

POLYGONAL SPLINE SPACES AND THE NUMERICAL SOLUTION OF THE POISSON EQUATION*

MICHAEL S. FLOATER[†] AND MING-JUN LAI[‡]

Abstract. It is known that generalized barycentric coordinates can be used to form Bernstein polynomial-like functions over a polygon with any number of sides. We propose to use these functions to form a space of continuous polygonal splines (piecewise defined functions) of order d over a partition consisting of polygons which is able to reproduce all polynomials of degree d . Locally supported basis functions for the space are constructed for order $d \geq 2$. The construction for $d = 2$ is simpler than the “serendipity” quadratic finite elements that have appeared in the recent literature. The number of basis functions is similar to, but fewer than, those of the virtual element method. We use them for the numerical solution of the Poisson equation on two special types of nontriangular partitions to present a proof of concept for solving PDEs over polygonal partitions. Numerical solutions based on quadrangulations and pentagonal partitions are demonstrated to show the efficiency of these polygonal spline functions. They can lead to a more accurate solution by using fewer degrees of freedom than the traditional continuous polynomial finite element method if the solutions are smooth although assembling the mass and stiffness matrices can take more time.

Key words. polygonal splines, generalized barycentric coordinates, numerical solution of PDEs

AMS subject classifications. 65M60, 65D07

DOI. 10.1137/15M101155X

1. Introduction. In this paper, we are interested in constructing spline functions (piecewise defined functions) over an arbitrary polygonal partition Δ of a domain $\Omega \subset \mathbb{R}^2$ and present an application. It is known that one can define bivariate spline functions over a triangulation \triangle of Ω consisting of triangles and use them for the numerical solution of partial differential equations (PDEs), scattered data fitting, etc.; see, e.g., [2], [14], [15], and [17]. If a triangulation \triangle is replaced by a collection Δ of polygons, say convex quadrilaterals or pentagons or even a mix of arbitrary polygons such as a Voronoi diagram of a bounded domain, can we extend the bivariate splines to Δ ? The answer is yes. The method of virtual elements of [6] is one approach, based on solutions to the Laplace equation in each polygon; see also [5] and [19]. The construction of serendipity quadratic finite elements using generalized barycentric coordinates (GBCs) in [22], [23] provides another approach. Motivated by these studies we use GBCs to define polygonal splines of order $d \geq 2$ on a partition Δ of convex polygons. Some advantages of these splines are the following:

- (1) we can partition a polygonal domain Ω of interest using various n -sided polygons including triangles, even though triangles are very convenient for partitioning Ω ;
- (2) as there are various GBCs which include polynomial, rational, trigonometric, logarithmic functions, etc. (see a summary of various GBCs in a recent survey article [10]), we are allowed to use more general functions than

*Received by the editors March 9, 2015; accepted for publication (in revised form) December 7, 2015; published electronically March 24, 2016.

<http://www.siam.org/journals/sinum/54-2/M101155.html>

[†]Department of Mathematics, University of Oslo, Blindern, 0316 Oslo, Norway (michaelf@math.uio.no).

[‡]Department of Mathematics, University of Georgia, Athens, GA 30602 (mjlai@uga.edu). This author’s research was partially supported by the Simons Foundation under a collaboration grant 280646 and the National Science Foundation under grant DMS-1521537.

polynomials to approximate known and unknown functions, e.g., solutions of PDEs;

- (3) there are redundant functions over each polygon when the side number $n > 3$, which may be useful for certain applications; and
- (4) our numerical solutions of the Poisson equation indicate that they have some advantages over the standard finite element method. See details in section 5.4.

The potential advantages of this method come, however, at a price because

- (i) the number of GBC-based functions on each polygon grows rapidly as the number of sides of the polygon increases. For example, the number of quadratic functions generated by GBCs in an n -gon is $n(n+1)/2$, which grows quadratically with n ;
- (ii) the functions are not in general linearly independent, making it difficult to choose suitable degrees of freedom; and
- (iii) the evaluation of functions based on GBCs is not as efficient as the evaluation of polynomials. In particular, it is not as efficient as the finite element method for assembling the mass and stiffness matrices associated with weak solutions of Poisson equations.

The polygonal splines can be attractive for computation when the overhead for generating the mass and stiffness matrices is not a major component of the computation, e.g., solving a time-dependent diffusion equation over a spatial polygon $\Omega \subset \mathbb{R}^2$ (IBVP) or when solving a nonlinear Poisson equation, e.g., $-\Delta u + f(u) = 0$, where $f(u)$ is a nonlinear function of u .

One way to curtail the redundancy of the polygonal GBC-based functions is to use a reduced space of functions which still forms a continuous polygonal spline when pieced together, and still reproduces all polynomials of degree $\leq d$. This is what we do in this paper. Let us first emphasize that this is not a brand new idea and such reduced elements were constructed in the quadratic case in [23], based on the observation that many of the quadratic functions are zero on the boundary of the polygon, and thus do not contribute to interelement continuity. By forming various linear combinations of the quadratic functions they obtained “reduced” quadratic elements, having only $2n$ degrees of freedom, but retaining interelement continuity. Their construction includes the usual 8-node serendipity element on squares and parallelograms as a special case [24] and [1]. However, their construction is not trivial. One of the purposes of this paper is to offer a simpler way of constructing quadratic elements of this kind. Our main idea is to use not only GBCs but also the *local* barycentric coordinates defined by triples of consecutive vertices of the polygon. Similarly to [23], this generates $2n$ functions in the quadratic case. From these functions we also recover the quadratic rational interpolatory element constructed by Wachspress [25], but in explicit form.

To obtain polygonal functions of higher order we limit ourselves to the rational GBCs of Wachspress and use additional degrees of freedom in the interior of the polygon. The dimension of the space of order d over an n -gon is then

$$(1) \quad dn + (d-1)(d-2)/2.$$

The space is able to reproduce all polynomials of degree d . This dimension is smaller than that of the virtual element method of [6] to be explained in section 4.1, Remark 1. Once we have these functions over each polygon, we can piece them together to make a polygonal spline space $S_d(\Delta)$ over a collection Δ of mixed polygons P_n with various $n \geq 3$. Hence, the dimension of the polygonal spline space of degree d over a mixed

partition Δ can be established as in Theorem 4 when $d = 2$ for any GBCs and when $d \geq 3$ for Wachspress coordinates.

When P_n is a triangle ($n = 3$), the number in (1) is $(d+2)(d+1)/2$, which is the number of Bernstein polynomials of degree d in P_n . When Δ is a triangulation, the dimension in (30) in Theorem 4 agrees with the dimension of the continuous spline space of degree d explained in [14].

In addition to the construction of a locally supported basis, we study the standard interpolation problem using these polygonal spline functions. As one might expect, we can use the space $S_d(\Delta)$ to do interpolation which will be useful to approximate the functions on the right-hand side of the Poisson equation for numerical solution.

Finally, we shall describe how to implement these functions for the numerical solution of the Poisson equation in the last section. Our numerical results demonstrate the efficiency of this approach. In addition, we introduce a novel refinement scheme for pentagons and show that the root mean square error (RMSE) of polygonal spline solutions over uniformly refined pentagonal partitions decreases faster than that of the standard finite element method over uniformly refined triangulations; see the tables of numerical results in section 5.4.

The major contributions of this paper are as follows:

- a. We provide a simpler construction of serendipity quadratic finite elements than the one in [23] and a construction of higher order finite elements using Wachspress coordinates.
- b. The number of basis functions per polygon is smaller than that of the virtual elements of [6] for order $d \geq 2$.
- c. We have used our elements to solve the Poisson equation numerically, which extends the study in [19] where polygonal finite elements of order $d = 1$ were used.
- d. We introduce a refinement scheme for pentagons which is new to the best of our knowledge.

However, the generation of the mass and stiffness matrices for these polygonal splines takes much more time than for polynomial finite elements. More research needs to be done to overcome this difficulty if these splines are to be useful in practice.

2. Background on GBCs. There are many ways to define barycentric coordinates in a polygon with n sides, $n \geq 3$; see [10]. We will consider only convex polygons in this paper. Let $P_n = \langle \mathbf{v}_1, \dots, \mathbf{v}_n \rangle$ be a convex polygon. Any functions $\phi_i : P_n \rightarrow \mathbb{R}$, $i = 1, \dots, n$, will be called GBCs if, for all $\mathbf{x} \in P_n$, $\phi_i(\mathbf{x}) \geq 0$,

$$(2) \quad \sum_{i=1}^n \phi_i(\mathbf{x}) = 1, \quad \text{and} \quad \sum_{i=1}^n \phi_i(\mathbf{x}) \mathbf{v}_i = \mathbf{x}.$$

When P_n is a triangle, the coordinates ϕ_1, ϕ_2, ϕ_3 are the usual barycentric coordinates. For $n > 3$, the ϕ_i are not uniquely determined by (2) but they share the basic property that they are piecewise linear on the boundary of P_n : $\phi_i(\mathbf{v}_j) = \delta_{ij}$ and

$$(3) \quad \phi_i((1-\mu)\mathbf{v}_j + \mu\mathbf{v}_{j+1}) = (1-\mu)\phi_i(\mathbf{v}_j) + \mu\phi_i(\mathbf{v}_{j+1}), \quad \mu \in [0, 1].$$

Here, and throughout the paper, we use the cyclic indexing, $\phi_{n+1} := \phi_1$, $\phi_0 := \phi_n$, and so on.

For convenience, let us describe three well-known GBCs, which are all smooth in the interior of P_n .

