concepts of projective geometry to develop rational polynomial interpolants for convex polygons. Recently, renewed interest in Wachspress basis functions [11, 12] and in the construction of barycentric coordinates on n-gons has emerged [15, 17, 22, 29–35, 50, 52, 53, 57–60]. Apart from finite element applications, the construction of barycentric coordinates on polygons is of significant interest for surface parametrization, geometric modeling, and computer graphics [15, 16, 35].

The purpose of this paper is to present the main developments in the construction of conforming polygonal interpolants and to explore their use as trial and test functions in a Galerkin method. There have been a number of recent papers in which different barycentric approximations on irregular polygons have been proposed, and we primarily focus our

attention on these advances. The main ingredients of a suitable finite element interpolant are stated, and a unifying prescription for the construction of polygonal interpolants is presented to better understand the similarities and distinctions of the various interpolants. The important issue of numerical integration of the weak form on polygonal domains is addressed, and the performance of the polygonal interpolants on the patch test is studied.

2 CONFORMING INTERPOLANTS ON IRREGULAR POLYGONS

Consider a polygonal domain $\Omega \subset \mathbf{R}^2$ that is described by n nodes. The ith node is labeled by p_i, and $\mathbf{x}_i \equiv (x_i, y_i)$ is used to denote the coordinate of node i. In addition to vertex-nodes, nodes can also be located along the edges (side- or mid-side nodes), or even in the interior of the polygon (Figure 1). Any point p with coordinate $\mathbf{x} \equiv (x, y) \in \bar{\Omega}$ ($\bar{\Omega} = \Omega \cup \partial\Omega$), has a set of associated shape functions $\phi_i(\mathbf{x})$. An interpolation scheme for a scalar-valued function $u(\mathbf{x}) : \bar{\Omega} \to \mathbf{R}$ can be written as:

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \phi_i(\mathbf{x})u_i, \tag{1}$$

where u_i are the unknowns at the n neighbors of point p. On using C^0 shape functions, the interpolant $u^h(\mathbf{x})$ should satisfy the following properties:

1. Form a partition of unity to assure constant precision, and that $\phi_i(\mathbf{x})$ is non-negative and bounded:

$$\sum_{i=1}^{n} \phi_i(\mathbf{x}) = 1, \quad 0 \leq \phi_i(\mathbf{x}) \leq 1. \tag{2}$$

2. Interpolate nodal data:

$$\phi_i(\mathbf{x}_j) = \delta_{ij}, \tag{3}$$

   where δ_{ij} is the Kronecker-delta. The above equation indicates that the interpolated result at a node is identical to the nodal data: $u^h(\mathbf{x}_i) = u_i$. Equation (2) ensures that the interpolated result is bounded between the minimum and maximum of the nodal values: $\min_i u_i \leq u^h(\mathbf{x}) \leq \max_i u_i$, which is a statement of the discrete maximum principle and a requirement for the numerical discretization of the diffusion equation.

3. Linear completeness [23] or linear precision:

$$\sum_{i=1}^{n} \phi_i(\mathbf{x})\mathbf{x}_i = \mathbf{x}, \tag{4}$$

   which indicates that the shape functions can exactly reproduce a linear function. For second-order partial differential equations (PDEs), constant and linear precision in the trial function are sufficient conditions for convergence in a Galerkin method [48].

4. The shape function $\phi_i \in C^\infty$ within the domain. Along the edges of the polygon, the interpolant must be piece-wise linear (C^0 function):

$$u^h(t) = tu_1 + (1 - t)u_2, \quad \mathbf{x} = t\mathbf{x}_1 + (1 - t)\mathbf{x}_2, \quad \mathbf{x} \in \partial\Omega, t \in [0, 1]. \tag{5}$$

   The above equation ensures that linear essential boundary conditions can be imposed exactly in a Galerkin method.
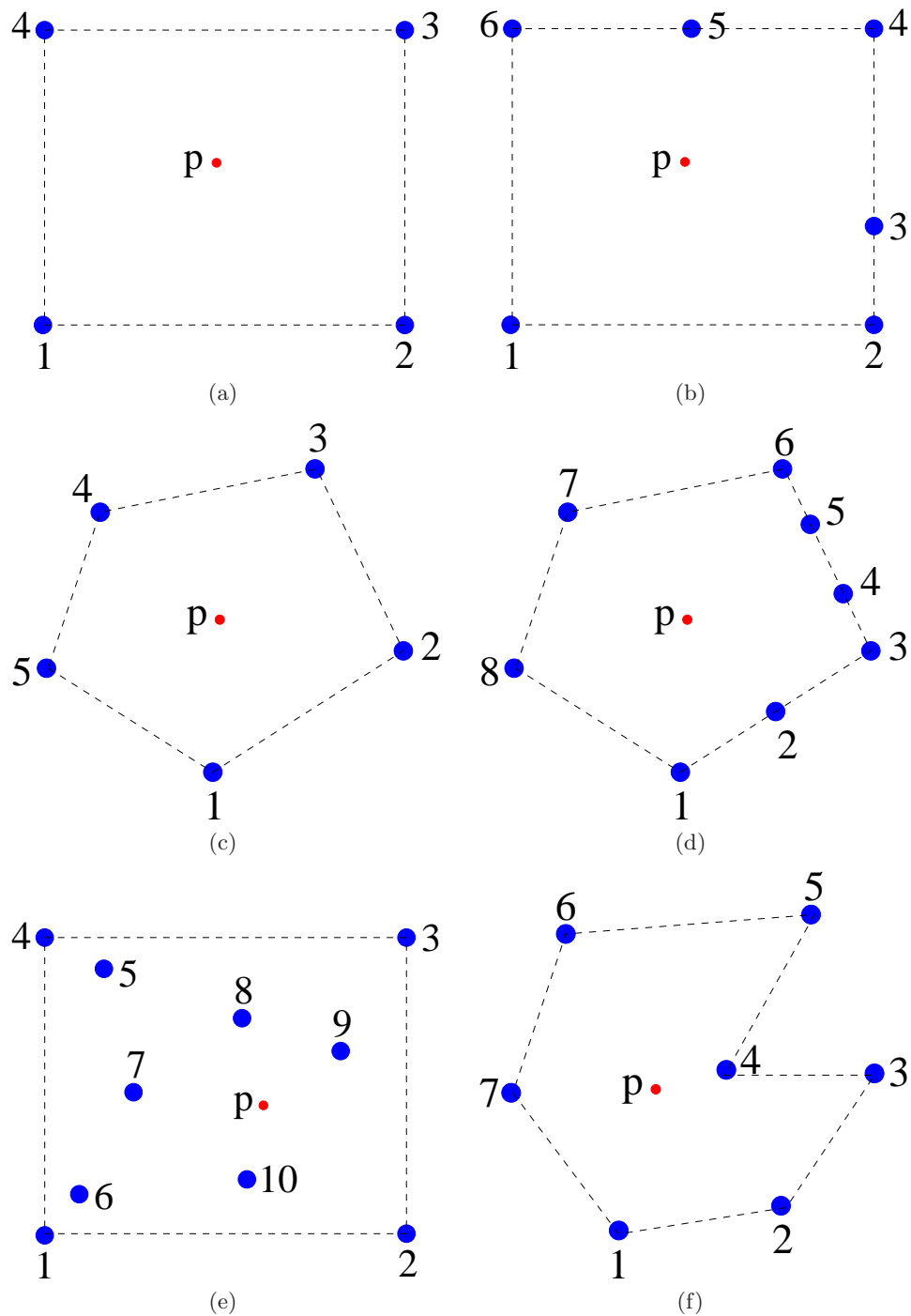
Figure 1. Nodal discretization of polygonal domains. (a) Square ($n = 4$); (b) Square with side- and mid-side nodes ($n = 6$); (c) Pentagon ($n = 5$); (d) Pentagon with side- and mid-side nodes ($n = 8$); (e) Square with interior nodes ($n = 10$); (f) Concave heptagon ($n = 7$)

2.1 Construction of Shape Functions using Length and Area Metrics

The polygonal shape functions $\phi_i(\mathbf{x})$ that are treated in this paper are constructed from a set of non-negative weight functions $\{w_1(\mathbf{x}), w_2(\mathbf{x}), \ldots, w_n(\mathbf{x})\}$. Each $w_i(\mathbf{x})$ attains its maximum value at $\mathbf{x}_i$ and monotonically decreases with increasing distance from $\mathbf{x}_i$. The polygonal shape functions are represented in the general (Shepard) form [45]:

$$\phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum\limits_{j=1}^{n} w_j(\mathbf{x})}, \tag{6}$$

and hence by construction the $\{\phi_i\}$ form a partition of unity.

By imposing further conditions on the form of $w_i(\mathbf{x})$, the remaining requirements are similarly satisfied by construction. In a simple polygon, a vertex p_k in $\mathbf{R}^d$ is connected to d other vertices. Let $ind(k)$ denote the collection of nodal indices that are connected to node p_k [60]. First, define a function $b_i(\mathbf{x})$ that interpolates nodal data and is C^∞ smooth over the domain Ω and C^0 on the boundary segment. In two dimensions, consider functions $r_{jk}(\mathbf{x})$, which are zero-valued on the boundary segment from p_j to p_k and positive and monotonically increasing everywhere else in the domain:

$$b_i(\mathbf{x}) = a_i \prod_{j,\, k \neq i} r_{jk}(\mathbf{x}). \tag{7}$$

A linearly independent set of shape functions $\{\phi_i\}$ is constructed by choosing $\phi_i(\mathbf{x})$ to be zero on all boundaries not adjacent to the ith node. This is achieved by letting $w_i(\mathbf{x}) \equiv b_i(\mathbf{x})$ and using Eq. (6). In Figure 2, the shape function for p_3 is zero-valued on the boundaries that are not adjacent to node p_3. The quantity a_i is a non-zero constant or a non-zero function. In two-dimensional convex formulations, the value of a_i is the area of the triangle that is formed by node p_i and its incident nodes in $ind(i)$. In general, it is convenient to set the value of a_i to unity.
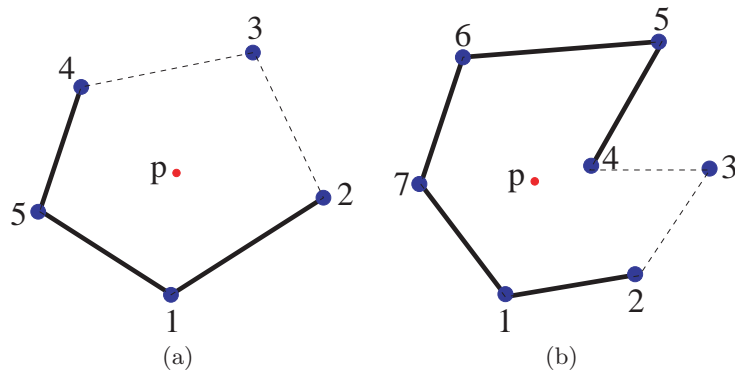
Figure 2. Adjacent boundaries. (a) Convex element; (b) Concave element

Alternatively, the function $b_i(\mathbf{x})$ can be reformulated such that it depends only on the neighboring boundaries [35]. This simplified form is valid everywhere in the domain except where the product of all the $r_{jk}(\mathbf{x})$ functions vanish. For example, in two dimensions

$$\hat{b}_i(\mathbf{x}) = \frac{b_i(\mathbf{x})}{\prod r_{jk}(\mathbf{x})} = \frac{a_i}{\prod\limits_{j,k \in ind(i)} r_{jk}(\mathbf{x})}, \tag{8}$$
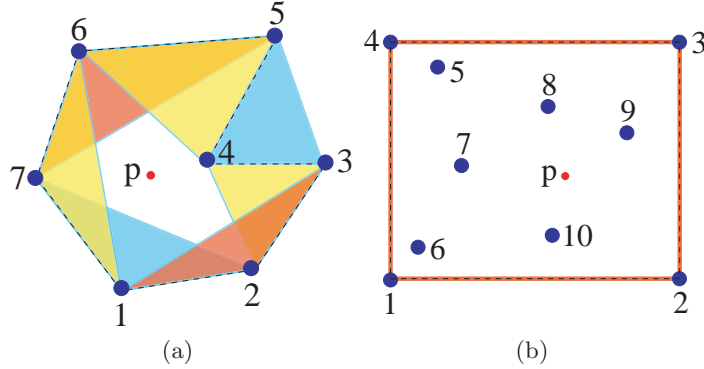
Figure 3. Shape functions to ensure satisfaction of linear precision within the domain and on the boundary. (a) Combining shape functions at the vertices; (b) Combining shape functions on the convex hull

and $w_i(\mathbf{x}) \equiv \hat{b}_i(\mathbf{x})$ is used in Eq. (6) to obtain the shape functions. The above construction is general enough to be applied to any polytope in $\mathbf{R}^d$, provided that a_i and $r_{i_1 i_2 \ldots i_d}(\mathbf{x})$ can be defined. The shape functions $\phi_i(\mathbf{x})$ derived by applying Eq. (7) are exactly the same as those derived using Eq. (8), except that the numerical evaluation of the function in Eq. (8) is undefined on the boundary segment. Shape functions are usually expressed as in Eq. (7), whereas barycentric coordinates are typically of the form given in Eq. (8).

Method	$r_{ij}(\mathbf{x})$	a_i	References
Rational polynomial[a]	$A(p, p_i, p_j)$	$A(p_i, p_j, p_k)$	[55]
Mean value coordinates	$A(p, p_i, p_j)$	$l_i(\mathbf{x})$	[15]
Metric coordinate	$\|\mathbf{x} - \mathbf{x}_i\| + \|\mathbf{x} - \mathbf{x}_j\| - \|\mathbf{x}_i - \mathbf{x}_j\|$	1	[34]
– interior node[b]	$\|\mathbf{x} - \mathbf{x}_i\|$	1	[32]

[a] The linear triangle and trapezoid shape functions are subsets of this method.
[b] The $r_i(\mathbf{x})$ function depends only on one node.

Table 1. General form of interpolants on polygons

The r- and a-functions are not uniquely prescribed by the shape function requirements. In Table 1, different values for $r_{ij}(\mathbf{x})$ and a_i in Eq. (8) for some of the common interpolant constructions are indicated. On a two dimensional polygon for example, a series of valid interpolants can be constructed using the following equation:

$$\hat{b}_j^{mn}(\mathbf{x}) = \frac{a_j}{r_{ij}^m(\mathbf{x}) r_{jk}^n(\mathbf{x})}, \tag{9}$$

which is valid for positive integers m and n. Using this approach, a power-series like solution on polygonal domains, similar to the R-function construction [42], is conceivable.

The third and fourth requirements, of linear completeness and linear edge behavior, similarly need to be satisfied. The interpolants formed using $b_i(\mathbf{x})$ and $\hat{b}_i(\mathbf{x})$ from Eqs. (7) and (8) can be combined such that the resulting shape functions are linearly precise. The number of nodes that are used to determine the weight $w_i(\mathbf{x})$ for any specific node i categorizes the different approaches for the construction of shape functions [15]. The so-called three-point method depends on the interpolation at the node and its two closest neighbors along the boundary. The simplest shape function construction that satisfies both the linearity
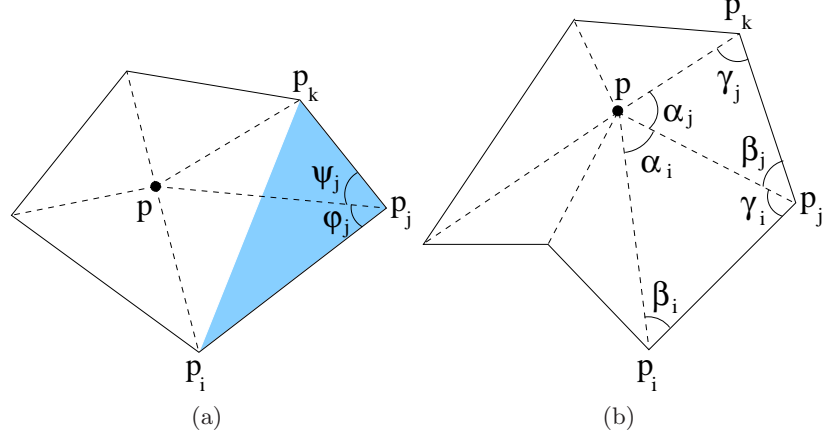
Figure 4. Barycentric coordinates. (a) Wachspress's basis function [35]; (b) Mean
value coordinates [15]

and independence requirements are Wachspress shape functions on a convex polygon [55]
and Floater's mean value coordinates [15] on a concave polygon. Both these approaches
are three-point methods (Figure 4). A general prescription for five-point barycentric coor-
dinates is outlined by Hormann [22].

Using a slight variation of Hormann's method, a general form for shape functions, which
satisfy linear independence, linear completeness, and linear edge behavior is constructed.
For each vertex p_j, defined by the triangle (p_i, p_j, p_k), the three vertex functions are pre-
scribed entirely by the conditions for reproducing a linear field:

$$
\begin{array}{rcl}
\lambda_{ij}(\mathbf{x}) + \lambda_{jj}(\mathbf{x}) + \lambda_{kj}(\mathbf{x}) &=& 1, \\
x_i\lambda_{ij}(\mathbf{x}) + x_j\lambda_{jj}(\mathbf{x}) + x_k\lambda_{kj}(\mathbf{x}) &=& x, \\
y_i\lambda_{ij}(\mathbf{x}) + y_j\lambda_{jj}(\mathbf{x}) + y_k\lambda_{kj}(\mathbf{x}) &=& y.
\end{array}
\tag{10}
$$

The weight $\lambda_{ij}(\mathbf{x})$ is the linear weight function for node i in the triangle (p_i, p_j, p_k). The new
weight function for node p_j is defined as a linear combination of the weighted interpolants
for the nodes:

$$
\begin{aligned}
w_j(\mathbf{x}) &= \lambda_{ji}(\mathbf{x})b_i(\mathbf{x}) + \lambda_{jj}(\mathbf{x})b_j(\mathbf{x}) + \lambda_{jk}(\mathbf{x})b_k(\mathbf{x}), \\
\hat{w}_j(\mathbf{x}) &= \lambda_{ji}(\mathbf{x})\hat{b}_i(\mathbf{x}) + \lambda_{jj}(\mathbf{x})\hat{b}_j(\mathbf{x}) + \lambda_{jk}(\mathbf{x})\hat{b}_k(\mathbf{x}),
\end{aligned}
\tag{11}
$$

where $b_j(\mathbf{x})$ and $\hat{b}_j(\mathbf{x})$ are defined per Eqs. (7) and (8), respectively. The general form of
the weight function is:

$$
\hat{w}_j(\mathbf{x}) = \lambda_{ji}(\mathbf{x})\frac{a_i(\mathbf{x})}{r_{(i-1)i}(\mathbf{x})r_{ij}(\mathbf{x})} + \lambda_{jj}(\mathbf{x})\frac{a_j(\mathbf{x})}{r_{ij}(\mathbf{x})r_{jk}(\mathbf{x})} + \lambda_{jk}(\mathbf{x})\frac{a_k(\mathbf{x})}{r_{jk}(\mathbf{x})r_{k(k+1)}(\mathbf{x})},
\tag{12}
$$

and using the partition of unity, Eq. (2), the shape function for node j is:

$$
\phi_j(\mathbf{x}) = \frac{w_j(\mathbf{x})}{\displaystyle\sum_{k=1}^{n} w_k(\mathbf{x})} = \frac{\hat{w}_j(\mathbf{x})}{\displaystyle\sum_{k=1}^{n} \hat{w}_k(\mathbf{x})}.
\tag{13}
$$

The weight function $\hat{w}_j(\mathbf{x})$ depends on the location of five nodes p_{i-1}, p_i, p_j, p_k and p_{k+1},
where p_{i-1} is a neighbor of node p_i and p_{k+1} is a neighbor of node k. The resulting shape

function is smooth wherever $r_{ij}(\mathbf{x})$ is non-zero, and the shape functions are also linearly complete as required by Eq. (4). The result, however, is not necessarily bounded between zero and one in the domain. The judicious choice of a_i that can assure boundedness is the subject of on-going research.

An alternative approach to ensure linear precision over concave domains or in domains with interior nodes, uses the set of interpolants $\{R_i(\mathbf{x})\}$ that are linearly precise within the convex hull of the set of nodes. Such an n-point approach may be better-suited for problems with interior nodes, since interior nodes do not have the associated vertices that are required in the three-point method.

Separate the nodes on the boundary of the convex hull and those in the interior into two disjoint sets, namely sets $\mathcal{A}$ and $\mathcal{B}$, respectively. First, derive the Wachspress shape functions for the nodes on the convex hull that satisfy linear precision in the entire domain. They define the $\{R_i(\mathbf{x})\}$ for all $i \in \mathcal{A}$. Also derive the shape functions for the interior nodes that are linearly independent and bounded, but are not necessarily linearly precise, as $\{\tilde{\phi}_i\}$ for all $i \in \mathcal{B}$. Split the summation over all the nodes into a sum over each set $\mathcal{A}$ and $\mathcal{B}$. Then, replace the vertex location $\mathbf{x}_j$ with the weighted sum of the $\{R_i(\mathbf{x}_j)\}$ over the set $\mathcal{A}$, according to Eq. (4), and re-arrange the order of summation:

$$\sum_{\mathcal{A}} \phi_i(\mathbf{x})\mathbf{x}_i + \sum_{\mathcal{B}} \tilde{\phi}_j(\mathbf{x})\mathbf{x}_j = \sum_{\mathcal{A}} \phi_i(\mathbf{x})\mathbf{x}_i + \sum_{\mathcal{A}} \mathbf{x}_i \sum_{\mathcal{B}} \tilde{\phi}_j(\mathbf{x})R_i(\mathbf{x}_j) = \sum_{\mathcal{A}} R_i(\mathbf{x})\mathbf{x}_i = \mathbf{x}. \quad (14)$$

Equating the components of the sum at a node on the convex hull returns a shape function for that node, which in combination with the shape functions for the interior nodes satisfies the linear field:

$$\phi_i(\mathbf{x}) = R_i(\mathbf{x}) - \sum_{\mathcal{B}} \tilde{\phi}_j(\mathbf{x})R_i(\mathbf{x}_j), \quad i \in \mathcal{A}. \quad (15)$$

Both $\phi_i(\mathbf{x})$ and $R_i(\mathbf{x})$ are positive over the domain, $\phi_i(\mathbf{x}) < R_i(\mathbf{x}) < 1$, but the $\phi_i(\mathbf{x})$ are not necessarily non-negative. The construction and implementation of methods to adjust the shape function so that each interpolant is both bounded and linearly precise are presented in [32, 33].

2.1.1 Wachspress shape functions

Using elements of projective geometry, Wachspress constructed polygonal shape functions on any convex polygonal domain [55]. Warren extended the formulation to three-dimensional convex polytopes [58]. The conventional linearly precise shape functions for triangles and trapezoids, and the Wachspress shape functions for convex polygons can be constructed using a linear $r_{ij}(\mathbf{x})$ and setting a_i to be constant. The shape functions can be defined using the area between the nodes p_i and p_j, and the inserted point p (Figure 5). Hereafter, for brevity and ease of notation, we denote the point by just p, mindful that it indicates the position $\mathbf{x} \equiv (x, y)$. The Vandermonde determinant in two dimensions is:

$$r_{ij}(\mathbf{x}) = A(p, p_i, p_j) = \frac{1}{2} \begin{vmatrix} x & y & 1 \\ x_i & y_i & 1 \\ x_j & y_j & 1 \end{vmatrix}, \quad (16)$$

and

$$a_j = A(p_i, p_j, p_k) = \frac{1}{2} \begin{vmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{vmatrix}. \quad (17)$$
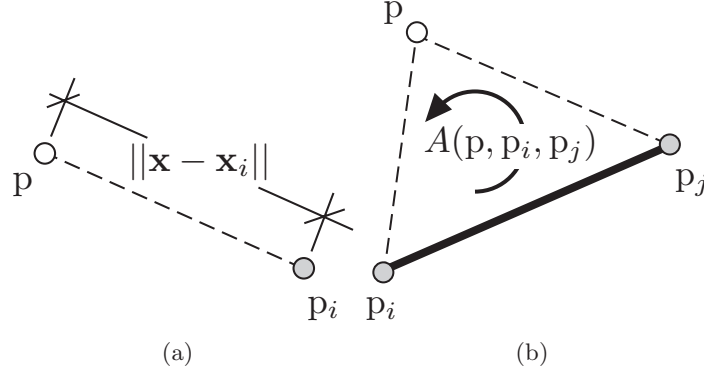
Figure 5. Geometric measures. (a) Length; (b) Area

From Eq. (7), the numerator for the triangular shape functions are constructed:

$$b_i(\mathbf{x}) = a_i r_{jk}(\mathbf{x}) = A(p_i, p_j, p_k)A(p, p_j, p_k). \tag{18}$$

The constants $a_i = a_j = a_k$, and the denominator of the shape function $\phi_i(\mathbf{x})$ is:

$$a_i r_{jk} + a_j r_{ki} + a_k r_{ij} = A(p_i, p_j, p_k)\big(A(p, p_j, p_k) + A(p_i, p, p_k) + A(p_i, p_j, p)\big), \tag{19}$$

which simplifies to

$$a_i r_{jk} + a_j r_{ki} + a_k r_{ij} = A^2(p_i, p_j, p_k), \tag{20}$$

and dividing Eq. (18) by Eq. (20) leads to the well-known expression for triangular shape functions:

$$\phi_i(\mathbf{x}) = \frac{A(p, p_j, p_k)}{A(p_i, p_j, p_k)}. \tag{21}$$

The linear shape function for a triangle is also termed as area coordinate or barycentric coordinate. The numerator is the area of the triangle between points p, p_j and p_k, and the denominator is the area of the entire triangle.

The shape function for a quadrilateral can be derived in the element coordinates using the above method. Given vertices p_i, p_j, p_k and p_ℓ, the numerator for the shape function $\phi_i(\mathbf{x})$ is:

$$b_i(\mathbf{x}) = a_i r_{jk} r_{k\ell} = A(p_\ell, p_i, p_j)A(p, p_j, p_k)A(p, p_k, p_\ell). \tag{22}$$

Unlike the triangle, the denominator is only constant in the case when the quadrilateral is a parallelogram. The shape function is:

$$\phi_i(\mathbf{x}) = \frac{A(p_d, p_a, p_b)A(p, p_b, p_c)A(p, p_c, p_d)}{\left(\begin{array}{c} A(p_\ell, p_i, p_j)A(p, p_j, p_k)A(p, p_k, p_\ell) + A(p_i, p_j, p_k)A(p, p_k, p_\ell)A(p, p_\ell, p_i) \\ +A(p_j, p_k, p_\ell)A(p, p_\ell, p_i)A(p, p_i, p_j) + A(p_k, p_\ell, p_i)A(p, p_i, p_j)A(p, p_k, p_\ell) \end{array}\right)}. \tag{23}$$

This approach applies to any convex quadrilateral, but the shape functions are only the same as those derived using the isoparametric transformation on trapezoids [29].

The shape function on any convex domain can be constructed using Wachspress's formulation. Using Eq. (8), the numerator for node j adjacent to nodes i and k is:

$$\hat{b}_j(\mathbf{x}) = \frac{A(p_i, p_j, p_k)}{A(p, p_i, p_j)A(p, p_j, p_k)}. \tag{24}$$
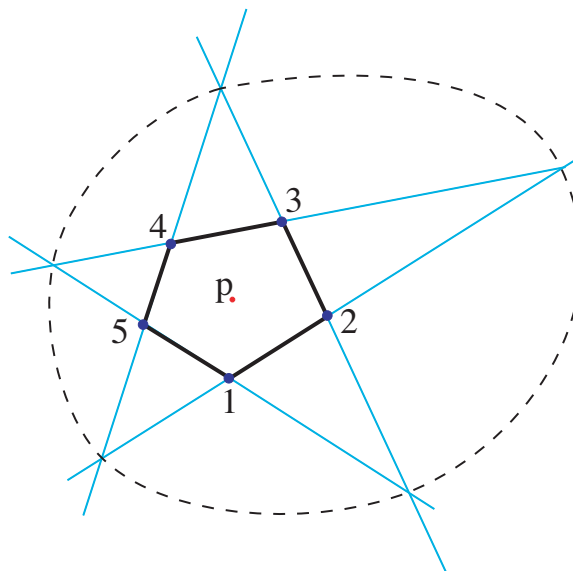
Figure 6. Path along which the Wachspress shape function for a pentagon is undefined

The shape function constructed using $\hat{b}_j(\mathbf{x})$ is undefined on the boundaries. The boundary behavior of a Wachspress shape function is linear and can be defined separately. The denominator of the shape function is a curve that passes through the intersections of the continuation of the boundary lines (Figure 6). For a polygon with n nodes the curve defined by the denominator is order $n-3$ and is proven to lie outside the domain of the convex domain [55]. Numerous examples of shape functions for specific convex polygons can be found in [12, 19, 34, 35, 47, 57].

The Wachspress coordinate formulation satisfies linear fields by construction. The original $\hat{b}_j(\mathbf{x})$ are equal to the transformed linearly precise weightings $\hat{w}_j(\mathbf{x})$ that are constructed using Eq. (12). Again, noting that the coordinate of p is $\mathbf{x}$,

$$\hat{w}_j(\mathbf{x}) = \frac{A(p_{i-1}, p_i, p)}{A(p_{i-1}, p_i, p_j)}\hat{b}_i(\mathbf{x}) + \frac{A(p_i, p, p_k)}{A(p_i, p_j, p_k)}\hat{b}_j(\mathbf{x}) + \frac{A(p, p_k, p_{k+1})}{A(p_j, p_k, p_{k+1})}\hat{b}_k(\mathbf{x})$$

$$= \frac{A(p_i, p_j, p_k)}{A(p_i, p_j, p)A(p, p_j, p_k)} = \hat{b}_j(\mathbf{x}). \tag{25}$$

The Wachspress shape functions are the lowest order shape functions that satisfy boundedness, linearity, and linear independence on convex shapes [59]. The numerical analysis of these basis functions [19] and their application to the construction of surface patches [10] have been explored. Warren [58] has presented its generalization to convex polytopes. Wachspress coordinates are rewritten in different forms depending on the application. Dasgupta [11, 12] used symbolic computations to compute the Wachspress basis function, whereas in [35], a simplified expression for the same is obtained:

$$\phi_j(\mathbf{x}) = \frac{\hat{w}_j(\mathbf{x})}{\sum\limits_{k=1}^{n} \hat{w}_k(\mathbf{x})}, \quad \hat{w}_j(\mathbf{x}) = \frac{A(p_i, p_j, p_k)}{A(p, p_i, p_j)A(p, p_j, p_k)} = 2\left(\frac{\cot \varphi_j + \cot \psi_j}{\|\mathbf{x} - \mathbf{x}_j\|^2}\right), \tag{26}$$

where the last expression is due to Meyer *et al.* [35], $A(a, b, c)$ is the signed area of triangle $[a, b, c]$, and φ_j and ψ_j are shown in Figure 4a.