Example 1 (inverse bilinear coordinates). For $n = 4$, the quadrilateral P_n is the image of a bilinear map from the unit square $[0, 1]^2$, and for each $\mathbf{x} \in P_n$, there exists a unique pair $(\alpha, \beta) \in [0, 1]^2$ such that

$$\mathbf{x} = (1 - \alpha)(1 - \beta)\mathbf{v}_1 + \alpha(1 - \beta)\mathbf{v}_2 + \alpha\beta\mathbf{v}_3 + (1 - \alpha)\beta\mathbf{v}_4.$$

Thus the four functions

$$(4) \quad \phi_1(\mathbf{x}) = (1 - \alpha)(1 - \beta), \quad \phi_2(\mathbf{x}) = \alpha(1 - \beta), \quad \phi_3(\mathbf{x}) = \alpha\beta, \quad \phi_4(\mathbf{x}) = (1 - \alpha)\beta$$

are GBCs. An explicit formula is given in [10].

Example 2 (Wachspress (rational) coordinates). For general $n \geq 3$, let $\mathbf{n}_i \in \mathbb{R}^2$ be the outward unit normal to the edge $e_i = [\mathbf{v}_i, \mathbf{v}_{i+1}]$, $i = 1, \dots, n$, and for any $\mathbf{x} \in P_n$ let $h_i(\mathbf{x})$ be the perpendicular distance of \mathbf{x} to the edge e_i , so that

$$(5) \quad h_i(\mathbf{x}) = (\mathbf{v}_i - \mathbf{x}) \cdot \mathbf{n}_i = (\mathbf{v}_{i+1} - \mathbf{x}) \cdot \mathbf{n}_i.$$

Let

$$(6) \quad w_i(\mathbf{x}) = d_i \prod_{\substack{j=1, \dots, n \\ j \neq i-1, i}} h_j(\mathbf{x}),$$

where d_i is the cross product

$$d_i = \mathbf{n}_{i-1} \times \mathbf{n}_i = \begin{vmatrix} n_{i-1}^x & n_i^x \\ n_{i-1}^y & n_i^y \end{vmatrix}$$

and $\mathbf{n}_j = (n_j^x, n_j^y)$. Then, with

$$(7) \quad W = \sum_{j=1}^n w_j,$$

the functions $\phi_i = w_i/W$, $i = 1, \dots, n$, are GBCs, which are rational of degree $(n - 2, n - 3)$. See [10] for several other representations of these coordinates.

Example 3 (mean value coordinates). For general $n \geq 3$, and for \mathbf{x} in the interior of P_n , let α_i be the angle at \mathbf{x} of the triangle $\langle \mathbf{v}_i, \mathbf{x}, \mathbf{v}_{i+1} \rangle$, $i = 1, \dots, n$, and define (cf. [9])

$$w_i(\mathbf{x}) = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\|\mathbf{x} - \mathbf{v}_i\|} \quad \text{and} \quad W = \sum_{i=1}^n w_i,$$

with $\|\cdot\|$ the Euclidean norm in \mathbb{R}^2 . Then the functions $\phi_i = w_i/W$, which extend continuously to the boundary of P_n , are GBCs. See [10] for an explanation of this and how these coordinates extend (though not necessarily positively) to nonconvex polygons.

Various other smooth GBCs have been constructed such as Gordon–Wixom and maximum entropy coordinates. We note also that both Wachspress and mean value coordinates have been generalized to three dimensions (polyhedra); see [10] for a summary.

3. Polygonal Bernstein–Bézier functions. We now study generalizations of Bernstein–Bézier polynomials to a convex polygon, dealing first with the “full” space of functions that has been studied in the literature, and then construct a reduced space.

3.1. The full space. It is well known that the standard barycentric coordinates on a triangle in \mathbb{R}^2 can be used to generate Bernstein–Bézier polynomials of degree d via the multinomial theorem (cf. [8]). Similarly, on a convex polygon in \mathbb{R}^2 one can use GBCs to generate analogous functions, whose span will contain polynomials of degree $\leq d$. Such functions were studied by Loop and DeRose [18] with a view to constructing multisided surface patches, “S-patches,” for modeling geometry.

For a convex polygon P_n with $n \geq 3$ sides, let ϕ_1, \dots, ϕ_n be a set of GBCs. For any $d \geq 0$, and any multi-index $\mathbf{j} = (j_1, \dots, j_n) \in \mathbb{N}_0^n$ with $|\mathbf{j}| := j_1 + \dots + j_n = d$, let

$$(8) \quad B_{\mathbf{j}}^d(\mathbf{x}) = \frac{d!}{j_1! \cdots j_n!} \phi_1^{j_1}(\mathbf{x}) \cdots \phi_n^{j_n}(\mathbf{x}), \quad \mathbf{x} \in P_n,$$

which we will call a Bernstein–Bézier function.

In the triangular case, $n = 3$, $B_{\mathbf{j}}^d$ is a Bernstein–Bézier polynomial of degree d (cf. [14]). For $n > 3$, $B_{\mathbf{j}}^d$ is no longer a polynomial in general and its function type will depend on the ϕ_i . Nevertheless, for any n , the linear space $\Phi_d(P_n)$ of functions of the form

$$(9) \quad s(\mathbf{x}) = \sum_{|\mathbf{j}|=d} c_{\mathbf{j}} B_{\mathbf{j}}^d(\mathbf{x}), \quad \mathbf{x} \in P_n,$$

with $c_{\mathbf{j}} \in \mathbb{R}$, inherits two important properties from the ϕ_i .

(P1) $\Pi_d \subset \Phi_d(P_n)$, where Π_d is the space of polynomials of degree $\leq d$. This is because, due to (2), any linear polynomial can be expressed as a linear combination of the ϕ_i . Then based on (8), polynomials of degree d can be generated.

(P2) Due to (3), the function $s(\mathbf{x})$ in (9) is a univariate polynomial of degree $\leq d$ on each edge of the polygon.

In general, the functions $B_{\mathbf{j}}^d$, $|\mathbf{j}| = d$, are not linearly independent, and, as we will see, the dimension of $\Phi_d(P_n)$ is less than the number of these polygonal Bernstein–Bézier functions. Indeed, let us consider the case $d = 2$. Among the pairwise products $\phi_i \phi_j$, $i, j = 1, \dots, n$, there are n products of the form ϕ_i^2 , $i = 1, \dots, n$ and $\binom{n}{2}$ of the form $\phi_i \phi_j$, $i \neq j$. So the total number of $B_{\mathbf{j}}^2$, $|\mathbf{j}| = 2$, is

$$(10) \quad D_n = n + \binom{n}{2} = \binom{n+1}{2}.$$

So,

$$D_3 = 6, \quad D_4 = 10, \quad D_5 = 15, \quad D_6 = 21,$$

and so on. However, the dimension of $\Phi_2(P_n)$ is lower than D_n when $n \geq 4$. We propose an open problem: What is the dimension of $\Phi_d(P_n)$ for various GBCs and $d \geq 2$ and $n > 3$?

For $d = 2$ and for Wachspress coordinates, we can show the following:

THEOREM 1.

$$(11) \quad \dim(\Phi_2(P_n)) = 2n + \frac{(n-2)(n-3)}{2}.$$

Proof. The products ϕ_i^2 , $\phi_i \phi_{i+1}$, $i = 1, \dots, n$, are linearly independent due to their values at \mathbf{v}_i , $(\mathbf{v}_i + \mathbf{v}_{i+1})/2$, $i = 1, \dots, n$. Therefore we can express $\Phi_2(P_n)$ as

$$\Phi_2(P_n) = B_n \oplus I_n,$$

where

$$B_n = \text{span}\{\phi_i^2, \phi_i\phi_{i+1}, i = 1, \dots, n\} \quad \text{and} \quad I_n = \text{span}\{\phi_i\phi_j : |i - j| \geq 2\}.$$

The sum is direct because all the functions in I_n are zero on the entire boundary ∂P_n while the only function in B_n with this property is 0.

Since $\dim(B_n) = 2n$, it remains to determine the dimension of I_n . Recalling h_j in (6), let b be the bubble function

$$(12) \quad b(\mathbf{x}) = \prod_{k=1}^n h_k(\mathbf{x}).$$

When $|i - j| \geq 2$, the four indices $i - 1, i, j - 1, j$ are distinct and so from (6),

$$\phi_i\phi_j = \frac{w_i w_j}{W^2} = \frac{bd_i d_j \nu_{ij}}{W^2},$$

where

$$(13) \quad \nu_{ij} = \prod_{k \neq i-1, i, j, j-1} h_k.$$

Therefore, $\dim(I_n) = \dim(H_n)$, where

$$H_n = \text{span}\{\nu_{ij} : |i - j| \geq 2\}.$$

Observe that $\nu_{ij} \in \Pi_{n-4}$ since h_k is a linear polynomial and so $H_n \subset \Pi_{n-4}$. In fact we will show that

$$(14) \quad H_n = \Pi_{n-4}$$

from which we deduce that $\dim(H_n) = (n - 2)(n - 3)/2$ which yields (11). We now present a proof of (14). For $n = 4$ it is easy as (13) is an empty product, i.e., $\nu_{13} = \nu_{24} = 1$. For $n \geq 5$, notice that each edge of the polygon can be parallel with at most one other edge. Therefore, any three lines $h_k = 0$ intersect in at least two points from which it follows that any three h_k are linearly independent, and thus form a basis for Π_1 . In the case $n = 5$, the ν_{ij} are linear polynomials, which are $\nu_{i, i+2} = h_{i+3}$, $i = 1, \dots, 5$. Since any three of them span Π_1 (14) follows.

Finally, for $n \geq 6$, consider the monomial x^{n-4} , where $\mathbf{x} = (x, y)$. For any distinct r, s, t in $\{1, \dots, n\}$, we can write it as

$$x^{n-4} = xx^{n-5} = (c_r h_r + c_s h_s + c_t h_t)x^{n-5} = c_r h_r x^{n-5} + c_s h_s x^{n-5} + c_t h_t x^{n-5}$$

for coefficients $c_r, c_s, c_t \in \mathbb{R}$, and a similar decomposition can be applied to any other monomial of degree $n - 4$, and so if, for any $r = 1, \dots, n$, we can express x^{n-5} as a linear combination of the polynomials

$$\mu_{i,j} = \prod_{k \neq r, i-1, i, j-1, j} h_k,$$

where $r, i - 1, i, j - 1, j$ are distinct, (14) will follow. Thus it is sufficient to show that the polynomials μ_{ij} span Π_{n-5} , and we may assume that r is fixed and $r = n$. The possible indices of the μ_{ij} with $i < j$, are $i = 2, \dots, n - 3$ and $j = i + 2, \dots, n - 1$, and

thus the number of μ_{ij} is $(n - 4)(n - 3)/2$, which is precisely the dimension of Π_{n-5} . So, showing that the μ_{ij} span Π_{n-5} is equivalent to showing that they are linearly independent.