2.1.2 Concave polygon

Concave domains can not be described using Wachspress convex polygon shape functions, or an isoparametric transformation. Either the constant a_i must be a function or the function $r_{jk}(\mathbf{x})$ can not be linear. Floater [15] proposed *mean value coordinates* (barycentric coordinates on n-gons), which satisfy the shape function requirements, except boundedness, on all two-dimensional concave domains everywhere except along the boundaries [15]. Define $a_j(\mathbf{x})$ as the distance between the point p and the node p_j. In two dimensions,

$$a_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_j\| = \sqrt{(x - x_j)^2 + (y - y_j)^2}. \tag{27}$$

On using Eq. (8), the numerator for vertex p_j with neighbor vertices p_i and p_k is:

$$\hat{b}_j(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_j\|}{A(p, p_i, p_j) A(p, p_j, p_k)}. \tag{28}$$

Alternatively, shape functions on a concave domain can be constructed using the *metric coordinate method* [34]. It is defined using a function that is zero only along a finite boundary segment. This allows for the construction of a shape function that is bounded and valid in $\mathbf{R}^2$. The boundary function $r_{ki}(\mathbf{x})$ can be formulated by exploiting the triangle inequality—the sum of two edge segments of a triangle is always greater or equal to the length of the third:

$$r_{ki}(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\| + \|\mathbf{x} - \mathbf{x}_i\| - \|\mathbf{x}_i - \mathbf{x}_k\|. \tag{29}$$

Now, from Eq. (8), the weight $\hat{b}_j$ is:

$$\hat{b}_j(\mathbf{x}) = \frac{1}{\big(\|\mathbf{x} - \mathbf{x}_k\| + \|\mathbf{x} - \mathbf{x}_i\| - \|\mathbf{x}_i - \mathbf{x}_k\|\big)\big(\|\mathbf{x} - \mathbf{x}_i\| + \|\mathbf{x} - \mathbf{x}_j\| - \|\mathbf{x}_j - \mathbf{x}_i\|\big)}. \tag{30}$$

Unlike convex domains, shape functions that are valid on concave domains can not be defined using a rational polynomial. Each rational polynomial is associated with a specific projective transformation of a convex polygon [55], and a concave shape can not be formed from a perspective transformation of a convex shape [8]. The Euclidean measure introduces a branch point singularity at a re-entrant node that allows for the construction of the shape function.

The form of the mean value coordinate function presented in Eq. (28) is made linearly precise using Eq. (12). In the five-point scheme with vertices p_{i-1}, p_i, p_j, p_k, and p_{k+1}, the influence of nodes p_{i-1} and p_{k+1} cancel out in the mean value coordinates [22]. The mean value coordinate is similar to the Wachspress shape function—both are three-point shape functions:

$$
\begin{aligned}
\hat{w}_j(\mathbf{x}) &= \frac{\|\mathbf{x} - \mathbf{x}_i\|}{A(p, p_i, p_j)} + \frac{\|\mathbf{x} - \mathbf{x}_j\| A(p_i, p, p_k)}{A(p, p_i, p_j) A(p, p_j, p_k)} + \frac{\|\mathbf{x} - \mathbf{x}_k\|}{A(p, p_j, p_k)} \\
&= 2\left(\frac{\tan(\alpha_i/2) + \tan(\alpha_j/2)}{\|\mathbf{x} - \mathbf{x}_j\|}\right) \\
&= \frac{8 A(p, p_i, p_j)/\|\mathbf{x} - \mathbf{x}_j\|}{\big(\|\mathbf{x} - \mathbf{x}_i\| + \|\mathbf{x} - \mathbf{x}_j\|\big)^2 - \|\mathbf{x}_i - \mathbf{x}_j\|^2} + \frac{8 A(p, p_j, p_k)/\|\mathbf{x} - \mathbf{x}_j\|}{\big(\|\mathbf{x} - \mathbf{x}_j\| + \|\mathbf{x} - \mathbf{x}_k\|\big)^2 - \|\mathbf{x}_j - \mathbf{x}_k\|^2},
\end{aligned} \tag{31}
$$

where the second equation is the weight function expression derived by Floater [15], and the angles α_i and α_j are shown in Figure 4b. The mean value coordinates are linearly precise and smooth in any polygonal domain, and bounded in convex polygonal domains. The boundary segments are defined in the limit and hence are handled separately in the numerical implementation [22], which is also the case in the computation of Laplace shape functions (Section 2.2) [52].

The shape functions $\phi_i(\mathbf{x})$ generated using the metric coordinate method are defined in the entire domain including the boundary, provided Eq. (8) is converted to Eq. (7). The resulting shape functions are five-point shape functions as defined in Eq. (12), and similar to mean value coordinates, they are necessarily bounded only in convex shapes. In the definition of Eq. (30) the constant a_i is set to one. A heuristic choice of a_i can result in bounded interpolants which satisfy all of the shape function requirements. The mean value coordinate method, on the other hand, is bounded only in star shaped polygons and does not afford the luxury of a free variable.

2.1.3 Polygon with side-nodes

While the mean value coordinates and metric coordinate method are applicable when nodes are added to a polygonal edge [22, 33, 34], a simpler formulation which assures linearity and boundedness, based on a projective transformation, can be applied to non-concave shapes [31]. To illustrate the construction of shape functions when side nodes are added to a convex polygon, we consider the quadtree data structure. Quadtree is a spatial data structure based on the recursive decomposition of a square in two dimensions [43]. In a quadtree, *hanging nodes* are generated, which leads to incompatibilities in the refined meshes (Figure 7). The construction of polygonal elements with side nodes [31] is adapted to quadtree elements with hanging nodes.

Consider the quadtree element A shown in Figure 7a with hanging node a at the midpoint of the edge 1–2. The procedure presented in [31] is followed to derive the shape functions on element A. Let the vertex nodes have coordinates: 1(0,1), 2(1,1), 3(1,0) and 4(0,0). The Wachspress (bilinear finite element) shape functions are:

$$\tilde{\phi}_1(\mathbf{x}) = y(1-x), \quad \tilde{\phi}_2(\mathbf{x}) = xy, \quad \tilde{\phi}_3(\mathbf{x}) = x(1-y), \quad \tilde{\phi}_4(\mathbf{x}) = (1-x)(1-y). \quad (32)$$

An extra node a is now inserted along the edge 1–2. The boundary is parametrized by s such that $s = 0$ at node 1 and $s = 1$ at node 2. Following [29], we can write

$$a = s(1-s), \quad b(\mathbf{x}) = 1 - (1-s)\tilde{\phi}_1(\mathbf{x}) - s\tilde{\phi}_2(\mathbf{x}), \quad c(\mathbf{x}) = \tilde{\phi}_1(\mathbf{x})\tilde{\phi}_2(\mathbf{x}), \quad (33\text{a})$$
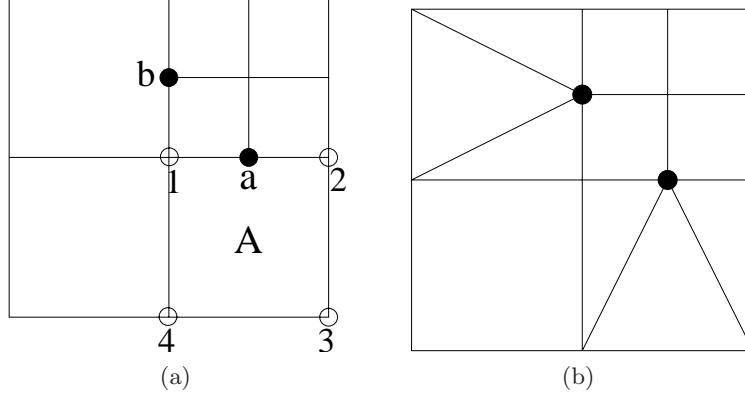
$$\phi_a(\mathbf{x}) = \frac{b - \sqrt{b^2 - 4ac}}{2a}, \quad (33\text{b})$$

$$\phi_1(\mathbf{x}) = \tilde{\phi}_1(\mathbf{x}) - s\phi_a(\mathbf{x}), \quad (33\text{c})$$

$$\phi_2(\mathbf{x}) = \tilde{\phi}_2(\mathbf{x}) - (1-s)\phi_a(\mathbf{x}), \quad (33\text{d})$$

and when node a is a mid-side node (Figure 7a), $s = 1/2$, and

$$a = \frac{1}{4}, \quad b(\mathbf{x}) = 1 - \frac{xy}{2} - \frac{y(1-x)}{2} = 1 - \frac{y}{2}, \quad c(\mathbf{x}) = x(1-x)y^2. \quad (34)$$

Figure 7. (a) Quadtree mesh. (a) Hanging nodes; (b) Conforming mesh. The hanging nodes are labeled as a and b, which lead to a non-conforming finite element approximation since C^0 continuity is lost along the edges containing nodes a and b. One approach to restore conformity is to connect the hanging nodes to the element vertices, which however leads to modifications in the quadtree data structure

The shape functions for nodes 1–4 and a can now be written as:

$$\phi_a(\mathbf{x}) = 2 - y - 2\sqrt{\left(1 - \frac{y}{2}\right)^2 - x(1-x)y^2}, \tag{35a}$$

$$\phi_1(\mathbf{x}) = y(1-x) - \frac{\phi_a(\mathbf{x})}{2}, \tag{35b}$$

$$\phi_2(\mathbf{x}) = xy - \frac{\phi_a(\mathbf{x})}{2}, \tag{35c}$$

$$\phi_3(\mathbf{x}) = \tilde{\phi}_3(\mathbf{x}) = x(1-y), \tag{35d}$$

$$\phi_4(\mathbf{x}) = \tilde{\phi}_4(\mathbf{x}) = (1-x)(1-y). \tag{35e}$$

The shape function for node a is plotted in Figure 8. On using the above approach, any number of nodes can be added to an edge [31]. From Eq. (35), the derivatives of ϕ_a are:

$$\frac{\partial \phi_a}{\partial x} = \frac{(1-2x)y^2}{\sqrt{\left(1 - \frac{y}{2}\right)^2 - x(1-x)y^2}}, \quad \frac{\partial \phi_a}{\partial y} = -1 + \frac{1 - \frac{y}{2} + 2xy(1-x)}{\sqrt{\left(1 - \frac{y}{2}\right)^2 - x(1-x)y^2}}, \tag{36}$$

which are unbounded at node a ($x = 1/2$, $y = 1$), but the derivatives are square-integrable in $\Omega = (0,1)^2$.

2.1.4 Polygon with interior nodes

The metric coordinate method is not limited to exterior boundary segments. Interior nodes can also be described by concave interpolants, and shape functions for interior nodes, which are similar to bubble functions in the finite element literature, can be formulated [32]. For an interior node, one point alone is sufficient to define r:

$$r_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\| = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \tag{37}$$
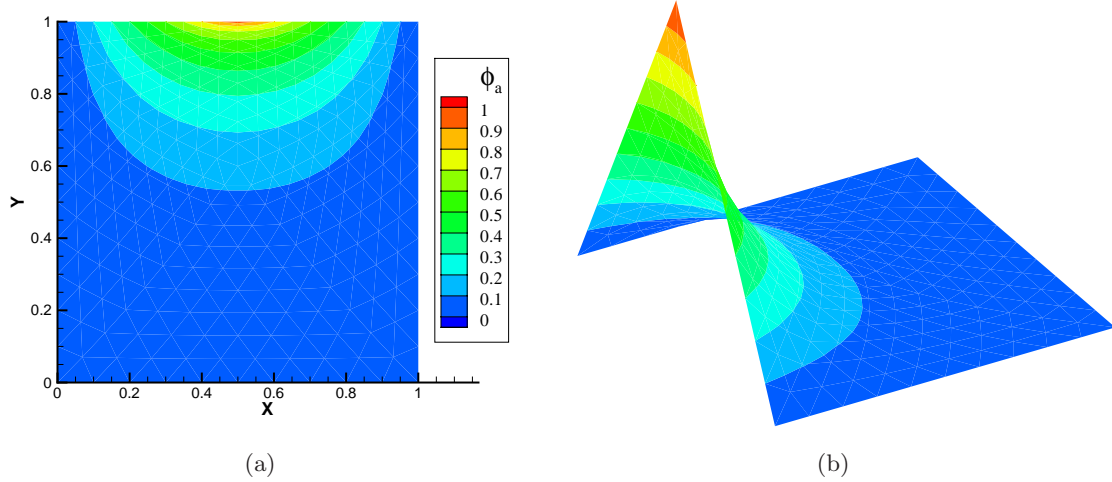
Figure 8. Shape function of mid-side node using the metric coordinate method.
(a) Contour; (b) 3D plot

The weight function in Eq. (8) is defined as

$$\hat{b}_i(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{x}_i\|}, \tag{38}$$

and on taking $w_i(\mathbf{x}) \equiv \hat{b}_i(\mathbf{x})$ in Eq. (6), the shape functions are determined. These shape functions are non-zero and well-defined over the entire domain.

Consider the shape function construction for an interior node that is located at the center of a unit square. The nodes at the vertices of the square are $\{p_1, p_2, p_3, p_4\}$ with coordinates $1(0,0)$, $2(1,0)$, $3(1,1)$, and $4(0,1)$. An interior node p_5 is located at $5(1/2, 1/2)$. Recalling the barycentric form of Wachspress's shape functions, Eq. (8), $\hat{b}_5$ is:

$$\hat{b}_5(\mathbf{x}) = \frac{1}{\sqrt{(x - 1/2)^2 + (y - 1/2)^2}}. \tag{39}$$

On letting $w_i(\mathbf{x}) = \hat{b}_i(\mathbf{x})$ and applying Eq. (2), shape functions that are bounded but not linearly precise are obtained. The expression for $\phi_1(\mathbf{x})$ and $\phi_5(\mathbf{x})$ are:

$$\phi_1(\mathbf{x}) = \frac{(1-x)(1-y)\sqrt{1 - 2x(1-x) - 2y(1-y)}}{\sqrt{2}xy(1-x)(1-y) + \sqrt{1 - 2x(1-x) - 2y(1-y)}}, \tag{40a}$$

$$\phi_5(\mathbf{x}) = \frac{\sqrt{2}xy(1-x)(1-y)}{\sqrt{2}xy(1-x)(1-y) + \sqrt{1 - 2x(1-x) - 2y(1-y)}}. \tag{40b}$$

The shape functions are bounded over the domain, and linear on the edges. The definition of the barycentric weighting $\hat{b}_5$ with the distance measure leads to C^0 behavior, with discontinuity in $\boldsymbol{\nabla}\phi_5(\mathbf{x})$ at the interior node. Alternatively, to ensure C^∞ continuity within the polygon, including at the interior node, let

$$\hat{b}_5(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{x}_5\|^2} = \frac{1}{(x - 1/2)^2 + (y - 1/2)^2}. \tag{41}$$

The shape function for nodes 1 and 5 are:

$$\phi_1(\mathbf{x}) = \frac{(1-x)(1-y)\big(1 - 2x(1-x) - 2y(1-y)\big)}{1 - 2x(1-x) - 2y(1-y) + 2xy(1-x)(1-y)}, \tag{42a}$$

$$\phi_5(\mathbf{x}) = \frac{2xy(1-x)(1-y)}{1 - 2x(1-x) - 2y(1-y) + 2xy(1-x)(1-y)}. \tag{42b}$$