We thus complete the proof by showing that the μ_{ij} are linearly independent. We prove this by induction on n , $n \geq 5$. It is true for $n = 5$ since there is only one μ_{ij} , namely, $\mu_{2,4}$. For $n > 5$ suppose that

$$(15) \quad \sum_{i=2}^{n-3} \sum_{j=i+2}^{n-1} c_{ij} \mu_{i,j}(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathbb{R}^2.$$

Let l_k denote the line $h_k = 0$ for each $k = 1, \dots, n - 1$. Suppose first that l_1 and l_{n-1} are not parallel, and let $\mathbf{x} = l_1 \cap l_{n-1}$. Then all the μ_{ij} are zero at \mathbf{x} except $\mu_{2,n-1}$, and so $c_{2,n-1} = 0$ in (15). Otherwise, l_1 and l_{n-1} are parallel, in which case let $\mathbf{x}_0 = (\mathbf{v}_1 + \mathbf{v}_n)/2$ and let \mathbf{d} be the unit vector parallel to l_1 and l_{n-1} in the direction away from the polygon, and consider the point $\mathbf{x} = \mathbf{x}_0 + \lambda \mathbf{d}$ as $\lambda \rightarrow \infty$. From (5), $h_1(\mathbf{x})$ and $h_{n-1}(\mathbf{x})$ are constant in λ , while

$$h_k(\mathbf{x}) \sim -\lambda \mathbf{d} \cdot \mathbf{n}_k, \quad k = 2, \dots, n - 2.$$

Therefore, since

$$\frac{\mu_{i,j}}{\mu_{2,n-1}} = \frac{h_1 h_2 h_{n-2} h_{n-1}}{h_{i-1} h_i h_{j-1} h_j},$$

we see that if $(i, j) \neq (2, n - 1)$,

$$\frac{\mu_{i,j}(\mathbf{x})}{\mu_{2,n-1}(\mathbf{x})} \rightarrow 0 \quad \text{as } \lambda \rightarrow \infty,$$

and dividing (15) by $\mu_{2,n-1}(\mathbf{x})$ and letting $\lambda \rightarrow \infty$ shows again that $c_{2,n-1} = 0$.

Next, let $\mathbf{x} = l_2 \cap l_{n-1}$ if l_2 and l_{n-1} are not parallel. Then all the μ_{ij} are zero at \mathbf{x} except $\mu_{2,n-1}$ and $\mu_{3,n-1}$. Since $c_{2,n-1} = 0$ this means that $c_{3,n-1} = 0$ as well. The previous argument for parallel lines also applies to show that $c_{3,n-1} = 0$ also if l_2 and l_{n-1} are parallel. We now continue in this way with the pairs of lines l_i and l_{n-1} for $i = 3, \dots, n - 4$, to show sequentially that $c_{i,n-1} = 0$ for $i = 4, \dots, n - 3$. This reduces (15) to

$$(16) \quad h_{n-1}(\mathbf{x}) \sum_{i=2}^{n-4} \sum_{j=i+2}^{n-2} c_{ij} \hat{\mu}_{i,j}(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathbb{R}^2,$$

where the $\hat{\mu}_{ij}$ are the analogs of μ_{ij} for the convex polygon with vertices $\mathbf{v}_1, \dots, \mathbf{v}_{n-1}$. By the induction hypothesis, the $\hat{\mu}_{ij}$ are linearly independent, and therefore the c_{ij} in the sum are zero. \square

Thus at least for Wachspress coordinates, the dimension of $\Phi_2(P_n)$, while lower than the number of functions $B_{\mathbf{j}}^2$, $|\mathbf{j}| = 2$, nevertheless grows quadratically with n . In fact, in this case, the redundancy of the $B_{\mathbf{j}}^2$, $|\mathbf{j}| = 2$, is

$$D_n - \dim(\Phi_2(P_n)) = n - 3.$$

We still do not know the dimension of $\Phi_2(P_n)$ for other GBCs.

3.2. A reduced space. Since the dimension of the whole space $\Phi_d(P_n)$ is high for $d \geq 2$, we would like to construct a subspace $\Psi_d(P_n)$ that satisfies the same two properties (P1) and (P2): a “reduced” space. Our idea for this is to expand any polynomial of degree $\leq d$ from its polar form. Recall that any $p \in \Pi_d$ has a d -variate polar form (or blossom) $\mathcal{P}(\mathbf{x}_1, \dots, \mathbf{x}_d)$ for points $\mathbf{x}_1, \dots, \mathbf{x}_d \in \mathbb{R}^2$ (see [21], [20]), which is uniquely defined by the three properties: (i) it is symmetric in the variables $\mathbf{x}_1, \dots, \mathbf{x}_d$; (ii) it is multiaffine, i.e., affine in each variable while the others are fixed; and (iii) it has the diagonal property

$$(17) \quad \mathcal{P}(\mathbf{x}^{[d]}) := \mathcal{P}(\underbrace{\mathbf{x}, \dots, \mathbf{x}}_d) = p(\mathbf{x}).$$

Given a polynomial $p \in \Pi_d$, we use (17) and expand one of the \mathbf{x} variables using the GBCs ϕ_i and the remaining $d - 1$ variables using the local barycentric coordinates, $\lambda_{i,j}$, $j = -1, 0, 1$, defined by

$$(18) \quad \mathbf{x} = \sum_{j=-1}^1 \lambda_{i,j}(\mathbf{x}) \mathbf{v}_{i+j}, \quad 1 = \sum_{j=-1}^1 \lambda_{i,j}(\mathbf{x}).$$

Consider first the quadratic case.

THEOREM 2. *Let*

$$F_i = \phi_i \lambda_{i,0}, \quad F_{i,1} = \phi_i \lambda_{i,1} + \phi_{i+1} \lambda_{i+1,-1}, \quad i = 1, \dots, n,$$

and

$$(19) \quad \Psi_2(P_n) = \text{span}\{F_i, F_{i,1}, i = 1, \dots, n\}.$$

Then $\Pi_2 \subset \Psi_2(P_n) \subset \Phi_2(P_n)$.

Proof. For any $p \in \Pi_2$, expanding one of the two \mathbf{x} variables in (17) using the GBCs ϕ_i gives

$$(20) \quad p(\mathbf{x}) = \mathcal{P}(\mathbf{x}, \mathbf{x}) = \sum_{i=1}^n \phi_i(\mathbf{x}) \mathcal{P}(\mathbf{v}_i, \mathbf{x}).$$

Then expanding the remaining \mathbf{x} variable using the coordinates λ_{ij} gives

$$(21) \quad p(\mathbf{x}) = \sum_{i=1}^n \phi_i(\mathbf{x}) \sum_{j=-1}^1 \lambda_{i,j}(\mathbf{x}) \mathcal{P}(\mathbf{v}_i, \mathbf{v}_{i+j}).$$

Considering the two cases $i = j$ and $i \neq j$, and since $\mathcal{P}(\mathbf{v}_i, \mathbf{v}_i) = p(\mathbf{v}_i)$, we thus obtain

$$p(\mathbf{x}) = \sum_{i=1}^n p(\mathbf{v}_i) F_i(\mathbf{x}) + \sum_{i=1}^n \mathcal{P}(\mathbf{v}_i, \mathbf{v}_{i+1}) F_{i,1}(\mathbf{x})$$

with the F_i and $F_{i,1}$ as in (19). This shows that $\Pi_2 \subset \Psi_2(P_n)$. Since each $\lambda_{i,j}$ can be expressed as a linear combination of the ϕ_k , $k = 1, \dots, n$, it also follows that $F_i, F_{i,1} \in \Phi_2(P_n)$. \square

Example 4. In order to understand these functions F_i and $F_{i,1}$ better, we plot them in Figures 1 and 2. These are F_i and $F_{i,1}$ based on Wachspress coordinates. The graphs of F_i and $F_{i,1}$ using mean value coordinates are very similar.

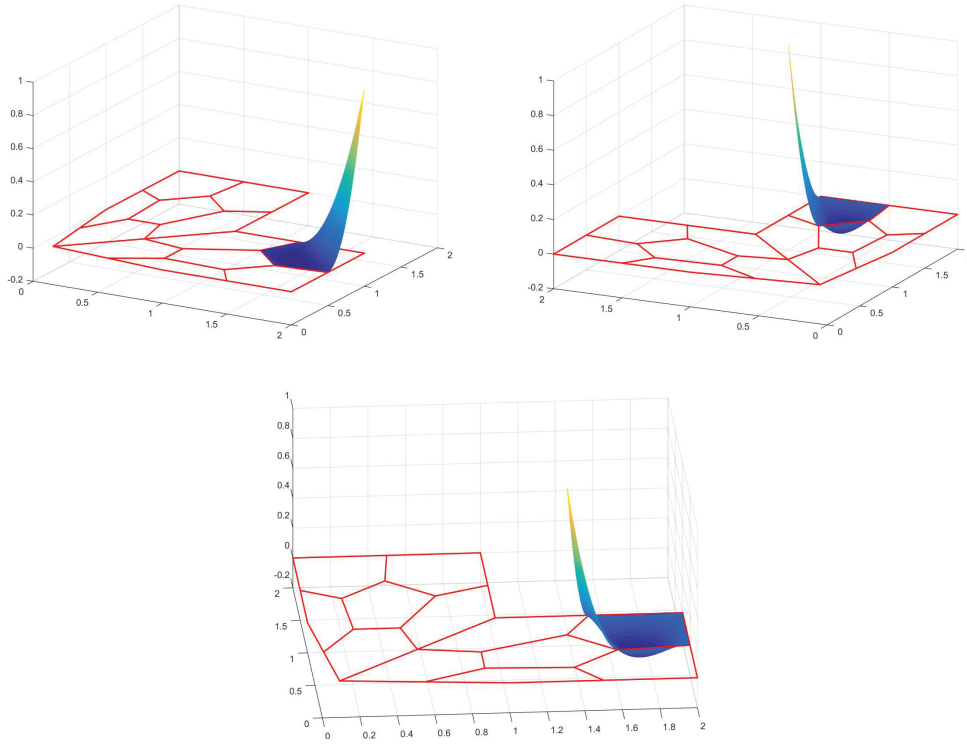


FIG. 1. Some F_i 's based on Wachspress coordinates.

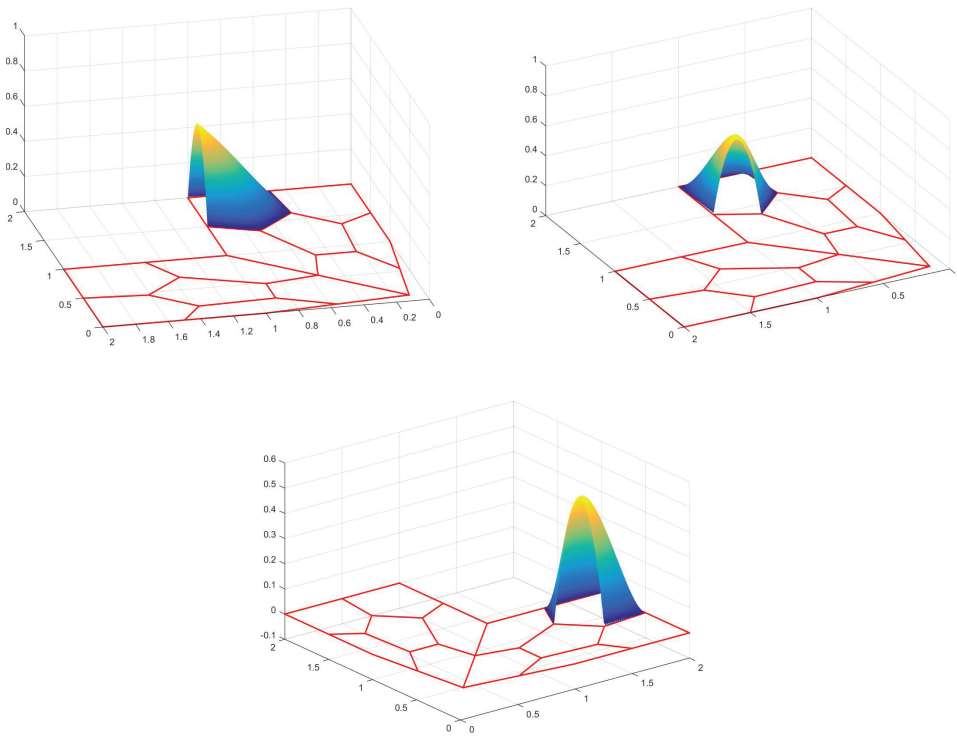


FIG. 2. Some $F_{i,1}$'s based on Wachspress coordinates.

On the edge $[\mathbf{v}_i, \mathbf{v}_{i+1}]$, the only F -functions that are nonzero are F_i , $F_{i,1}$, and F_{i+1} , and if

$$(22) \quad \mathbf{x} = (1 - \mu)\mathbf{v}_i + \mu\mathbf{v}_{i+1}, \quad 0 \leq \mu \leq 1,$$

then

$$(23) \quad F_i(\mathbf{x}) = (1 - \mu)^2, \quad F_{i,1}(\mathbf{x}) = 2\mu(1 - \mu), \quad F_{i+1}(\mathbf{x}) = \mu^2.$$

Thus the $2n$ functions F_i and $F_{i,1}$ are linearly independent.

In constructing a reduced space for order $d \geq 3$ the polar form expansion involves some functions that are zero on the boundary of P_n , and some of these could be linearly dependent. However, by limiting the coordinates ϕ_i to be Wachspress coordinates, we are able to make a precise statement about these dependencies and ascertain the function space and its dimension.

THEOREM 3. *For $d \geq 3$ and with Wachspress coordinates ϕ_i , let*

$$\begin{aligned} F_i &= \phi_i \lambda_{i,0}^{d-1}, \quad i = 1, \dots, n, \\ F_{i,k} &= \binom{d-1}{k} \phi_i \lambda_{i,1}^k \lambda_{i,0}^{d-1-k} + \binom{d-1}{k-1} \phi_{i+1} \lambda_{i+1,0}^{k-1} \lambda_{i+1,-1}^{d-k}, \\ & \quad i = 1, \dots, n, \quad k = 1, \dots, d-1, \end{aligned}$$

and

$$(24) \quad \Psi_d(P_n) := \text{span}\{F_i\} \oplus \text{span}\{F_{i,k}\} \oplus \frac{b}{W} \Pi_{d-3}$$

with W as in (7) and b as in (12). Then $\Pi_d \subset \Psi_d(P_n) \subset \Phi_d(P_n)$.

Proof. For any $p \in \Pi_d$ with $d \geq 3$, expanding one of the \mathbf{x} variables in (17) using any GBCs ϕ_i gives

$$(25) \quad p(\mathbf{x}) = \sum_{i=1}^n \phi_i(\mathbf{x}) q_i(\mathbf{x}),$$

where

$$q_i(\mathbf{x}) = \mathcal{P}(\mathbf{v}_i, \mathbf{x}^{[d-1]}).$$

Then expanding the remaining \mathbf{x} 's using the λ_{ij} gives

$$q_i(\mathbf{x}) = \sum_{j_1, \dots, j_{d-1} = -1}^1 \lambda_{i,j_1}(\mathbf{x}) \cdots \lambda_{i,j_{d-1}}(\mathbf{x}) \mathcal{P}(\mathbf{v}_i, \mathbf{v}_{i+j_1}, \dots, \mathbf{v}_{i+j_{d-1}}),$$

and so q_i is the Bernstein–Bézier polynomial

$$q_i(\mathbf{x}) = \sum_{|\mathbf{j}|=d-1} \mathcal{P}(\mathbf{v}_{i-1}^{[j_1]}, \mathbf{v}_i^{[j_2+1]}, \mathbf{v}_{i+1}^{[j_3]}) B_{\mathbf{j}}^{d-1}(\mathbf{x}),$$

where

$$B_{\mathbf{j}}^{d-1} = \frac{(d-1)!}{j_1! j_2! j_3!} \lambda_{i,-1}^{j_1} \lambda_{i,0}^{j_2} \lambda_{i,1}^{j_3}.$$

By grouping together the cases $\mathbf{j} = (0, d - 1, 0)$, $\mathbf{j} = (k, d - 1 - k, 0)$, and $\mathbf{j} = (0, d - 1 - k, k)$, $k = 1, \dots, d - 1$, we obtain

$$\begin{aligned}
 q_i(\mathbf{x}) &= \mathcal{P}(\mathbf{v}_i^{[d]})\lambda_{i,0}^{d-1}(\mathbf{x}) \\
 &+ \sum_{j=-1,1} \sum_{k=1}^{d-1} \mathcal{P}(\mathbf{v}_i^{[d-k]}, \mathbf{v}_{i+j}^{[k]}) \binom{d-1}{k} \lambda_{i,0}^{d-1-k}(\mathbf{x}) \lambda_{i,j}^k(\mathbf{x}) \\
 (26) \quad &+ \lambda_{i,-1}(\mathbf{x}) \lambda_{i,1}(\mathbf{x}) r_i(\mathbf{x})
 \end{aligned}$$

for some polynomial $r_i \in \Pi_{d-3}$.

Now, for Wachspress coordinates ϕ_i , the polynomial weight function w_i in (6) contains the factors h_j , $j \neq i - 1, i$, and $\lambda_{i,-1}$ contains the factor h_i , and $\lambda_{i,+1}$ contains the factor h_{i-1} , and so there is some constant K_i such that

$$\phi_i(\mathbf{x}) \lambda_{i,-1}(\mathbf{x}) \lambda_{i,1}(\mathbf{x}) = K_i \frac{b(\mathbf{x})}{W(\mathbf{x})},$$

with b the bubble function in (12), and W as in (7). Therefore, substituting (26) into (25) gives

$$(27) \quad p = \sum_{i=1}^n p(\mathbf{v}_i) F_i + \sum_{i=1}^n \sum_{k=1}^{d-1} \mathcal{P}(\mathbf{v}_i^{[d-k]}, \mathbf{v}_{i+1}^{[k]}) F_{i,k} + \frac{b}{W} r$$

with the F_i and $F_{i,k}$ as in (24) and r some polynomial in Π_{d-3} . This shows that $\Pi_d \subset \Psi_d(P_n)$. The sum in (24) is direct due to b being zero on the edges of P_n and the $F_{i,k}$ being zero at the vertices of P_n . \square

Example 5. To help understand these $F_i, F_{i,k}$ better, we plot these F_i (the left one on the top row) and $F_{i,k}, k = 1, 2$ of order 3 as well as a $\psi = b/W$ (the right one on the second row) in Figure 3.

In analogy to the quadratic case, if \mathbf{x} is the boundary point in (22), then

$$F_i(\mathbf{x}) = b_0^d(\mu), \quad F_{i,k}(\mathbf{x}) = b_k^d(\mu), \quad k = 1, \dots, d - 1, \quad F_{i+1}(\mathbf{x}) = b_d^d(\mu),$$

where

$$(28) \quad b_k^d(\mu) = \binom{d}{k} \mu^k (1 - \mu)^{d-k}, \quad k = 0, 1, \dots, d,$$

the univariate Bernstein polynomials of degree d in μ , and all remaining F_i 's are zero on e_i .