To ensure linear precision and still maintain the interpolation property and linear behavior along the edges, one needs to modify the shape functions as follows:

$$\phi_5(\mathbf{x}) = \frac{2xy(1-x)(1-y)}{1 - 2x(1-x) - 2y(1-y) + 2xy(1-x)(1-y)}, \tag{43a}$$

$$\phi_i(\mathbf{x}) = \tilde{\phi}_i(\mathbf{x}) - \frac{\phi_5(\mathbf{x})}{4}, \quad (i = 1\text{--}4), \tag{43b}$$

where $\tilde{\phi}_i(\mathbf{x})$ are the shape functions for a four-node bilinear element and $\phi_5(\mathbf{x})$ is taken from Eq. (42b). The shape functions $\phi_1(\mathbf{x})$ and $\phi_5(\mathbf{x})$ are plotted in Figure 9; note that $\phi_1(\mathbf{x})$ assumes negative values.
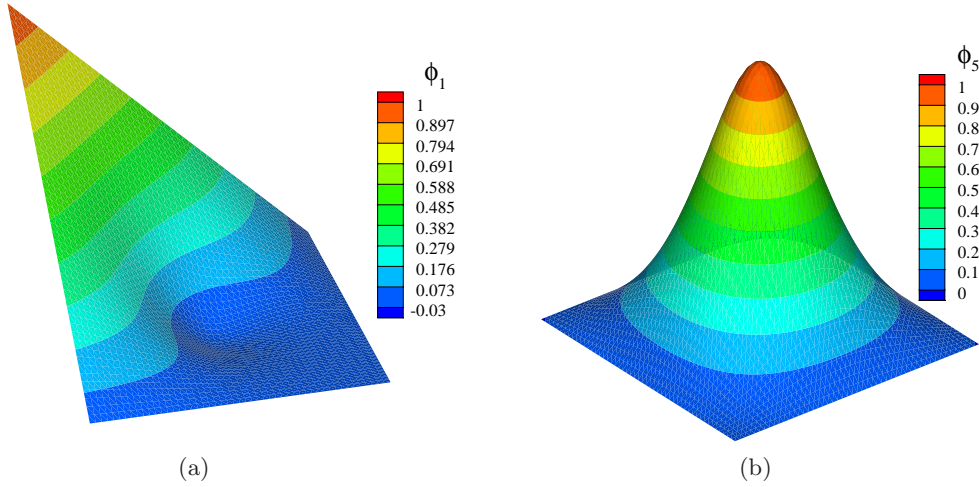
Figure 9. Linearly precise metric coordinate shape functions. (a) Shape function for the boundary node; (b) Shape function for the interior node

2.2 Natural Neighbor Shape Functions on Convex Polygons

As indicated in Section 2.1.1, Wachspress [55] was the first to propose rational basis functions (ratio of polynomial functions) on convex polygonal domains. In [52], a polygonal interpolant based on the concept of *natural neighbors* [46] was proposed. Given a set of nodes in the plane, the Voronoi diagram partitions the space into closest-point regions (Voronoi cells). The Delaunay tessellation is the dual of the Voronoi diagram [37]. In Figure 10, a set of seven nodes is shown and a point p is inserted. The *natural neighbors* of p are defined through the Delaunay circumcircles; if p lies within the circumcircle of a Delaunay triangle t, the nodes that define t are neighbors of p. Two well-known natural neighbor-based interpolants are the Sibson coordinate [46] and the Laplace interpolant [4, 7, 21]. In Figure 10,
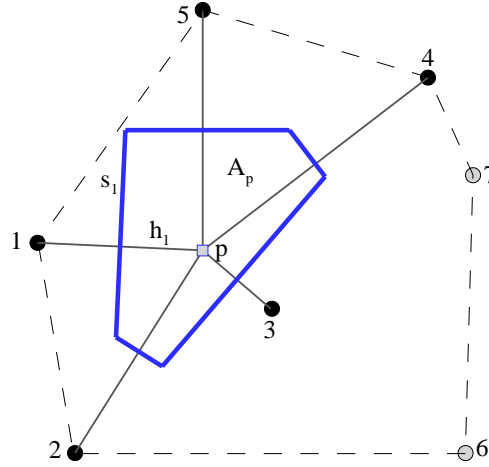
Figure 10. Voronoi cell and natural neighbors (filled circle) of point p

the Voronoi cell of p is depicted, and the Laplace shape functions at p are determined using the Voronoi cell of p. Formally, we define the Laplace shape function as [7]:

$$\phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum\limits_{j=1}^{n} w_j(\mathbf{x})}, \quad w_j(\mathbf{x}) = \frac{s_j(\mathbf{x})}{h_j(\mathbf{x})}, \tag{44}$$

where $s_i(\mathbf{x})$ is the length of the Voronoi edge and $h_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|$ is the Euclidean distance from p to node i (Figure 10). The above form of the Laplace shape function can be established using integral identities [4, 7, 21]. Referring to Figure 10, we can write

$$\int\limits_{A_p} \boldsymbol{\nabla} f \, d\Omega = \int\limits_{\partial A_p} f\mathbf{n} \, d\Gamma \tag{45}$$

by virtue of Gauss's theorem. On setting $f = 1$, we have

$$\int\limits_{\partial A_p} \mathbf{n} \, d\Gamma = \mathbf{0}, \tag{46}$$

which is the Minkowski theorem [20] for convex polytopes. On discretizing the above integral over the Voronoi cell of point p, we obtain [21]

$$\sum_{i=1}^{n} \frac{\mathbf{x}_i - \mathbf{x}}{h_i(\mathbf{x})} s_i(\mathbf{x}) = \mathbf{0}, \tag{47}$$

and therefore

$$\mathbf{x} = \sum_{i=1}^{n} \phi_i \mathbf{x}_i, \quad \phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum\limits_{j=1}^{n} w_j(\mathbf{x})}, \quad w_i(\mathbf{x}) = \frac{s_i(\mathbf{x})}{h_i(\mathbf{x})}, \tag{48}$$

which is precisely (linear precision) Eq. (4). The Laplace shape function satisfies all the properties indicated in Equations (2)–(5) [7]. For further details on the prior developments

related to Laplace shape functions, the interested reader can refer to [49] and the review article on natural neighbor Galerkin methods by Cueto *et al.* [9].

We now consider Laplace shape functions on regular n-gons, which provides a natural means to derive polygonal shape functions on irregular polygons. In a simplex-partition of a regular polygon, all triangles have a common center and the nodes all lie on the same circumcircle. The circumcircle of a polygon is the circle inside which the polygon can be inscribed, and the radius of the circle is the circumradius R. All regular polygons and regular polyhedra possess a circumradius. For the regular polygons shown in Figure 11, the vertex-nodes lie on the same circumcircle, and hence all the nodes of a polygon are the *natural neighbors* for any point in Ω_0. Since $\phi_i \equiv \phi_i(\boldsymbol{\xi})$ is piece-wise linear on the boundary $\partial\Omega_0$, the isoparametric mapping given in Eq. (48) is used to obtain the shape functions on irregular (convex) polygon. In Figures 12–12c, the Voronoi polygon for a point p in a pentagon, hexagon and a nonagon are shown; the vertices of the Voronoi polygon are located at the center of the Delaunay circumcircle of triangle (p, p_i, p_j). Since $\phi_i \equiv \phi_i(\boldsymbol{\xi})$ is piece-wise linear on the boundary $\partial\Omega_0$, the isoparametric mapping given in Eq. (48) is used to obtain the shape functions on irregular (convex) polygon. The measures s_i and h_i for a regular hexagon are shown in Figure 12d.
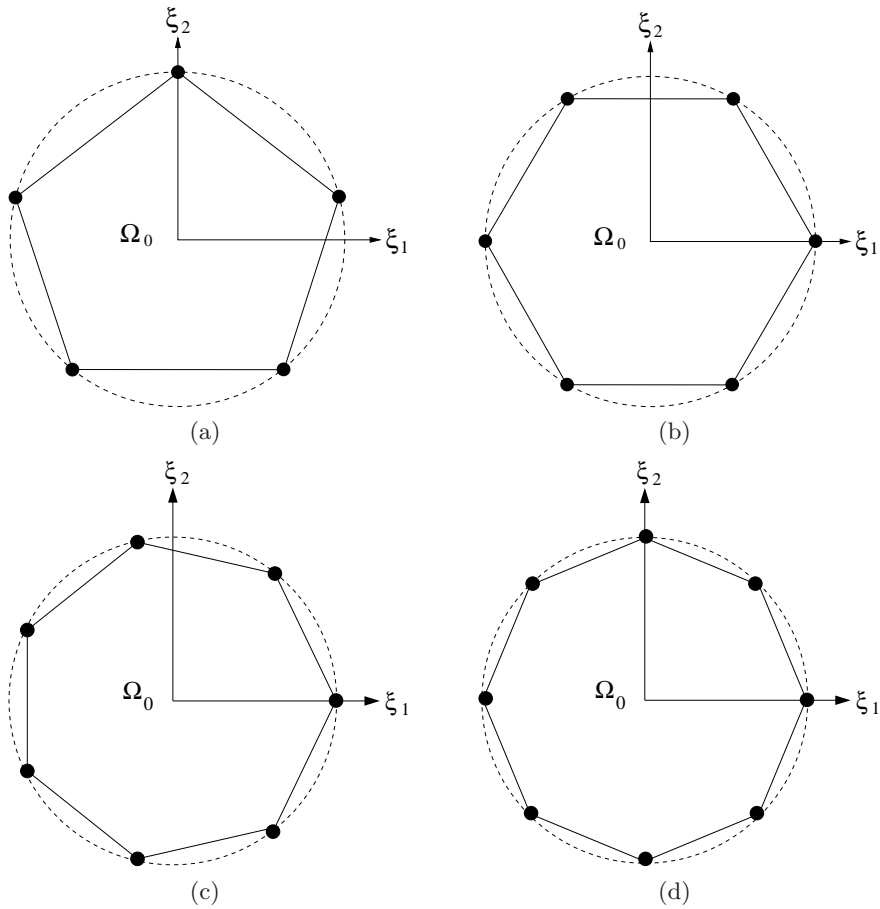
Figure 11. Reference elements (pentagon, hexagon, heptagon, and octagon) in 2D. This is a generalization of triangular- and quadrilateral-FEM to irregular (convex) n-gons
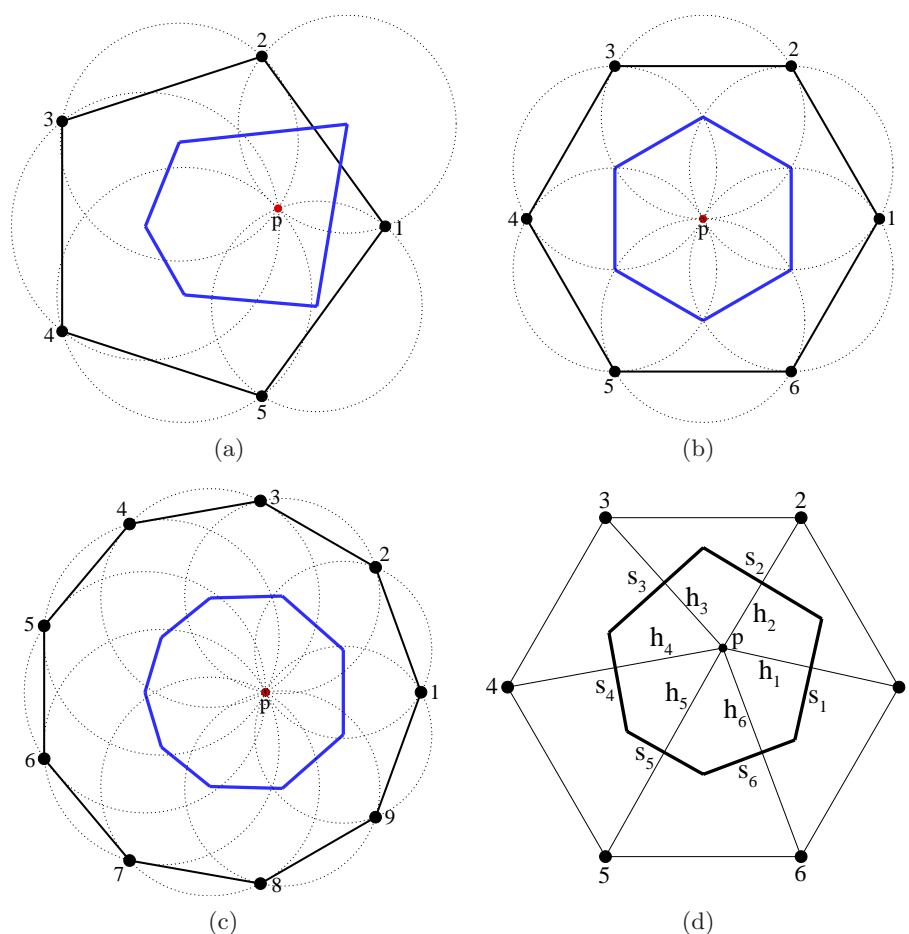
Figure 12. Laplace shape function. (a), (b), (c) Delaunay circumcircles and
Voronoi cell for p in a pentagon, hexagon, and nonagon; (d) Length
measures in the definition of the Laplace shape function

The above construction is a generalization of finite elements to convex n-gons. The barycentric coordinate for a triangle are unique. On a bi-unit square, Laplace, Wachspress, and finite element shape functions are identical, but the mean value coordinate is distinct [15]. Different barycentric forms on the square lead to bilinear shape functions, which are the product of one-dimensional Lagrange interpolants. The use of barycentric forms in 1D has been recently studied. In [5], the construction and merits (compact representation and computational efficiency) of barycentric one-dimensional Lagrange interpolation was presented. For $(n-1)$th order precision in 1D, consider n nodes that are located at $x_1, x_2, \ldots, x_n$. The shape function $\phi_i(x)$ can be written as in Eq. (6) with the weight function $w_i(x)$ given by [5]

$$w_i(x) = \frac{b_i}{x - x_i}, \quad b_i = \frac{1}{\prod_{k \neq i} (x_i - x_k)}, \quad i = 1, 2, \ldots, n. \tag{49}$$

In the case of linear interpolation in 1D, the above weights are equivalent to those obtained using the Laplace shape function [51]. Laplace and Wachspress shape functions are the same on regular polygons [52]. Numerical tests have also revealed that on circumscribable

polygons (vertices lies on a circle), the Sibson coordinate [46], Laplace, and Wachspress shape functions are all identical. The *discrete harmonic weight*, which was introduced by Pinkall and Polthier [38] in the computation of minimal surfaces, is the solution to the minimization of the Dirichlet integral. The expression for the Laplace shape function [7] is also identical to the *discrete harmonic weight*, which is proven below.

Proof. Referring to Figure 13, $s_j(\mathbf{x})$ is the Voronoi edge length associated with node p_j and point p, and the vertices p_i, p_j and p_k. The Voronoi edge length can also be written in terms of the geometric measures shown in Figure 5. The circumradius of a triangle defined by vertices (p, p_i, p_j) is:

$$R_{ij}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_i\| \|\mathbf{x} - \mathbf{x}_j\| \|\mathbf{x}_i - \mathbf{x}_j\|}{4A(p, p_i, p_j)}, \tag{50}$$

and the Voronoi edge length $s_j(\mathbf{x})$ is then

$$s_j(\mathbf{x}) = \sqrt{R_{ij}^2(\mathbf{x}) - \frac{1}{4}\|\mathbf{x} - \mathbf{x}_j\|^2} + \sqrt{R_{jk}^2(\mathbf{x}) - \frac{1}{4}\|\mathbf{x} - \mathbf{x}_j\|^2}. \tag{51}$$

The Laplace weight function $w_j(\mathbf{x}) = s_j(\mathbf{x})/\|\mathbf{x} - \mathbf{x}_j\|$ that is defined in Eq. (44) is now given by

$$
\begin{aligned}
4w_j(\mathbf{x}) &= \frac{\sqrt{(\|\mathbf{x} - \mathbf{x}_i\| \|\mathbf{x}_i - \mathbf{x}_j\|)^2 - 4A^2(p, p_i, p_j)}}{A(p, p_i, p_j)} \\
&\quad + \frac{\sqrt{(\|\mathbf{x} - \mathbf{x}_k\| \|\mathbf{x}_j - \mathbf{x}_k\|)^2 - 4A^2(p, p_j, p_k)}}{A(p, p_j, p_k)} \\
&= \frac{\|\mathbf{x} - \mathbf{x}_i\| \|\mathbf{x}_i - \mathbf{x}_j\| \cos\beta_i}{A(p, p_i, p_j)} + \frac{\|\mathbf{x} - \mathbf{x}_k\| \|\mathbf{x}_j - \mathbf{x}_k\| \cos\gamma_j}{A(p, p_j, p_k)},
\end{aligned}
\tag{52}
$$

where the angles β_i and γ_j are shown in Figure 13. Since

$$A(p, p_i, p_j) = \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\| \|\mathbf{x} - \mathbf{x}_i\| \sin\beta_i, \quad A(p, p_j, p_k) = \frac{1}{2}\|\mathbf{x}_j - \mathbf{x}_k\| \|\mathbf{x} - \mathbf{x}_k\| \sin\gamma_j, \tag{53}$$