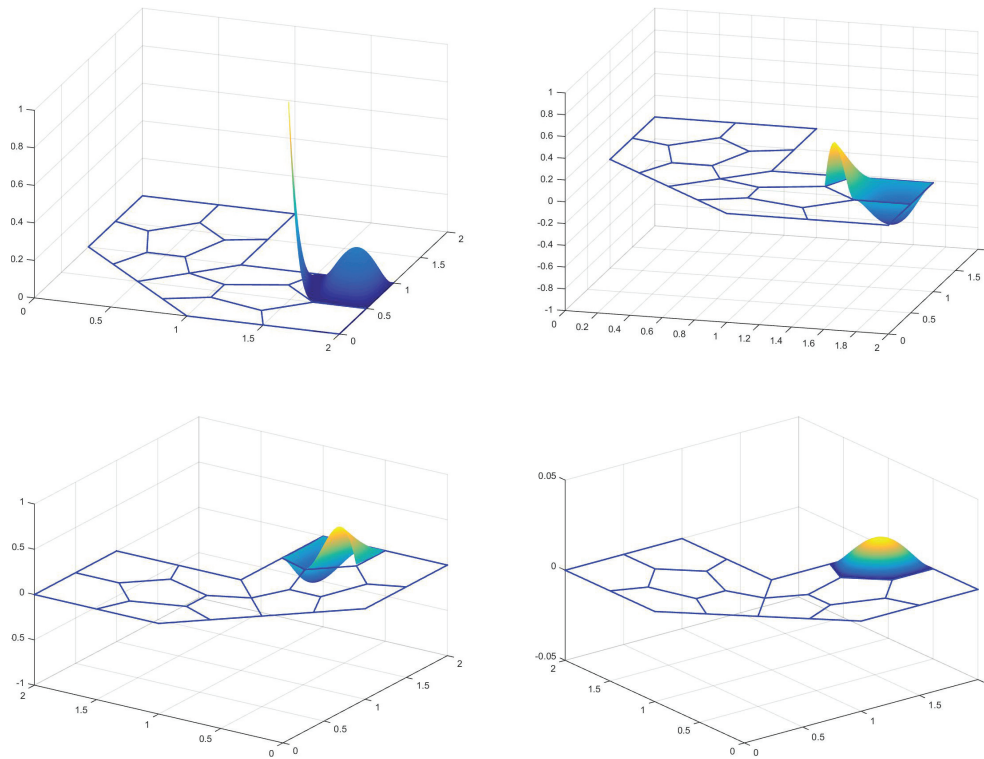
By the linear independence of the b_k^d , the n functions F_i and the $(d - 1)n$ functions $F_{i,k}$ are linearly independent. By the dimension of Π_{d-3} , we have

$$\dim \Psi_d(P_n) = dn + (d - 1)(d - 2)/2.$$

Remark 1. This space dimension is similar to, but less than, that of the virtual element space of [6] which is

$$dn + d(d - 1)/2.$$

The functions in [6, section 4] are defined by the following properties: (i) they are continuous on the boundary of P_n , (ii) their restriction to an edge of P_n is a univariate polynomial of degree $\leq d$, and (iii) their Laplacian is a polynomial of degree $\leq d - 2$ (interpreted in the case $d = 1$ to mean they are harmonic). From (iii) it trivially follows that this space, like $\Psi_d(P_n)$, contains $\Pi_d(P_n)$.

FIG. 3. Some $F_{i,k}$'s of order 3.

4. Polygonal spline spaces. Recall that Bernstein–Bézier polynomials on triangles can be pieced together (continuously and even smoothly) over a collection of triangles to form a bivariate spline function (cf., e.g., [8, 14]). Similarly, polygonal Bernstein–Bézier functions can be pieced together continuously over a collection of convex polygons to form a bivariate polygonal spline function.

Let Δ be a collection of convex polygons and let $\Omega = \bigcup_{P \in \Delta} P$ be the domain consisting of these polygons in Δ . We assume that the interiors of any two polygons P and \tilde{P} do not intersect and the intersection of P and \tilde{P} is either their common edge or common vertex if the intersection is not empty. Then let

$$(29) \quad \mathcal{S}_d(\Delta) = \{s \in C(\Omega), \quad s|_{P_n} \in \Psi_d(P_n), \forall P_n \in \Delta\},$$

where, for general GBCs,

$$\Psi_1(P_n) := \Phi_1(P_n) = \text{span}\{\phi_i, i = 1, \dots, n\},$$

and $\Psi_2(P_n)$ is defined by (19), and for Wachspress coordinates and $d \geq 3$, $\Psi_d(P_n)$ is defined in (24).

4.1. Space dimension. By the results of the previous section, we have the following.

THEOREM 4. *For any GBCs,*

$$\dim(\mathcal{S}_1(\Delta)) = \#(V), \quad \dim(\mathcal{S}_2(\Delta)) = \#(V) + \#(E),$$

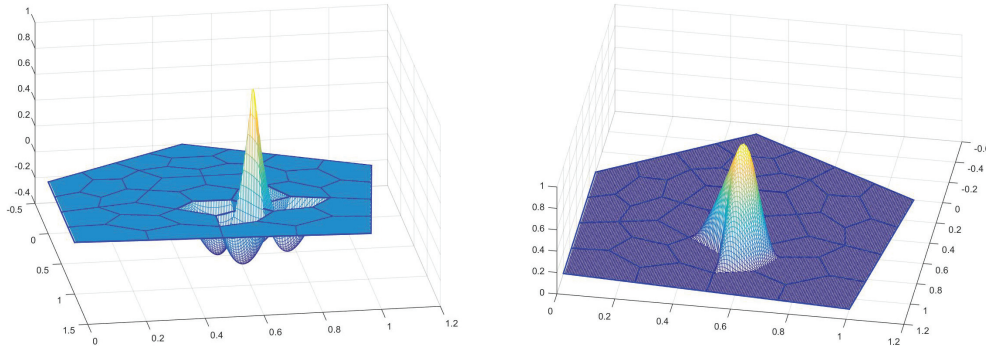


FIG. 4. Vertex and edge splines.

and for Wachspress coordinates and $d \geq 3$,

$$(30) \quad \dim(S_d(\Delta)) = \#(V) + (d - 1)\#(E) + \frac{(d - 1)(d - 2)}{2}\#(\Delta),$$

where V and E are the sets of vertices and edges of Δ , respectively.

Proof. We prove this only for the quadratic case, $d = 2$. The other cases are similar. For each vertex $\mathbf{v} \in \Delta$, let $\mathcal{P}(\mathbf{v})$ be the set of polygons in Δ sharing the vertex \mathbf{v} . For each $P \in \mathcal{P}(\mathbf{v})$, recall the functions F_i and $F_{i,1}$ of (19). Since \mathbf{v} is a vertex of P , we let $F_{\mathbf{v},P}$ be the function F_i associated with vertex \mathbf{v} . Then we define the vertex spline

$$(31) \quad S_{\mathbf{v}}(\mathbf{x}) = \begin{cases} F_{\mathbf{v},P}(\mathbf{x}) & \text{if } \mathbf{x} \in P \in \mathcal{P}_{\mathbf{v}}, \\ 0 & \text{otherwise.} \end{cases}$$

Its support is the union of the polygons in $\mathcal{P}_{\mathbf{v}}$. We now show that $S_{\mathbf{v}}$ is continuous on Ω . For any two polygons P and \tilde{P} in $\mathcal{P}_{\mathbf{v}}$ which share a common edge $[\mathbf{v}, \mathbf{u}]$, we see from (23) that at the point $\mathbf{x} = (1 - \mu)\mathbf{v} + \mu\mathbf{u}$, with $\mu \in [0, 1]$,

$$F_{\mathbf{v},P}(\mathbf{x}) = (1 - \mu)^2 = F_{\mathbf{v},\tilde{P}}(\mathbf{x}).$$

As all $F_{\mathbf{v},P}$ are zero on the edges of P which do not connect to \mathbf{v} , we know that $S_{\mathbf{v}}$ is zero on the boundary of the union of the polygons in $\mathcal{P}_{\mathbf{v}}$, and thus $S_{\mathbf{v}}$ is continuous.

For a boundary edge e , we define S_e similarly. Since the functions $S_{\mathbf{v}}$ and S_e form a basis for $S_2(\Delta)$, the dimension of $S_2(\Delta)$ follows. \square

Next for each interior edge $e = [\mathbf{v}, \mathbf{u}]$ in Δ , there are two polygons P and \tilde{P} sharing e . Let $F_{e,P}$ be the function $F_{i,1}$ of (19) on P associated with e , and $F_{e,\tilde{P}}$ the one on \tilde{P} associated with e . Then we define the edge spline

$$(32) \quad S_e(\mathbf{x}) = \begin{cases} F_{e,P}(\mathbf{x}) & \text{if } \mathbf{x} \in P, \\ F_{e,\tilde{P}}(\mathbf{x}) & \text{if } \mathbf{x} \in \tilde{P}, \\ 0 & \text{otherwise,} \end{cases}$$

whose support is $P \cup \tilde{P}$. Since S_e is zero on the boundary of $P \cup \tilde{P}$, and since at the point $\mathbf{x} = (1 - \mu)\mathbf{v} + \mu\mathbf{u}$, $\mu \in [0, 1]$,

$$F_{e,P}(\mathbf{x}) = 2\mu(1 - \mu) = F_{e,\tilde{P}}(\mathbf{x}),$$

it follows that S_e is continuous on Ω . Figure 4 shows examples of $S_{\mathbf{v}}$ and S_e .