Eq. (52) can be written as

$$4w_j(\mathbf{x}) = \frac{2\cos\beta_i}{\sin\beta_i} + \frac{2\cos\gamma_j}{\sin\gamma_j}, \tag{54}$$

and therefore

$$w_j(\mathbf{x}) = \frac{\cot\beta_i + \cot\gamma_j}{2}, \tag{55}$$

which is the *discrete harmonic weight* [38]. □

The above construction on convex polygons has been adapted to conforming approximation on *weakly* convex polygons (quadtree meshes) [53]. Referring once again to the quadtree element A shown in Figure 7a, the Laplace shape function ϕ_a for node a is constructed. An isoparametric transformation of the pentagon in Figure 11a to element A is used. The plot of the shape function is illustrated in Figure 14. The shape function ϕ_a is C^∞ within element A and is C^0 (piece-wise linear behavior) along the edge 1–2 [53].
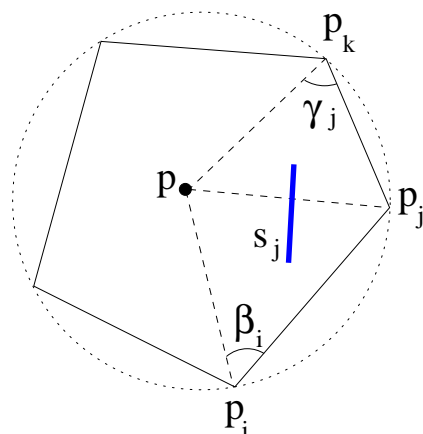
Figure 13. Equivalence of Laplace and discrete harmonic weight functions

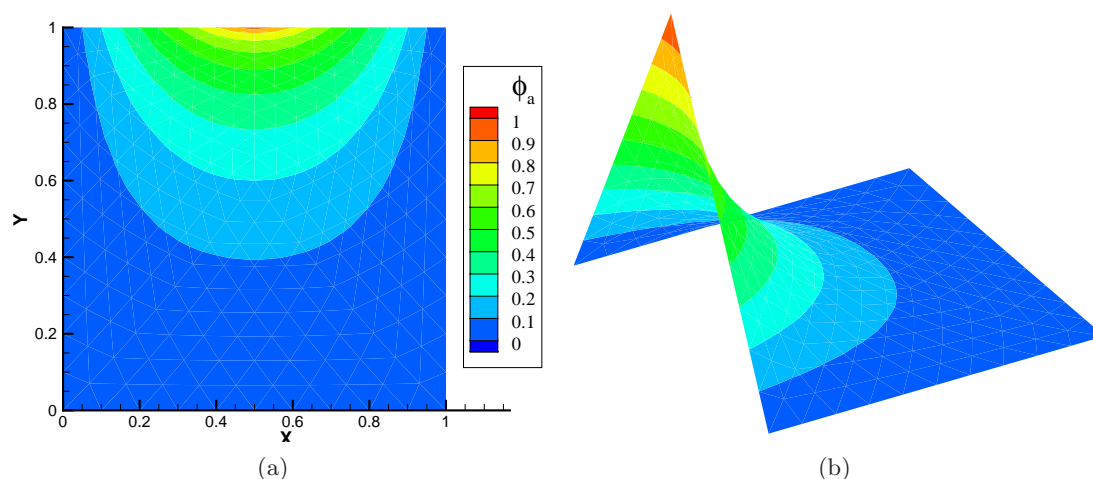(a)   (b)

Figure 14. Shape function of mid-side node using an isoparametric transformation.
(a) Contour; (b) 3D plot

The conforming Laplace interpolant on quadtree meshes has been used to solve Poisson problems with sharp gradients and corner singularities [53]. However, an issue that demands further investigation is that the inverse of the Jacobian does not exist ($\det \mathbf{J} = 0$) at $\mathbf{x} = \mathbf{x}_a$ (Figure 14). This degeneracy arises since a convex polygon is mapped to a square with multi-nodes, and the derivatives blow up at node a. The patch test is passed to $\mathcal{O}(10^{-10})$ on even refined grids [53], which suggests that the numerical integration errors are bounded. Furthermore, an eigenanalysis in Section 4.2 reveals that the stiffness matrix of the quadtree element A in Figure 7a has no spurious modes.

2.3 Maximum Entropy Approximant

In the previous sections, we have described the construction of different polygonal interpolants. The restrictions of linear completeness and partition of unity in Eqs. (2) and (4) do not prescribe unique shape functions for any polygon with more than three nodes. By deriving shape functions from a constrained optimization problem instead of the approach presented in Sections 2.1 and 2.2, a more general methodology to construct scattered data

approximants is conceivable. To this end, elements from approximation theory and information theory are utilized, which can potentially lead to new insights and the ability to design interpolation and approximation schemes by using suitable constraints in optimization problems. In [50], the least-biased approximant or equivalently the one that maximizes the Shannon entropy [24] is derived. We recently became aware of the work of Arroyo and Ortiz [3] who have used the maximum entropy principle to construct local meshfree approximation schemes.

In information theory, the notion of entropy as a measure of uncertainty or incomplete knowledge was introduced by Shannon [44]. The Shannon entropy of a discrete probability distribution is:

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p(x_i) \log\big(p(x_i)\big), \tag{56}$$

where $p_i \equiv p(x_i)$ is the probability of the occurrence of the event x_i, and the above form of the function $H(\cdot)$ satisfies the axiomatic requirements of an uncertainty measure [28]. Jaynes [24] proposed the maximum entropy (uncertainty) principle as a means for least-biased statistical inference when insufficient information is available. It provides a maximally non-committal distribution that is consistent with the given data. Both the maximum entropy concentration theorem and Wallis's combinatorial argument substantiate this claim [25, 26, 41]. The MAXENT principle has been applied to many problems in the applied sciences and engineering [27]. The Shannon measure of uncertainty is one of the first entropy measures proposed within probability theory. More recently, several general families of entropy measures have been introduced, among which, the Rényi and Tsallis measures [40, 54] are two of the most prominent generalizations of Shannon entropy.

As an example of a MAXENT formulation, consider a random variable x with possible non-negative real values $x_1, x_2, \ldots, x_n$. Assume that the probability p_i of value x_i is unknown, but the expected value $E(x)$ is known. Then, we determine p_i by solving the following problem [24]:

$$\max_{\mathbf{p}} \left(H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log p_i \right), \tag{57a}$$

$$\sum_{i=1}^{n} p_i = 1, \tag{57b}$$

$$\sum_{i=1}^{n} p_i x_i = E(x). \tag{57c}$$

First, we establish a link between the maximum entropy formulation and the data approximation problem. Consider a scattered set of nodes in $\mathbf{R}^2$ (Figure 1), with $\mathbf{x}_i$ denoting the nodal coordinate of node i. Consider the introduction of a point p with coordinate $\mathbf{x}$ within the convex hull of the set of nodes (Figure 1). The shape function value ϕ_i at the point p is viewed as the *probability of influence of node i at p*; on a convex polygon, as p approaches node i ($\mathbf{x} \to \mathbf{x}_i$), $\phi_i \to 1$ and $\phi_j \to 0$ for $j \neq i$. The maximum entropy formulation for the shape functions in $\mathbf{R}^2$ (extends to $\mathbf{R}^3$) is as follows: find $\phi_i \geq 0$ (for ease of notation, we suppress the spatial dependence in ϕ_i) as the solution of the following constrained optimization problem:

$$\max_{\boldsymbol{\phi}} \left(H(\boldsymbol{\phi}) = -\sum_{i=1}^{n} \phi_i \log \phi_i \right), \tag{58a}$$

subject to the linear reproducing conditions:

$$\sum_{i=1}^{n} \phi_i = 1, \tag{58b}$$

$$\sum_{i=1}^{n} \phi_i x_i = x, \tag{58c}$$

$$\sum_{i=1}^{n} \phi_i y_i = y. \tag{58d}$$

Let λ_r $(r = 0, 1, 2)$ be the Lagrange multipliers associated with the three constraints, and we set the first variation of the augmented Lagrangian to zero [24]:

$$\delta \left[\sum_{i=1}^{n} -\phi_i \log \phi_i + \lambda_0 \left(1 - \sum_{i=1}^{n} \phi_i \right) + \lambda_1 \left(x - \sum_{i=1}^{n} \phi_i x_i \right) + \lambda_2 \left(y - \sum_{i=1}^{n} \phi_i y_i \right) \right] = 0. \tag{59}$$

On carrying out the above variation and letting $\lambda_0 = \log Z - 1$ [24] (Z is used to denote the partition function in statistical mechanics), the shape functions ϕ_i can be written as:

$$\phi_i = \frac{e^{-\lambda_1 x_i - \lambda_2 y_i}}{Z} \equiv \frac{w_i}{\sum_{j=1}^{n} w_j}, \quad Z = \sum_{j=1}^{n} e^{-\lambda_1 x_j - \lambda_2 y_j}, \tag{60}$$

where we note that the MAXENT shape functions are ratios of exponential functions, and they assume the partition of unity form form given in Eq. (6). The MAXENT shape functions are also continuously differentiable in the interior of the domain. On substituting Eq. (60) in Eq. (58), we obtain the following non-linear equations for λ_1 and λ_2:

$$f_1(\lambda_1, \lambda_2) = \frac{\sum_{i=1}^{n} e^{-\lambda_1 x_i - \lambda_2 y_i} x_i}{Z} - x = 0, \tag{61a}$$

$$f_2(\lambda_1, \lambda_2) = \frac{\sum_{i=1}^{n} e^{-\lambda_1 x_i - \lambda_2 y_i} y_i}{Z} - y = 0, \tag{61b}$$

and on solving for λ_1 and λ_2, the shape functions ϕ_i are obtained from Eq. (60).

Using Newton's method directly to solve the non-linear equations in Eq. (61) will, in general, not lead to convergence. Agmon *et al.* [1, 2] recast the original (primal) problem into one (dual problem) in which the Lagrange multipliers are determined as the set that minimizes a convex *scalar potential function* $F(\lambda_1, \lambda_2)$. Letting

$$\tilde{x}_i = x_i - x, \quad \tilde{y}_i = y_i - y, \tag{62}$$

the linear reproducing conditions in Eq. (58) become

$$\sum_{i=1}^{n} \phi_i \tilde{x}_i = 0, \quad \sum_{i=1}^{n} \phi_i \tilde{y}_i = 0. \tag{63}$$

Now,

$$\phi_i = \frac{e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i}}{\tilde{Z}}, \quad \tilde{Z}(\lambda_1, \lambda_2) = \sum_{j=1}^{n} e^{-\lambda_1 \tilde{x}_j - \lambda_2 \tilde{y}_j}, \tag{64}$$

and instead of Eq. (61), we have [2]

$$f_1(\lambda_1, \lambda_2) = \frac{\partial(\log \tilde{Z})}{\partial \lambda_1} = -\frac{\sum\limits_{i=1}^{n} e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{x}_i}{\tilde{Z}} = 0, \tag{65a}$$

$$f_2(\lambda_1, \lambda_2) = \frac{\partial(\log \tilde{Z})}{\partial \lambda_2} = -\frac{\sum\limits_{i=1}^{n} e^{-\lambda_1 \tilde{x}_i - \lambda_2 \tilde{y}_i} \tilde{y}_i}{\tilde{Z}} = 0. \tag{65b}$$

The vector field $\mathbf{f}$ is conservative with scalar potential $\mathbf{f} = \boldsymbol{\nabla} \log \tilde{Z}(\lambda_1, \lambda_2) \equiv \boldsymbol{\nabla} F(\lambda_1, \lambda_2)$ [2], and hence the dual problem is: find $(\lambda_1^t, \lambda_2^t)$ such that

$$F = \log \tilde{Z}(\lambda_1^t, \lambda_2^t) \text{ is minimized.} \tag{66}$$

The algorithm to solve Eq. (66) consists of an initial guess, a suitable search direction (steepest descent), and a scalar α that provides a measure of the distance to be traversed along the search direction at each iteration step. Let $\boldsymbol{\lambda}^k \equiv (\lambda_1^k, \lambda_2^k)^T$ be the solution at the kth iteration; the initial guess is $\boldsymbol{\lambda}^0$. Given the solution at iteration k, the update for the Lagrange multipliers is [1]:

$$\lambda_r^{k+1} = \lambda_r^k + \alpha \Delta \lambda_r^k, \quad \Delta \lambda_r^k = -\boldsymbol{\nabla} F, \tag{67}$$

where α is determined through the condition that $F(\lambda_1^{k+1}, \lambda_2^{k+1})$ attains a minimum along the search direction. A search algorithm to bracket the minimum [39] of F is used to find α with a starting guess $\boldsymbol{\lambda}^0 = \mathbf{0}$ for all points in the domain. The convergence criterion at iteration k is: $\|\boldsymbol{\nabla} F\|^{\{k\}} < \epsilon$; $\epsilon = 10^{-7}$ is used in the numerical computations. Further details on the numerical algorithm and its implementation are provided in [50]. As an alternative to the method of steepest descent that was adopted in [50], the use of non-linear conjugate gradient methods [36] to compute the search direction leads to better efficiency and faster convergence.

The maximum entropy shape functions satisfy all the properties indicated in Equations (2)–(5). They are identical to bilinear finite element shape functions on a square, and are smooth and bounded within the convex hull of a set of nodes [50]. To illustrate a simple closed-form computation, consider one-dimensional approximation in $\Omega = (0, 1)$ with three nodes located at $1(0, 0)$, $2(1/2, 0)$, and $3(1, 0)$. The MAXENT solution using Eq. (60) is:

$$\phi_i = \frac{e^{-\lambda_1 x_i}}{\sum\limits_{j=1}^{3} e^{-\lambda_1 x_j}}, \tag{68}$$

where λ_1 is the solution of the equation

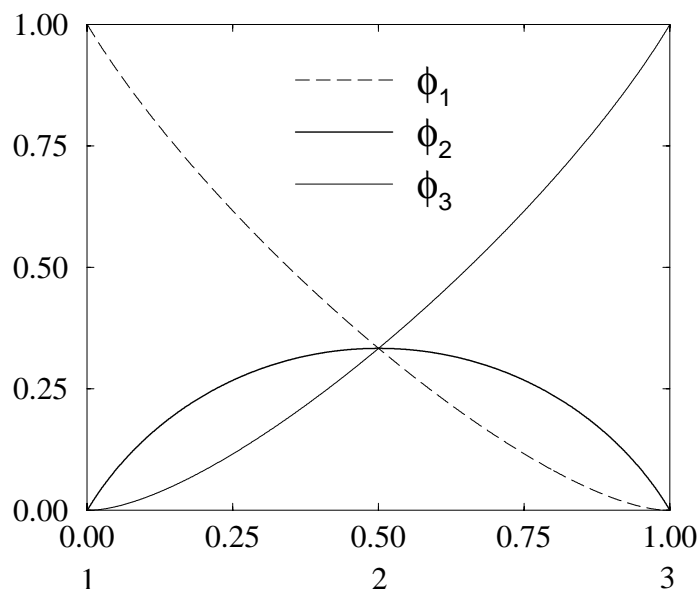$$\sum_{i=1}^{3} \phi_i x_i = x. \tag{69}$$

Figure 15. Maximum entropy shape functions in one-dimension

On carrying out the algebraic computations, the following solution for ϕ_i is obtained:

$$\phi_1(x) = \frac{1}{1 + \eta + \eta^2}, \quad \phi_2(x) = \frac{\eta}{1 + \eta + \eta^2}, \quad \phi_3(x) = \frac{\eta^2}{1 + \eta + \eta^2}, \qquad (70a)$$

where

$$\eta \equiv \eta(x) = \frac{2x - 1 + \sqrt{12x(1 - x) + 1}}{4(1 - x)}. \qquad (70b)$$

These shape functions are depicted in Figure 15. Note that $\phi_i(1/2) = 1/3 \; \forall i$, and since interpolation is not attained at $x = 1/2$, an approximant is obtained. Referring once again to the quadtree element in Figure 7a, the MAXENT shape function for the *hanging node* is constructed. The shape function for node a and the maximum entropy distribution are shown in Figure 16. By construction, the maximum entropy formalism yields $\{\phi_i\}$ that are smooth and also the *flattest possible distribution*. Since along the top edge, the one-dimensional expressions given in Eq. (70) are valid, we obtain smoothness but lose interpolation at node a as well as the piece-wise linear behavior along the edge containing the *hanging node*.