4.2. Interpolation. We now explain how to use the functions in $S_d(\Delta)$ for interpolation. Given the values $f(\mathbf{v})$ of a function f defined at the vertices $\mathbf{v} \in V$ of Δ there is a unique function $S_f \in S_1(\Delta)$ satisfying

$$(33) \quad S_f(\mathbf{v}) = f(\mathbf{v}), \quad \mathbf{v} \in V,$$

namely,

$$(34) \quad S_f = \sum_{\mathbf{v} \in V} f(\mathbf{v})S_{\mathbf{v}},$$

where $S_{\mathbf{v}}$ is a vertex spline in $S_1(\Delta)$, analogous to (31). From the definition of GBCs, S_f reproduces linear polynomials. We now extend the study to find Lagrange functions for interpolation for degree $d \geq 2$. We begin with $d = 2$ and have the following.

THEOREM 5. *For a fixed polygon P_n in Δ , the functions defined by*

$$(35) \quad \begin{aligned} L_i &= -\frac{1}{2}F_{i-1,1} + F_i - \frac{1}{2}F_{i,1}, \quad i = 1, \dots, n, \\ L_{i,1} &= 2F_{i,1}, \quad i = 1, \dots, n, \end{aligned}$$

are an interpolatory basis for $\Psi_2(P_n)$, i.e., for any function $f : P_n \rightarrow \mathbb{R}$, if $s_f \in \Psi_2(P_n)$ is defined by

$$(36) \quad s_f(\mathbf{x}) = \sum_{i=1}^n f(\mathbf{v}_i)L_i(\mathbf{x}) + \sum_{i=1}^n f(\mathbf{v}_{i,1})L_{i,1}(\mathbf{x}), \quad \mathbf{x} \in P_n,$$

then $s_f = f$ for all $f \in \Psi_2(P_n)$.

Proof. Let us start with a single polygon P_n . Following an observation of [23] we can transform the functions F_i and $F_{i,1}$ into an interpolatory basis, interpolating at the vertices \mathbf{v}_j and the midpoints

$$\mathbf{v}_{j,1} := (\mathbf{v}_j + \mathbf{v}_{j+1})/2.$$

Indeed, since

$$F_i(\mathbf{v}_j) = \delta_{ij}, \quad F_i(\mathbf{v}_{i-1,1}) = F_i(\mathbf{v}_{i,1}) = 1/4, \quad F_i(\mathbf{v}_{j,1}) = 0, \quad j \neq i-1, i,$$

and

$$F_{i,1}(\mathbf{v}_j) = 0, \quad F_{i,1}(\mathbf{v}_{j,1}) = \delta_{ij}/2,$$

we can easily check that when $f = F_i$, s_f defined in (36) is equal to $L_i - \frac{1}{4}L_{i-1,1} - \frac{1}{4}L_{i,1} = F_i$. We proceed similarly for the case $f = F_{i,1}$. These complete the proof. \square

We are now ready to consider $d \geq 3$ with Wachspress coordinates. Let ℓ_i^d , $i = 0, 1, \dots, d$, be the univariate Lagrange fundamental polynomials of degree $\leq d$ w.r.t. the points j/d , $j = 0, 1, \dots, d$, satisfying $\ell_i^d(j/d) = \delta_{ij}$. Then the inner polynomials in (28) can be expressed as

$$b_i^d = \sum_{j=1}^{d-1} b_i^d(j/d)\ell_j^d, \quad i = 1, \dots, d-1.$$

Then if $A = [a_{ij}]$ is the inverse of the matrix $[b_i^d(j/d)]_{i,j=1,\dots,d}$, we know that

$$\ell_i^d = \sum_{j=1}^{d-1} a_{ij} b_j^d, \quad i = 1, \dots, d-1.$$

It follows that the functions

$$L_{i,k} = \sum_{r=1}^{d-1} a_{kr} F_{i,r}, \quad i = 1, \dots, n, \quad k = 1, \dots, d-1,$$

$$L_i = F_i - \sum_{k=1}^{d-1} (1 - k/d)^d (L_{i,k} + L_{i-1,d-k}), \quad i = 1, \dots, n,$$

are an interpolatory basis for the “boundary” part

$$\Psi_{d,B}(P_n) := \text{span}\{F_i\} \oplus \text{span}\{F_{i,k}\}$$

of $\Psi_d(P_n)$, with respect to the side nodes,

$$\mathbf{v}_{i,k} := \frac{(d-k)\mathbf{v}_i + k\mathbf{v}_{i+1}}{d}, \quad k = 1, \dots, d-1.$$

In other words

$$I_B f(\mathbf{x}) := \sum_{i=1}^n f(\mathbf{v}_i) L_i(\mathbf{x}) + \sum_{i=1}^n \sum_{k=1}^{d-1} f(\mathbf{v}_{i,k}) L_{i,k}(\mathbf{x}), \quad \mathbf{x} \in P_n,$$

is the unique interpolant to f in $\Psi_{d,B}(P_n)$. For example, when $d = 3$,

$$\begin{bmatrix} b_1^3 \\ b_2^3 \end{bmatrix} = \frac{2}{9} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} l_1^3 \\ l_2^3 \end{bmatrix},$$

and then

$$\begin{bmatrix} l_1^3 \\ l_2^3 \end{bmatrix} = \frac{3}{2} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} b_1^3 \\ b_2^3 \end{bmatrix},$$

and so the interpolatory basis is

$$L_{i,1} = \frac{3}{2}(2F_{i,1} - F_{i,2}), \quad L_{i,2} = \frac{3}{2}(-F_{i,1} + 2F_{i,2}), \quad i = 1, \dots, n,$$

$$(37) \quad L_i = F_i - \frac{8}{27}(L_{i,1} + L_{i-1,2}) - \frac{1}{27}(L_{i,2} + L_{i-1,1}), \quad i = 1, \dots, n.$$

Example 6. We plot the L_i 's (on the top row) and $L_{i,k}$'s, $k = 1, 2$ of order 3 (on the second row) in Figure 5.

Interpolation in the whole space $\Psi_d(P_n)$ can now be carried out in two steps: by first interpolating f using $\Psi_{d,B}(P_n)$ and then interpolating the error on a set of unisolvent points in the interior of P_n and combining the two terms. Let $\xi_{\mathbf{j}} \in P_n$, $\mathbf{j} \in \mathbb{N}_0^3$, $|\mathbf{j}| = d - 3$, be a set of distinct points which are unisolvent for polynomial interpolation in Π_{d-3} . Let $\tilde{L}_{\mathbf{j}} \in \Pi_{d-3}$, $|\mathbf{j}| = d - 3$, be the Lagrange fundamental polynomials for these points, i.e., $\tilde{L}_{\mathbf{j}}(\xi_{\mathbf{k}}) = \delta_{\mathbf{j},\mathbf{k}}$. Then

$$(38) \quad s_f := I_B f + \frac{b}{W} \sum_{|\mathbf{j}|=d-3} \frac{W(\xi_{\mathbf{j}})}{b(\xi_{\mathbf{j}})} (f(\xi_{\mathbf{j}}) - I_B f(\xi_{\mathbf{j}})) \tilde{L}_{\mathbf{j}}$$

is an interpolant to f in $\Psi_d(P_n)$.

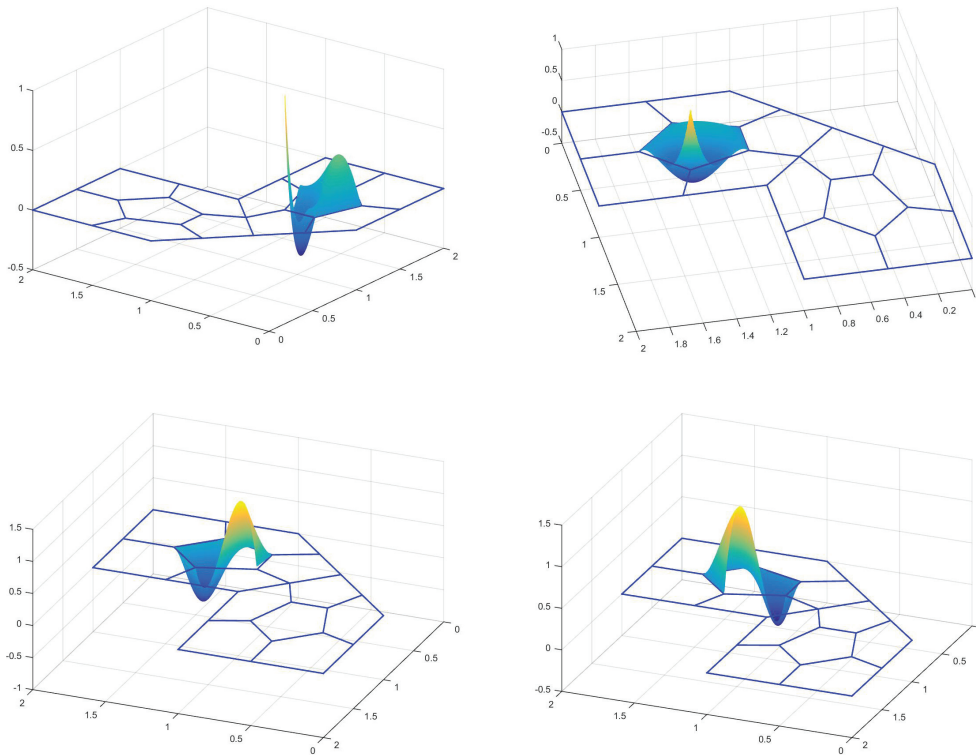


FIG. 5. Some L_i 's and $L_{i,k}$'s of order 3.

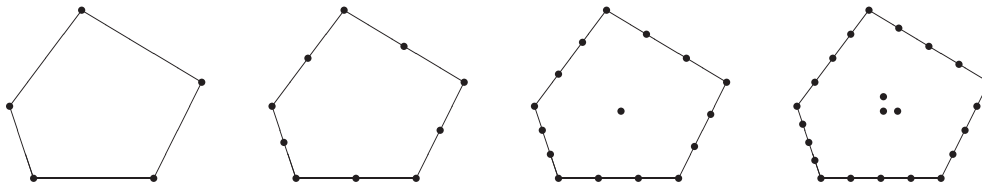


FIG. 6. Linear, quadratic, cubic, and quartic elements.