In Section 2.1.4, the metric coordinate method was used to determine shape functions for polygons with interior nodes. The maximum entropy approach provides an elegant means to determine shape functions within a polygonal domain with nodes at the vertices and in its interior (Figure 1e). On convex polygons with only vertex nodes, an interpolant is obtained, but on a *weakly* convex polygon (quadtree mesh) or in a polygon with interior nodes, an approximant is realized. In this formulation, only the nodal coordinates are used, and neither the connectivity of the nodes nor any user-defined parameters are required—the defining characteristics of a *meshfree approximant*. The interpolation and piece-wise linear behavior on the boundary of convex polygons permits this approximation to be compatible on the boundary with finite elements or natural-neigbor based interpolation schemes [9].
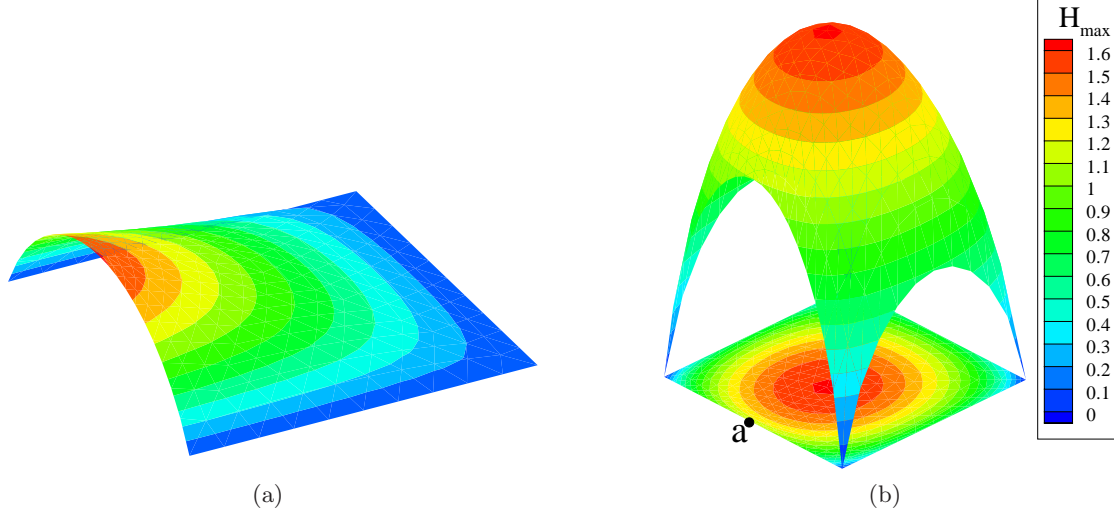
Figure 16. `MAXENT` computations on element A. (a) ϕ_a; (b) Maximum entropy distribution

The maximum entropy shape functions can also be viewed as a particular form of asymmetric linearly precise Gaussian radial basis functions [6]. As in the previous example, where interpolation was not met at the mid-side node, the `MAXENT` shape functions are non-interpolating at the interior node. The square with an interior node in its center is once again considered. Plots of the shape function for the interior node are shown in Figure 17. The shape function is smooth, $\phi_5(\mathbf{x})$ vanishes on the boundary of the square, and at the interior nodal location, all the shape functions assume the same value: $\phi_i(1/2, 1/2) = 0.2$ ($i = 1$–5).

3 NUMERICAL INTEGRATION OF THE WEAK FORM

Consider the Poisson equation with homogeneous essential boundary conditions:

$$-\nabla^2 u = f \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega. \tag{71}$$

The weak form of the above boundary-value problem is:

$$\int_\Omega \boldsymbol{\nabla}\delta u \cdot \boldsymbol{\nabla} u \, d\Omega = \int_\Omega \delta u f \, d\Omega \quad \forall \delta u \in V_0, \tag{72}$$

where u and δu are trial and test approximations, respectively, and V_0 is the Sobolev space with functions that have square-integrable derivatives in Ω and vanishing values on $\partial\Omega$.

In a Galerkin procedure, the trial and test approximations are chosen from a finite-dimensional space $V_0^h \subset V_0$. The trial function u^h and the test function δu^h are of the form given in Eq. (1). On substituting these in Eq. (72), and using the arbitrariness of nodal variations, we obtain the following system of linear equations:

$$\mathbf{Kd} = \mathbf{f}, \quad K_{ij} = \int_\Omega \boldsymbol{\nabla}\phi_i \cdot \boldsymbol{\nabla}\phi_j \, d\Omega, \quad f_i = \int_\Omega \phi_i f \, d\Omega, \tag{73}$$
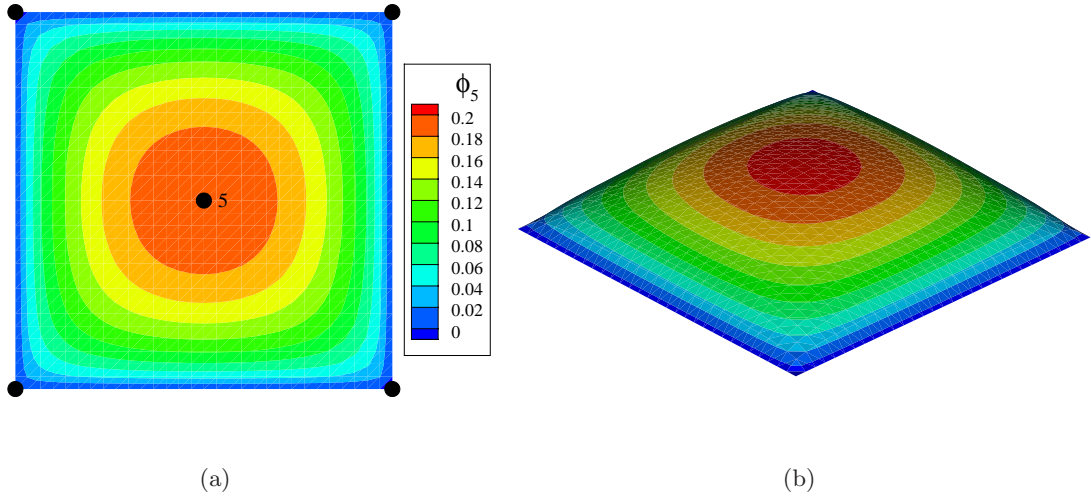
Figure 17. Maximum entropy shape function for an interior node. (a) Contour; (b) 3D plot

where $\mathbf{d}$ is the vector of nodal unknowns. Let $\bar{\Omega}$ be discretized by a set of non-overlapping polygonal elements, i.e., $\bar{\Omega} = \cup_{k=1}^{M} V_k$. Numerical integration on V_k is required to compute K_{ij} and f_i in Eq. (73). To this end, three approaches are possible:

1. Integration on the polygonal element V_k by partitioning it into triangles and then using standard quadrature rules on a triangle.

2. Partitioning the reference element in Figure 11 into n triangles and then performing numerical quadrature on each triangle; or

3. Development of polynomial-precision quadrature rules on irregular polygons. On combining elements of group theory and numerical optimization, a class of cubature rules for triangles was proposed by Wandzura and Xiao [56], and the extension of this approach for the design of cubature rules on irregular n-gons holds promise.

In this study, the first and second approaches are tested, i.e., numerical integration is performed by sub-dividing the physical element or the reference element, into triangles. Since the shape functions in the Wachspress, mean value, metric coordinate, and `MAXENT` interpolation methods are directly computed in the Cartesian coordinate system, the use of the physical element for the integration is a natural choice, whereas for the Laplace shape functions, integration on the reference element is suitable. If the numerical integration is carried out by partitioning the physical element, then the integration of a scalar function f over V_k is written as [52]:

$$\int_{V_k} f \, d\Omega = \sum_{j=1}^{n} \int_{V_k^{\triangle j}} f \, d\Omega = \sum_{j=1}^{n} \int_0^1 \int_0^{1-\xi} f |\mathbf{J}_1^j| \, d\xi d\eta, \tag{74}$$

and a nsp-point quadrature rule on each reference right-triangle is used to compute the last integral. In the above case, for a given quadrature point, $\mathbf{x} = \sum_{i=1}^{3} N_i \mathbf{x}_i$ is used, where N_i are the finite element shape functions for a three-node triangle (Figure 18a).

If the numerical integration is performed by partitioning the reference elements shown in Figure 11, then the following transformations are performed to integrate f [52]:

$$\int\limits_{V_k} f\, d\Omega = \int\limits_{\Omega_0} f\, |\mathbf{J}_2| d\Omega = \sum_{j=1}^{n} \int\limits_{\Omega_0^{\triangle j}} f\, |\mathbf{J}_2| d\Omega = \sum_{j=1}^{n} \int_0^1 \int_0^{1-\xi} f |\mathbf{J}_1^j||\mathbf{J}_2|\, d\xi d\eta. \tag{75}$$

In Eq. (75), the local coordinate $\boldsymbol{\xi}$ in the reference element is obtained via the isoparametric transformation $\boldsymbol{\xi} = \sum_{i=1}^3 N_i \boldsymbol{\xi}_i$, which enables the Laplace shape functions to be readily computed. This approach parallels that used in classical finite elements. The sequence of transformations used in this scheme are illustrated in Figure 18b.
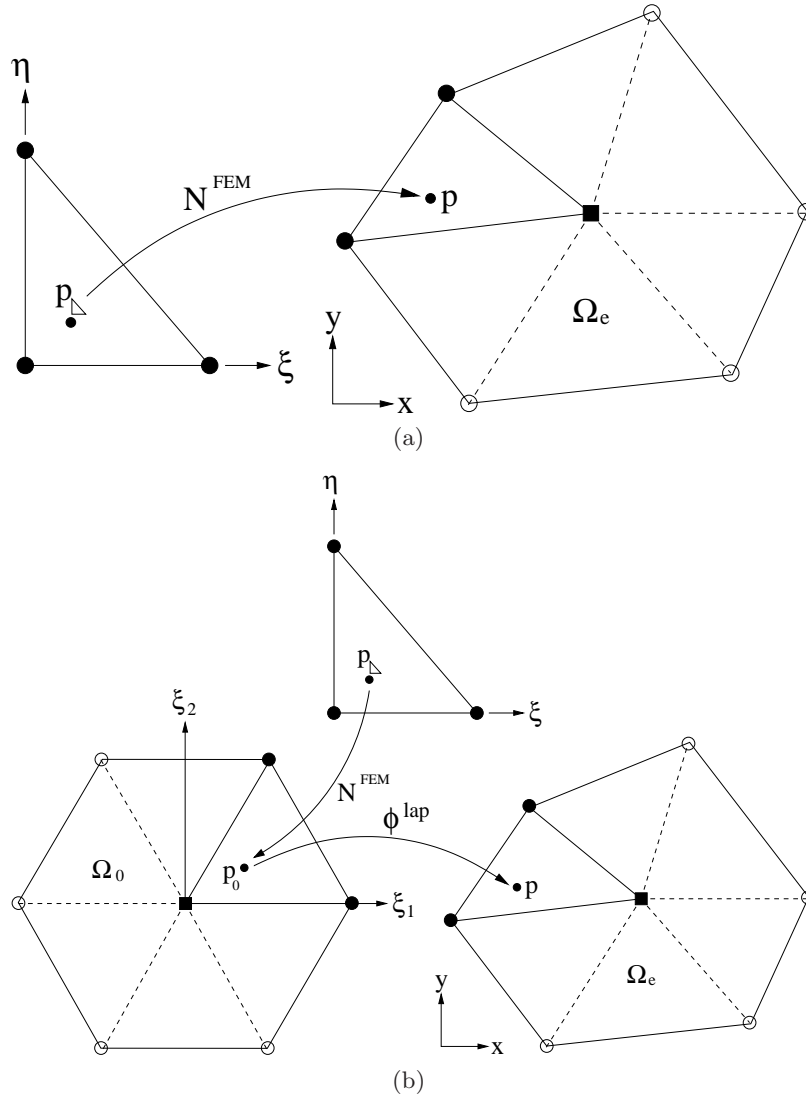
Figure 18. Numerical integration schemes. (a) Partition of the physical element; (b) Partition of the reference element [52]

4 NUMERICAL RESULTS

The construction of polygonal shape functions and their use within a Galerkin method is explored. First, the shape functions are plotted on a few polygonal domains, and then numerical results for the patch test on polygonal meshes are presented. In the latter, $L^2(\Omega)$ and energy error norms are used:

$$\|u - u^h\|_{L^2(\Omega)} = \sqrt{\int_\Omega \left(u - u^h\right)^2 d\Omega}, \quad \|u - u^h\|_{E(\Omega)} = \sqrt{\int_\Omega \left(u_{,i} - u^h_{,i}\right)\left(u_{,i} - u^h_{,i}\right) d\Omega}, \quad (76)$$

where $(\cdot)_{,i} = \frac{\partial(\cdot)}{\partial x_i}$, and u and u^h are the exact and the finite element solutions, respectively.

4.1 Polygonal Shape Functions

The Wachspress, mean-value, metric, Laplace, and maximum entropy shape functions are denoted by ϕ_i^{wsp}, ϕ_i^{mvc}, ϕ_i^{mcm}, ϕ_i^{lap}, and ϕ_i^{mxt}, respectively. Consider the pentagonal element shown in Figure 1c. In Figure 19, the shape function for node 5 using the different interpolation methods is plotted. The maximum value of ϕ_5 in each case is unity, and ϕ_5 is linear along the edges that include node 1.

As a second example, consider the *weakly* convex eight-noded element shown in Figure 1d. The Laplace and metric coordinate methods are used to compute the shape functions. In Figure 20, the shape functions for the mid-side node and for a side-node are depicted.

Shape function computations for concave elements are performed using the metric coordinate method and Floater's mean value coordinates. In the numerical implementation, the mean value coordinates are evaluated using the weight function expressions presented by Hormann [22]:

$$\phi_j(\mathbf{x}) = \frac{w_j(\mathbf{x})}{\sum\limits_{k=1}^{n} w_k(\mathbf{x})}, \quad w_j(\mathbf{x}) = \frac{\tan(\alpha_i/2) + \tan(\alpha_j/2)}{\|\mathbf{x} - \mathbf{x}_j\|}, \tag{77a}$$

$$\tan\frac{\alpha_i}{2} = \frac{\|\mathbf{x} - \mathbf{x}_i\|\|\mathbf{x} - \mathbf{x}_j\| - (\mathbf{x} - \mathbf{x}_i) \cdot (\mathbf{x} - \mathbf{x}_j)}{2A(p_i, p_j, p)}, \tag{77b}$$

$$\tan\frac{\alpha_j}{2} = \frac{\|\mathbf{x} - \mathbf{x}_j\|\|\mathbf{x} - \mathbf{x}_k\| - (\mathbf{x} - \mathbf{x}_j) \cdot (\mathbf{x} - \mathbf{x}_k)}{2A(p_j, p_k, p)}, \tag{77c}$$

where the angles α_i and α_j are shown in Figure 4b. Consider a four-node element with nodes located at: $1(0,0)$, $2(1,0)$, $3(1,1)$, and $4(0.5,.25)$. In Figure 21, the shape function for node 4 is plotted for the above element as well as for the concave element depicted in Figure 1f. Using the metric coordinate method, $\phi_4(\mathbf{x})$ is bounded in Figure 21a, but $\phi_4(\mathbf{x})$ exceeds unity in the vicinity of node 4 in the plot shown in Figure 21b. In Figures 21c and 21d, the plots using the mean value coordinates are illustrated.

4.2 Patch Test

The patch test for the Laplacian operator is carried out: $\nabla^2 u = 0$ in $\Omega = (-1, 1)^2$, with $u = g(\mathbf{x}) = x_1 + x_2$ imposed on the boundary of the bi-unit square. The exact solution is: $u(\mathbf{x}) = x_1 + x_2$. In the analyses, different types of meshes are considered. In Figures 22a–22c, convex polygonal elements are shown, whereas quadtree meshes are illustrated in Figures 22d–22f. The hanging nodes are labeled in the above figures. Meshes with concave
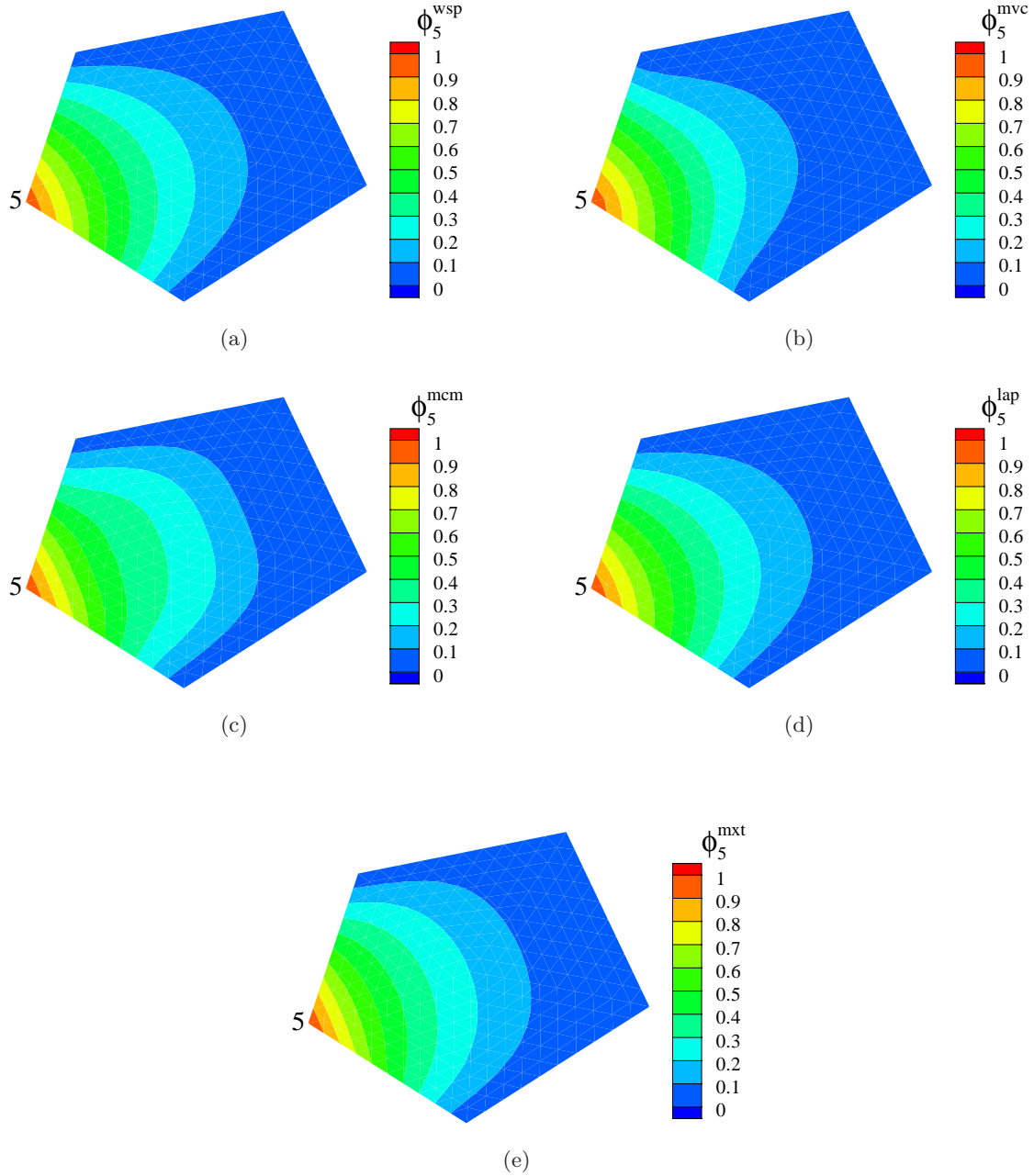
Figure 19. Shape function for node 5 in a pentagon. (a) Wachspress; (b) Mean value; (c) Metric; (d) Laplace; (e) `MAXENT`

elements are presented in Figures 22g and 22h. In the numerical computations, a twenty-five point ($nsp = 25$) quadrature rule is used within each partitioned sub-triangle [52]. For the Laplace shape functions, numerical integration is done on the reference element (Figure 18b), whereas for all other interpolation schemes, the physical element (Figure 18a) is used in the numerical integration. The relative L^2 and energy error norms for the convex polygonal elements are presented in Table 2. The Laplace interpolant provides the most accurate results on the patch test with $\mathcal{O}(10^{-8})$–$\mathcal{O}(10^{-7})$ errors.
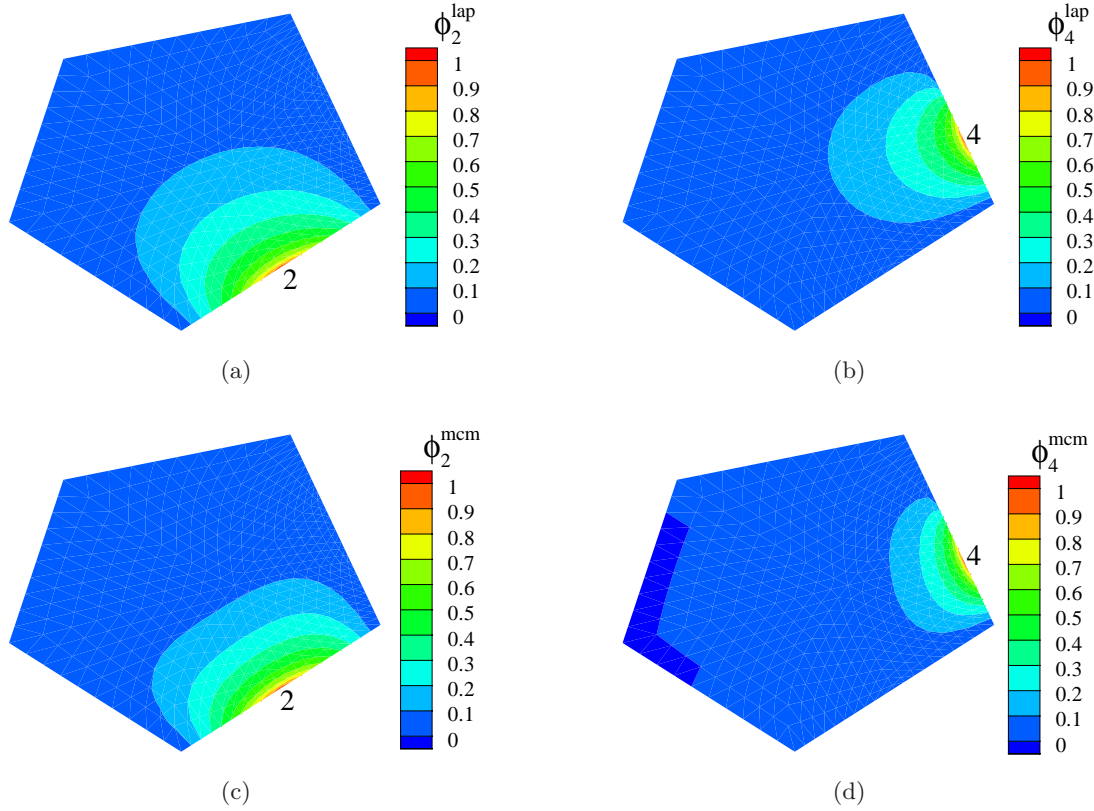
Figure 20. Shape functions for mid-side node and side-node in a weakly convex polygon. (a), (b) Laplace shape functions; (c), (d) Metric coordinate method

The performance of metric coordinate, mean value coordinate, and the Laplace shape functions on quadtree meshes is compared in Table 3. On quadtree meshes, the Laplace interpolant was the most accurate with $\mathcal{O}(10^{-10})$ error in the relative $L^2(\Omega)$ and energy error norms. To investigate the properties of the stiffness matrix (linear elasticity) on quadtree meshes, the five-node element A shown in Figure 7a is considered. The finite-dimensional eigenvalue problem for the stiffness matrix is: $\mathbf{Kd} = \lambda\mathbf{d}$. The material properties used are: $E = 1$, $\nu = 0.3$, and plane strain conditions are assumed. On using the Laplace interpolant as trial and test functions, the solution for the eigenvalues of $\mathbf{K}$ is: $\boldsymbol{\lambda} = \{0, 0, 0, 0.567, 0.575, 0.678, 0.685, 1.227, 1.409, 1.941\}$, whereas with the metric coordinate method, $\boldsymbol{\lambda} = \{0, 0, 0, 0.567, 0.577, 0.679, 0.683, 1.150, 1.587, 2.102\}$ is obtained. The quadrature scheme used was sufficient to yield a well-conditioned system. The three zero-eigenvalues correspond to the physical (rigid-body) modes, and λ_i is positive for $i = 4$–10, which indicates that $\mathbf{K}$ has full rank and the absence of spurious modes.

The numerical integration errors are the cause for the loss in accuracy on the convex and weakly convex (quadtree) meshes. On using the Laplace shape function, the results can be further improved by performing a partitioning of each sub-triangle (m divisions on each edge) of the reference polygon. This results in nm^2 additional triangles for an n-gon. On using $m = 3$, the results for the relative $L^2(\Omega)$ error norm on meshes a–c are $\mathcal{O}(10^{-16})$, $\mathcal{O}(10^{-13})$, and $\mathcal{O}(10^{-12})$, respectively, whereas the relative energy error norms are an order less in each case. The results for the relative $L^2(\Omega)$ and energy error norms on the quadtree meshes d–f are $\mathcal{O}(10^{-16})$ and $\mathcal{O}(10^{-15})$, respectively. In Figure 23a, the integration cells on
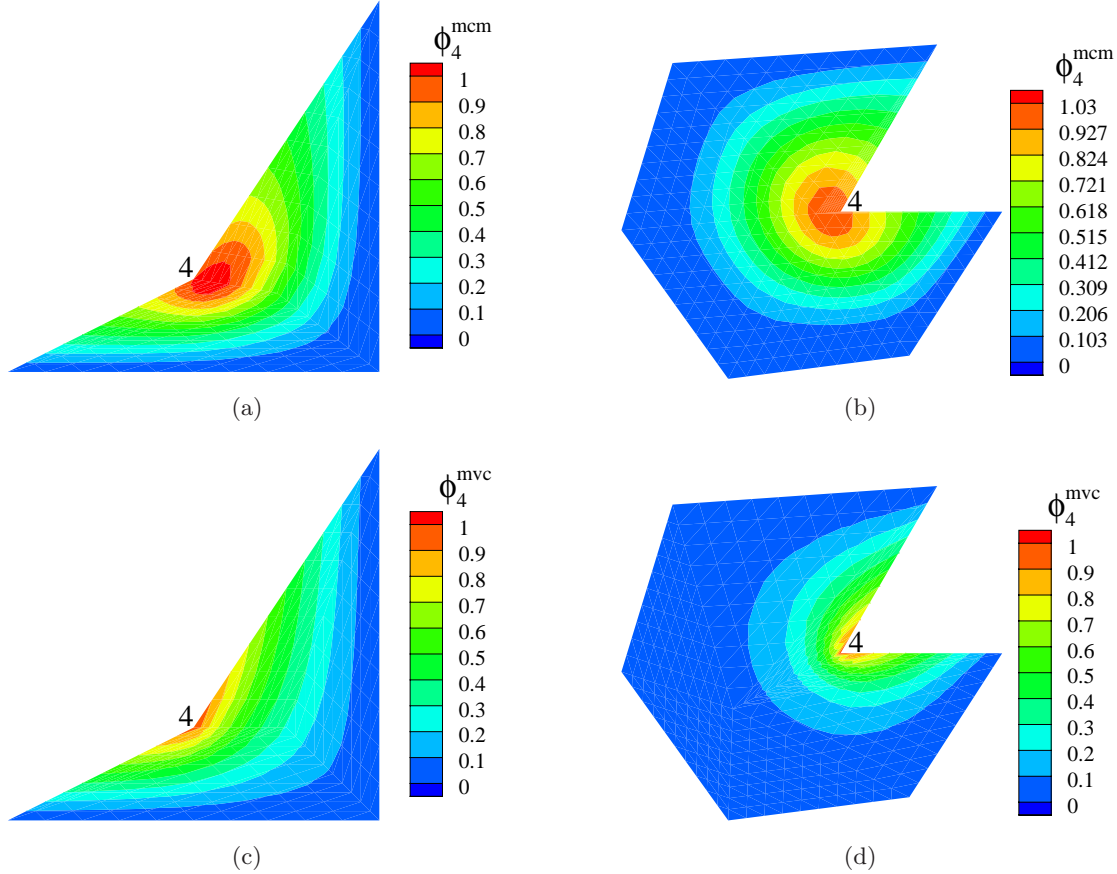
Figure 21. Shape functions in non-convex elements. (a), (b) Metric coordinate
method; (c), (d) Mean value coordinates

a reference pentagon are shown, and the partitioning of the elements in the meshes shown
in Figures 22a and 22d are illustrated in Figures 23b and 23c, respectively.

The relative error norms obtained on meshes with concave elements are presented in
Table 4. The mean value coordinates are used to evaluate the shape functions on convex
and concave elements. The results for the relative $L^2(\Omega)$ and energy error norm on these
meshes are of $\mathcal{O}(10^{-4})$.