For $d = 3$ we could choose any single point ξ_{000} in P_n . For $d \geq 4$ we could choose a triangular grid of points

$$\xi_{\mathbf{j}} := \frac{j_1 \mathbf{p}_1 + j_2 \mathbf{p}_2 + j_3 \mathbf{p}_3}{d - 3}, \quad |\mathbf{j}| = d - 3,$$

where $T = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ is a triangle inside P_n . Figure 6 shows possible configurations of boundary and interior points for interpolation.

This explains how to use polygonal Bernstein–Bézier functions to do interpolation on a single polygon. Clearly, we can extend the interpolation scheme to a polygonal partition Δ so that we can use the spline functions in $S_d(\Delta)$ to interpolate any continuous function f at the vertices of a polygonal partition.

Remark 2. Certainly, it is interesting to study a smooth polygonal spline space, e.g.,

$$(39) \quad \mathcal{S}_d^r(\Delta) = \{s \in C^r(\Omega), s|_{P_n} \in \Psi_d(P_n), \forall P_n \in \Delta\}$$

for $r \geq 1$. This is one of our future research topics.

5. Numerical solution of Poisson equation. We will use the polygonal splines of order $d \geq 2$ to solve the Dirichlet boundary value problem

$$(40) \quad \begin{cases} -\Delta u = f, & x \in \Omega \subset \mathbb{R}^2, \\ u = g, & x \in \partial\Omega, \end{cases}$$

where Ω is a polygonal domain and Δ is the Laplace operator. Our main point is to provide a proof of concept that these splines can be used to solve PDEs numerically in analogy to any standard finite element method, macroelement method, virtual element method, multivariate splines, and polygonal finite elements of order 1, etc.; see, for example, [7], [4], [3], [2], [5], [6], [19]. Although we employ the standard weak formulation of the Poisson equation, we shall use the constraint minimization approach to find a numerical solution. This approach is different than the traditional finite element method and thus we give an outline of it below.

5.1. Basics on weak solution. For simplicity, let us illustrate the approach with some simple partitions. We will take Δ to consist either of quadrilaterals or of pentagons. A method of generating a quadrangulation of Ω can be found in, e.g., [14]. For a pentagonal partition we will propose a refinement scheme later in this section.

Now let $S_d(\Delta)$ be the collection of polygonal spline functions of degree $d \geq 1$ over Δ . Let $h = |\Delta|$ be the size of Δ . That is, h is the longest of the length of edges of Δ . Let $B(u, v) = \int_{\Omega} \nabla u \cdot \nabla v dx$ be the standard bilinear form associated with the Poisson equation. Our numerical method is to solve the following weak solution: for $u_h \in S_d(\Delta) \cap H_0^1(\Omega)$ such that

$$(41) \quad B(u_h, v_h) = \langle f, v_h \rangle, \forall v_h \in S_d(\Delta) \cap H_0^1(\Omega).$$

Using the standard argument in the finite element method (see the proof of the Céa lemma; cf. [7] and [3]), we have the following.

LEMMA 1. *Let u_h be an approximate weak solution satisfying (41) in $V_h = S_d(\Delta) \cap H_0^1(\Omega)$. Then*

$$\|u - u_h\|_{H_0^1(\Omega)} \leq \frac{1}{m} \min_{v \in V_h} \|u - v\|_{H_0^1(\Omega)},$$

where $m > 0$ is a constant dependent on the domain Ω (see p. 104 in [7] or p. 62 in [3]).

As $S_d(\Delta)$ is a space containing all piecewise polynomials of degree d over Δ , it is known, i.e., by the Bramble–Hilbert lemma that if $u \in H^{d+1}(\Omega)$, there exists an interpolant $S_u \in S_d(\Delta)$ such that

$$(42) \quad \|u - S_u\|_{L_2(\Omega)} \leq C_1 h^{d+1} \text{ and } \|\nabla(u - S_u)\|_{L_2(\Omega)} \leq C_2 h^d$$

for some positive constants C_1 and C_2 which are dependent on the chunkiness of the underlying polygons Δ (cf. [3], [14]) as well as the seminorm of u in $H^{d+1}(\Omega)$. Thus, combining Lemma 1 and the second inequality in (42), one has the following.

THEOREM 6 (the serendipity finite element approximation of the weak solution). *Suppose that the weak solution $u \in H^{d+1}(\Omega)$. Then*

$$\|u - u_h\|_{H_0^1(\Omega)} \leq \frac{C_3}{m} |u|_{H^{d+1}(\Omega)} h^d,$$

where C_3 is a positive constant dependent on the chunkiness of the underlying polygons \square

5.2. Numerical solution of the Poisson equation. We have implemented these polygonal spline functions of order d with $d = 2, d = 3$ over arbitrary partition Δ

in MATLAB for numerical solution of Poisson equations. Our implementation follows the approach discussed in [2] based on multivariate splines for numerical solutions of many linear and nonlinear PDEs. See similar approaches in [17] and [13].

As the approach is not a standard one in finite element literature, we summarize the steps as follows. For convenience, let us focus on $d = 2$. A set of our MATLAB programs can be downloaded from the webpage alpha.math.uga.edu/~mjla1/indexPublications.html

- (1) First, we use a coefficient vector \mathbf{c} ($2n$ per polygon with n sides over the list of polygons in Δ) to represent a polygonal spline function in $S_2(\Delta)$. Note that the vector \mathbf{c} is redundant which is different from the traditional finite element method. An advantage of such a vector \mathbf{c} is that it allows us to compute the mass and stiffness matrix in parallel and the function evaluation can be done in parallel.
- (2) Second, when two polygons share a common edge e , there are 3 pairs of the polygonal spline functions supported on these two polygons and each pair have the same value on the common edge e due to the continuity of our polygonal splines. For each pair, they belong to the same locally supported polygonal spline function and thus, the coefficients associated with each pair should be the same. In this way we build a continuity condition matrix H such that $H\mathbf{c} = 0$ represents all these continuity conditions among the coefficients in \mathbf{c} .
- (3) Third, we form the mass and stiffness matrices by using the tensor product of the Gauss quadrature formula with order 5 or larger. Note that our mass and stiffness matrices are block diagonal due to the redundancy of vector \mathbf{c} . For example, the mass matrix $M = \text{diag}(M_p, p \in \Delta)$ with $M_p = [M_{p,i,j}]_{i,j=1,\dots,2n}$ with $n = n(p)$ being the number of sides of p , where

$$M_{p,i,j} = \int_p L_i L_j dx dy$$

with $L_i = L_{i,p}$ being interpolatory polygonal spline functions in (35) which are defined on p . When $p = P_n$ is a quadrilateral with $n = 4$, we simply use a tensor product of the Gauss quadrature formula with order 5 or larger which is exact for all polynomials of degree $d \leq 5$. When $n > 4$, we divide p into a few (strictly convex or degenerated) quadrilaterals so that we can apply the tensor product of the Gauss quadrature formula to each quadrilateral and then add these numerical values together to form a numerical approximation of the integral. For example, when p is a pentagon, we first choose the middle point from the longest edge of p and divide p into two quadrilaterals. Next we apply the Gauss quadrature of order 5 to each of two quadrilaterals and then add two numerical values together to yield a numerical approximation of the integral. We proceed similarly for the stiffness matrix K .

- (4) Then we use the interpolatory scheme discussed in a previous section to find the polygonal interpolant I_f and then use the mass matrix to generate an approximation to the right-hand side $\langle f, L_{i,p} \rangle$ in (41) for $v = L_{i,p}$. That is, we approximate f by $I_f = \sum_{p \in \Delta} \sum_{i=1}^{n(p)} f(\mathbf{v}_i(p)) L_{2i-1} + f((\mathbf{v}_i(p) + \mathbf{v}_{i+1}(p))/2) L_{2i}$ and then compute

$$\begin{aligned} \langle f, L_{i,p} \rangle &= \sum_{j=1}^{n(p)} f(\mathbf{v}_j(p)) \int_p L_{2j-1} L_{i,p} dx dy \\ &\quad + f\left(\frac{\mathbf{v}_i(p) + \mathbf{v}_{i+1}(p)}{2}\right) \int_p L_{2i} L_{i,p} dx dy, \end{aligned}$$

where the two integrals are already calculated in the mass matrix. Here $\mathbf{v}_i(p)$, $i = 1, \dots, n$ are vertices of p . These are entries for the right-hand side vector \mathbf{F} .

- (5) Fifth, boundary conditions are also generated by using the interpolation scheme. We can write them together in terms of a system of linear equations $B\mathbf{c} = G$.
- (6) Finally, it is known that the Poisson equation (40) is the Euler–Lagrange equation of the following minimization,

$$(43) \quad \min_{\substack{u \in H^1(\Omega) \\ u=g \text{ on } \partial\Omega}} \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx dy - \int_{\Omega} f(x, y)u(x, y) dx dy.$$

In the finite dimensional setting, the weak solution (41) is the Euler–Lagrange equation of the minimization

$$(44) \quad \min_{\substack{u_h \in S_d(\Delta) \cap H_0^1(\Omega) \\ u_h=g}} \frac{1}{2} \int_{\Omega} |\nabla u_h|^2 dx dy - \int_{\Omega} f(x, y)u(x, y) dx dy.$$

In terms of the coefficient vector \mathbf{c} of polygonal spline functions, we numerically solve the following constrained minimization problem

$$(45) \quad \min_{\mathbf{c}} \frac{1}{2} \mathbf{c}^{\top} K \mathbf{c} - \mathbf{F}^{\top} M \mathbf{c}, \quad H \mathbf{c} = 0, B \mathbf{c} = G,$$

where $H\mathbf{c} = 0$ represents all the continuity conditions and $B\mathbf{c} = G$ all the boundary conditions, M and K are mass and stiffness matrices, and $\mathbf{F}^{\top} M \mathbf{c}$ the right-hand side of the weak formulation (41). This minimization problem (45) can be solved using the iterative method discussed in [2] in a few iterative steps. In our computation, we only do 5 iterative steps.