5 CONCLUDING REMARKS

In this paper, we have summarized the past developments and recent advances in the con-
struction of polygonal finite elements interpolants. The shape functions were required to
form a partition of unity, satisfy linear precision, be non-negative on convex domains, and
be strictly linear on the boundary of the polygonal domain. A general prescription for
the construction of polygonal interpolants was presented, and where appropriate, links to
known shape functions were provided. The Wachspress, mean-value, and Laplace shape
functions fall within the category of three-point schemes [22]. The Laplace shape function
was shown to be identical to the *discrete harmonic weight*, which was introduced by Pinkall
and Polthier [38]. The construction of maximum entropy shape functions [50] via a con-
strained optimization problem (Jaynes's maximum entropy principle [24]) was distinct from
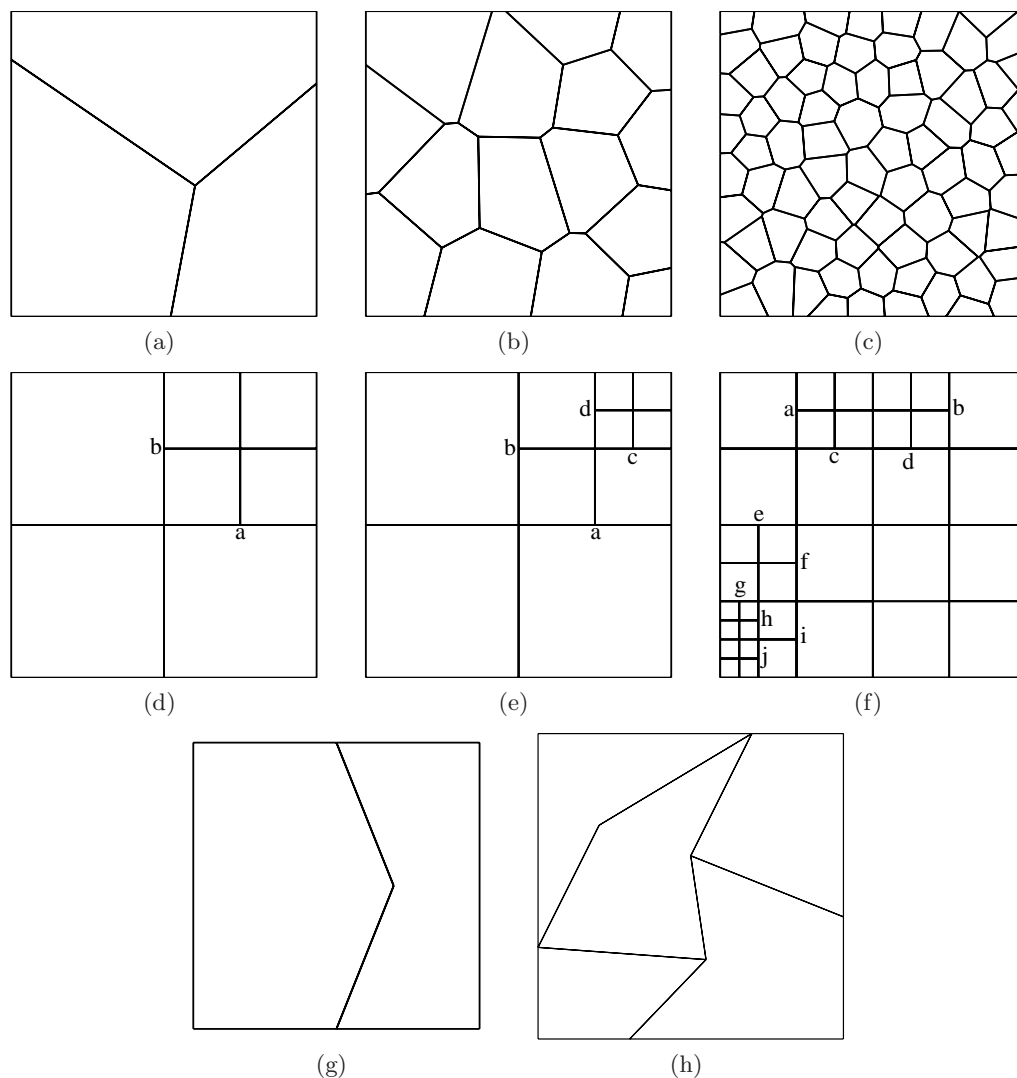
Figure 22. Patch test. (a)–(c) Convex polygonal meshes *a*, *b*, and *c* with 8, 22, and 156 nodes, respectively; (d)–(f) Quadtree meshes *d*, *e*, and *f* with 14, 19, and 52 nodes, respectively; (g),(h) Concave polygonal meshes *g* and *h* with 7 and 11 nodes, respectively

the other polygonal interpolants. The plots of these polygonal shape functions on convex, weakly convex (quadtree meshes), and concave elements were presented.

Numerical studies on the patch test for convex polygonal meshes, quadtree meshes, and meshes with concave elements were performed. Patch test results using the Laplace interpolant were the most accurate, with relative $L^2(\Omega)$ error norm of $\mathcal{O}(10^{-8})$ and $\mathcal{O}(10^{-10})$ on convex polygonal meshes and quadtree meshes, respectively. On meshes with concave elements, the mean value coordinate yielded relative $L^2(\Omega)$ and energy error norms of $\mathcal{O}(10^{-4})$. Mesh generation using polygonal elements and the use of polygonal interpolants in finite element computations holds promise. Further numerical studies and mathematical analysis of these interpolants is required to develop a more complete understanding of their accuracy, robustness, and convergence in the numerical solution of partial differential equations.

Meshes	Wachspress		Mean value coordinate		Laplace	
	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$
a	8.8×10^{-6}	2.4×10^{-5}	3.5×10^{-3}	9.1×10^{-3}	5.4×10^{-11}	1.5×10^{-10}
b	2.0×10^{-4}	1.7×10^{-4}	8.2×10^{-4}	6.3×10^{-3}	4.0×10^{-9}	1.9×10^{-8}
c	1.1×10^{-4}	2.5×10^{-3}	2.6×10^{-4}	4.0×10^{-3}	7.7×10^{-9}	1.8×10^{-7}

Table 2. Relative L^2 and energy error norms for the patch test on convex polygonal meshes

Meshes	Metric coordinate		Mean value coordinate		Laplace	
	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$
d	9.9×10^{-6}	5.3×10^{-5}	2.4×10^{-5}	9.1×10^{-5}	9.9×10^{-11}	3.2×10^{-10}
e	1.0×10^{-5}	6.0×10^{-5}	2.3×10^{-5}	1.1×10^{-4}	8.7×10^{-11}	4.0×10^{-10}
f	4.0×10^{-6}	5.1×10^{-5}	1.2×10^{-5}	9.3×10^{-5}	6.3×10^{-11}	3.9×10^{-10}

Table 3. Relative L^2 and energy error norms for the patch test on quadtree meshes

(a)   (b)   (c)
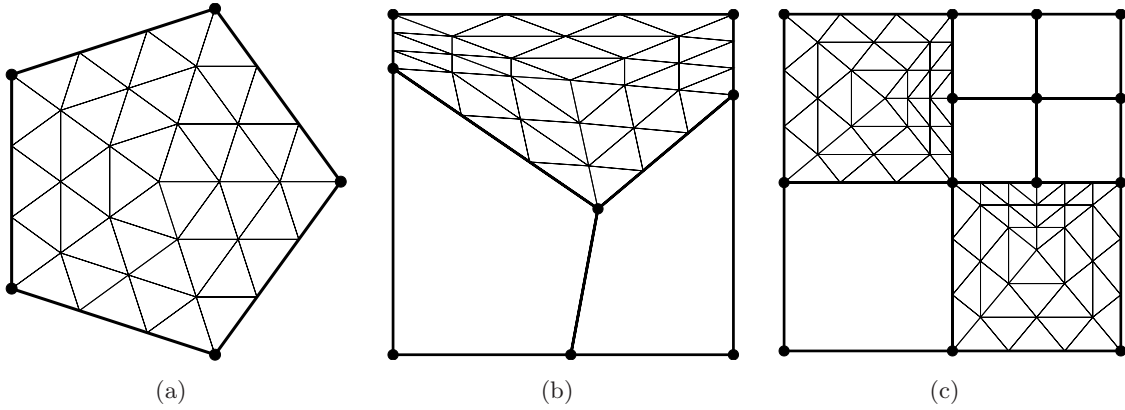
Figure 23. Improved numerical integration scheme. (a) Partition of the reference pentagon; (b) and (c) Mapped integration cells for polygonal and quadtree meshes

Meshes	Metric coordinate method	
	$\dfrac{\|u-u^h\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}$	$\dfrac{\|u-u^h\|_{E(\Omega)}}{\|u\|_{E(\Omega)}}$
g	2.2×10^{-4}	6.5×10^{-4}
h	1.5×10^{-4}	6.2×10^{-4}

Table 4. Relative L^2 and energy error norms for the patch test on meshes with concave elements

ACKNOWLEDGEMENTS

REFERENCES

1 N. Agmon, Y. Alhassid and R. D. Levine (1978). An algorithm for determining the Lagrange parameters in the maximal entropy formalism. In M. Tribus and R. D. Levine (Eds.), *The Maximum Entropy Formalism*, pp. 206–209, Cambridge, MA. MIT Press.

2 N. Agmon, Y. Alhassid and R. D. Levine (1979). An algorithm for finding the distribution of maximal entropy. *Journal of Computational Physics*, **30**, 250–258.

3 M. Arroyo and M. Ortiz (2005). Local *maximum-entropy* approximation schemes: A seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering*, in press.

4 V.V. Belikov, V.D. Ivanov, V.K. Kontorovich, S.A. Korytnik and A. Yu. Semenov (1997). The non-Sibsonian interpolation: A new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics*, **37**(1), 9–15.

5 J.-P. Berrut and L.N. Trefethen (2004). Barycentric Lagrange interpolation. *SIAM Review*, **46**(3), 501–517.

6 M. D. Buhmann (2000). Radial basis functions. *Acta Numerica*, **9**, 1–38.

7 N. H. Christ, R. Friedberg and T. D. Lee (1982). Weights of links and plaquettes in a random lattice. *Nuclear Physics B*, **210**(3), 337–346.

8 H.S.M. Coxeter (1961). *Introduction to Geometry*. John Wiley and Sons, New York, N.Y.

9 E. Cueto, N. Sukumar, B. Calvo, M.A. Martínez, J. Cegoñiino and M. Doblaré (2003). Overview and recent advances in natural neighbour Galerkin methods. *Archives of Computational Methods in Engineering*, **10**(4), 307–384.

10 W. Dahmen, H.P. Dikshit and A. Ojha (2000). On Wachspress quadrilateral patches. *Computer Aided Geometric Design*, **17**, 879–890.

11 G. Dasgupta (2003). Integration within polygonal finite elements. *Journal of Aerospace Engineering*, **16**(1), 9–18.

12 G. Dasgupta (2003). Interpolants within convex polygons: Wachspress' shape functions. *Journal of Aerospace Engineering*, **16**(1), 1–8.

13 A.R. Diaz and A. Bénard (2003). Designing materials with prescribed elastic properties using polygonal cells. *International Journal for Numerical Methods in Engineering*, **57**(3), 301–314.

14 C.R. Dohrmann, S. W. Key and M.W. Heinstein (2000). A method for connecting dissimilar finite element meshes in two dimensions. *International Journal for Numerical Methods in Engineering*, **48**, 655–678.

15 M.S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

16 M.S. Floater and K. Hormann (2005). Surface parameterization: a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin (Eds.), *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pp. 157–186. Springer, Berlin, Heidelberg.

17 M.S. Floater, K. Hormann, and G. Kós (2004). A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*. in press.

18 S. Ghosh and S. Moorthy (1995). Elastic-plastic analysis of arbitrary heterogeneous materials with the Voronoi cell finite-element method. *Computer Methods in Applied Mechanics and Engineering*, **121**(1–4), 373–409.

19 J.L. Gout (1985). Rational Wachspress-type finite elements on regular hexagons. *IMA Journal of Numerical Analysis*, **5**(1), 59–77.

20 B. Grünbaum (1967). *Convex polytopes.* John Wiley and Sons, New York.

21 H. Hiyoshi and K. Sugihara (1999). Two generalizations of an interpolant based on Voronoi diagrams. *International Journal of Shape Modeling*, **5**(2), 219–231.

22 K. Hormann (2004). Barycentric coordinates for arbitrary polygons in the plane. Technical Report, Clausthal University of Technology, September.

23 T.J.R. Hughes (1987). *The Finite Element Method.* Prentice-Hall, Englewood Cliffs, N.J.

24 E.T. Jaynes (1957). Information theory and statistical mechanics. *Physical Review*, **106**(4), 620–630.

25 E.T. Jaynes (1989). *Concentration of Distributions at Entropy Maxima*, pp. 317–336. In Rosenkrantz [41].

26 E.T. Jaynes (2003). *Probability Theory: The Logic of Science.* Cambridge University Press, Cambridge, UK, 1st. Edition.

27 J.N. Kapur (1993). *Maximum-Entropy Models in Science and Engineering.* John Wiley & Sons, Inc., New Delhi, India, 1st. (revised) Edition.

28 A. Khinchin (1957). *Mathematical Foundations of Information Theory.* Dover, New York, N.Y.

29 E.A. Malsch (2003). *Test functions for elliptic operators satisfying essential edge conditions on both convex and concave polygonal domains.* PhD thesis, Columbia University.

30 E. A. Malsch and G. Dasgupta (2001). Shape functions for concave quadrilaterals. In K. J. Bathe, editor, *Proceedings of the first MIT Conference on Fluid and Solid Mechanics*, Volume 2, pp. 1617–1622, Amsterdam, The Netherlands, Elsevier Press.

31 E.A. Malsch and G. Dasgupta (2004). Interpolation constraints and thermal distributions: a method for all non-concave polygons. *International Journal of Solids and Structures*, **41**(8), 2165–2188.

32 E.A. Malsch and G. Dasgupta (2004). Shape functions for polygonal domains with interior nodes. *International Journal for Numerical Methods in Engineering*, **61**(12), 1153–1172.

33 E.A. Malsch and G. Dasgupta (2005). Algebraic construction of smooth interpolants on polygonal domains. *Mathematica Journal*, **9**(3).

34 E.A. Malsch, J.J. Lin, and G. Dasgupta (2005). Smooth two dimensional interpolants: a recipe for all polygons. *Journal of Graphics Tools*, **10**(2).

35 M. Meyer, H. Lee, A. Barr and M. Desbrun (2002). Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, **7**(1), 13–22.

36 J. Nocedal and S.J. Wright (1999). *Numerical Optimization.* Springer-Verlag, New York.27

37 A. Okabe, B. Boots and K. Sugihara (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, Chichester, England.

38 U. Pinkall and K. Polthier (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, **2**(1), 15–36.

39 W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling (1992). *Numerical Recipes in Fortran: The Art of Scientific Computing.* Cambridge University Press, New York, NY, 2nd. Edition.

40 A. Rényi (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability vol. 1*, pages 547–561, Berkeley, CA. University of California Press.

41 R. D. Rosenkrantz (Ed) (1989). *E.T. Jaynes: Paper on Probability, Statistics and Statistical Physics*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

42 V.L. Rvachev, T.I. Sheiko, V. Shapiro and I. Tsukanov (2000). On completeness of RFM solution structures. *Computational Mechanics*, **25**(2–3), 305–316.

43 H. Samet (1984). The quadtree and related hierarchical data structure. *ACM Computing Surveys*, **16**(2), 187–260.

44 C.E. Shannon (1948). A mathematical theory of communication. *The Bell Systems Technical Journal*, **27**, 379–423.

45 D. Shepard (1968). A two-dimensional interpolation function for irregularly spaced points. In *ACM National Conference*, pp. 517–524.

46 R. Sibson (1980). A vector identity for the Dirichlet tesselation. *Mathematical Proceedings of the Cambridge Philosophical Society*, **87**, 151–155.

47 A. K. Soh, L. Zhifei and C. Song (2000). Development of a new quadrilateral thin plate element using area coordinates. *Computer Methods in Applied Mechanics and Engineering*, **190**(8–10), 979–987.

48 G. Strang and G. Fix (1973). *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J.

49 N. Sukumar (2003). Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids. *International Journal for Numerical Methods in Engineering*, **57**(1), 1–34.

50 N. Sukumar (2004). Construction of polygonal interpolants: A maximum entropy approach. *International Journal for Numerical Methods in Engineering*, **61**(12), 2159–2181.

51 N. Sukumar, B. Moran, A. Yu. Semenov and V.V. Belikov (2001). Natural neighbor Galerkin methods. *International Journal for Numerical Methods in Engineering*, **50**(1), 1–27.

52 N. Sukumar and A. Tabarraei (2004). Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering*, **61**(12), 2045–2066.

53 A. Tabarraei and N. Sukumar (2005). Adaptive computations on conforming quadtree meshes. *Finite Elements in Analysis and Design*, **41**(7–8), 686–702.

54 C. Tsallis (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**(1–2), 479–487.

55 E.L. Wachspress (1975). *A Rational Finite Element Basis*. Academic Press, New York, N.Y.

56 S. Wandzura and H. Xiao (2003). Symmetric quadrature rules on a triangle. *Computers and Mathematics with Applications*, **45**, 1829–1840.

57 J. Warren (1996). Barycentric coordinates for convex polytopes. Technical report, Department of Computer Science, Rice University.

58 J. Warren (1996). Barycentric coordinates for convex polytopes. *Advances in Computational Mathematics*, **6**(1), 97–108.

59 J. Warren (2003). On the uniqueness of barycentric coordinates. In *Contemporary Mathematics, Proceedings of AGGM02*, pp. 93–99.

60 J. Warren, S. Schaefer, A.N. Hirani and M. Desbrun (2004). Barycentric coordinates for convex sets. preprint.