The rest is to evaluate the polygonal spline solution and its gradient over the domain. In our experiments, we evaluate the numerical solution and its gradient at those points from 1,002,001 equally spaced points over the minimal rectangle containing Ω that are located within Ω to display the surface of the solution. We also use the values for computing the RMSE of the solution against the exact solution to see the performance of our polygonal splines. As the gradient of polygonal splines is not continuous, we have to exclude the points located on the edges of the underlying polygonal partition so that we are able to compute the accuracy of the gradients. For the domain Ω listed in Figure 8, there are 726,551 points located inside Ω , but not on the edges of the partition. The RMSE estimates for numerical solutions and their gradients are based on the values over these 726,551 points. In finite element literature, the errors in the L_2 norm and H^1 norm of the numerical solutions are usually calculated. The calculation is heavily dependent on the accuracy of the numerical quadrature for the exact solution. That is, the numerical integration of the L_2 norm of the exact solution has to be more accurate than the L_2 error of the numerical solution. In particular, when the partition is refined and the solution is very accurate, it is hard to know how accurate the numerical quadrature should be. On the other hand, the RMSE approach does not need such a quadrature formulation and hence, is a much simpler calculation. In addition, the RMSE is widely used in all other areas of computational sciences such as statistics, machine learning, image processing, and so on.

5.3. A refinement scheme of pentagonal partitions. Refinements of a triangulation and a quadriangulation are well known. Let us introduce a scheme to

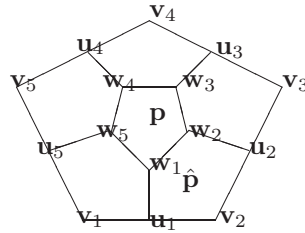


FIG. 7. An illustration of the pentagon refinement with vertices $\mathbf{v}_i, \mathbf{u}_i, \mathbf{w}_i, i = 1, \dots, 5$.

refine a pentagon partition Δ . We start with a pentagon $P = \langle \mathbf{v}_1, \dots, \mathbf{v}_5 \rangle$. Let $\mathbf{p} = (\mathbf{v}_1 + \dots + \mathbf{v}_5)/5$ be the geometric center of P . Write $\mathbf{u}_i = (\mathbf{v}_i + \mathbf{v}_{i+1})/2$ for $i = 1, \dots, 5$ with $\mathbf{v}_6 := \mathbf{v}_1$ and $\mathbf{w}_i = (\mathbf{u}_i + \mathbf{p})/2, i = 1, \dots, 5$, as shown in Figure 7. Connect these $\mathbf{v}_i, \mathbf{u}_i, \mathbf{w}_i, i = 1, \dots, 5$ as shown to form a uniform refinement of the pentagon P .

For a collection Δ of pentagons, we apply the above refinement scheme to each pentagon from Δ . Let us recall that the size $|\Delta|$ of Δ is the longest of the lengths of edges in Δ . An elementary proof shows the following.

LEMMA 2. Let Δ_1 be the refinement of pentagon partition Δ . Then

$$(46) \quad |\Delta_1| \leq \frac{1}{2} |\Delta|.$$

If we let $d(P)$ be the smallest of the diameters of the circle containing P , then we know that $|P| \rightarrow 0$ is equivalent to $d(P) \rightarrow 0$.

In addition to refining triangles, quadrilaterals, pentagons, one is also interested in how to refine a general convex n -gon for $n \geq 6$. The interested reader is referred to [16] for detail.

5.4. Numerical results of polygonal splines. We now use our MATLAB code to show some numerical results of Poisson equations based on polygonal spline functions. We have tested that our codes for $d = 2$ and $d = 3$ can reproduce all polynomials of degree 2 and degree 3, respectively, based on Wachspress coordinates. As we explained above, we shall compute the RMSE for the solution and its gradient to show the performance of polygonal spline functions in Tables 1–8. All the numerical results are obtained by using a laptop computer Dell XPS which has 4 CPU cores with 4 GB memory and speed at 2.20 GHz. The MATLAB parallel computation toolbox was used.

TABLE 1

Convergence of polygonal spline of order 2 over repeatedly refined pentagon partitions in Figure 8, where $u_1 = 1/(1 + x^2 + y^2)$.

DoF	RMSE (u_1)	Rate	RMSE (∇u_1)	Rate	Etime	Stime	MKtime
76	0.00236333		0.022119		2.83	0.01	0.47
392	0.00016613	3.83	0.004138	2.42	4.21	0.02	2.41
2224	0.00001388	3.58	0.000798	2.37	17.73	0.03	14.00
13088	0.00000120	3.53	0.000155	2.36	99.41	0.19	84.37
78016	0.00000010	3.53	0.000031	2.34	586.47	1.42	663.16

TABLE 2

Convergence of quadratic finite element over repeatedly refined triangulations as in Figure 9, where $u_1 = 1/(1 + x^2 + y^2)$.

DoF	RMSE (u_1)	Rate	RMSE (∇u_1)	Rate	Etime	Stime	MKtime
88	0.00141741		0.017789		3.18	0.04	0.49
320	0.00016451	3.11	0.004509	1.98	15.64	0.07	0.63
1216	0.00001996	3.04	0.001131	1.99	58.55	0.19	2.18
4736	0.00000248	3.01	0.000283	2.00	227.27	0.70	9.01
18688	0.00000031	3.00	0.000071	2.00	907.29	2.63	42.55

TABLE 3

Convergence of polygonal spline of order 2 over repeatedly refined pentagon partitions in Figure 8, where $u_2 = x^4 + y^4$.

DoF	RMSE (u_2)	Rate	RMSE (∇u_2)	Rate	Etime	Stime	MKtime
76	0.031270	0.00	0.446502	0.00	2.72	0.002	0.47
392	0.002379	3.72	0.072348	2.63	4.26	0.003	2.44
2224	0.000195	3.61	0.013119	2.46	17.68	0.03	13.50
13088	0.000016	3.57	0.002501	2.39	99.47	0.19	84.26
78016	0.000002	3.37	0.000550	2.19	585.15	1.38	664.99

TABLE 4

Convergence of quadratic finite element over repeatedly refined triangulations as in Figure 9, where $u_2 = x^4 + y^4$.

DoF	RMSE (u_2)	Rate	RMSE (∇u_2)	Rate	Etime	Stime	MKtime
88	0.0241256	0.00	0.313222	0.00	5.22	0.003	0.23
320	0.0030206	3.00	0.078873	1.99	15.42	0.06	0.65
1216	0.0003777	3.00	0.019763	2.00	56.81	0.19	2.22
4736	0.0000472	3.00	0.004945	2.00	223.75	0.66	9.16
18688	0.0000059	3.00	0.001236	2.00	898.80	2.57	42.66

TABLE 5

Convergence of polygonal spline of order 2 over repeatedly refined pentagon partitions in Figure 8, where $u_3 = \sin((x^2 + y^2)\pi) + 1$.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
76	1.417091		8.211055		2.63	0.001	0.004
392	0.065557	4.43	1.227940	2.74	4.24	0.003	2.41
2224	0.004502	3.86	0.243838	2.33	17.70	0.024	13.72
13088	0.000355	3.66	0.047945	2.35	100.36	0.199	86.14
78016	0.000029	3.59	0.009245	2.37	595.08	1.421	658.04

TABLE 6

Convergence of quadratic finite element over repeatedly refined triangulations as in Figure 9, where $u_3 = \sin((x^2 + y^2)\pi) + 1$.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
88	0.626154		5.724203		5.38	0.03	0.26
320	0.107462	2.54	2.079933	1.46	15.98	0.06	0.64
1216	0.012978	3.05	0.608579	1.77	58.13	0.19	2.25
4736	0.001498	3.11	0.159646	1.93	227.82	0.67	9.20
18688	0.000182	3.04	0.040416	1.98	898.31	2.59	42.84

TABLE 7

Convergence of polygonal spline of order 2 over repeatedly refined pentagon partitions in Figure 8, where $u_4 = 10 \exp(-x^2 - y^2)$.

DoF	RMSE (u_4)	Rate	RMSE (∇u_4)	Rate	Etime	Stime	MKtime
76	0.026166		0.319205		2.65	0.002	0.52
396	0.001915	3.77	0.053791	2.57	4.21	0.002	2.30
2224	0.000161	3.57	0.010215	2.40	17.74	0.02	13.76
13088	0.000014	3.54	0.001975	2.37	100.19	0.18	86.41
78016	0.000001	3.53	0.000389	2.34	586.68	1.42	656.69

TABLE 8

Convergence of quadratic finite element over repeatedly refined triangulations as in Figure 9, where $u_4 = 10 \exp(-x^2 - y^2)$.

DoF	RMSE (u_4)	Rate	RMSE (∇u_4)	Rate	Etime	Stime	MKtime
88	0.018453	0.00	0.235958	0.00	3.59	0.04	0.48
320	0.002149	3.10	0.059854	1.98	10.65	0.03	0.32
1216	0.000263	3.03	0.015053	1.99	40.56	0.10	1.13
4736	0.000032	3.01	0.003770	2.00	159.75	0.34	4.52
18688	0.000004	3.00	0.000943	2.00	643.77	1.30	21.56

Example 7. We have performed several tests on smooth solutions

$$\begin{aligned} u_1(x, y) &= 1/(1 + x^2 + y^2), & u_2(x, y) &= x^4 + y^4, \\ u_3(x, y) &= \sin(\pi(x^2 + y^2)) + 1, & u_4(x, y) &= 10 \exp(-x^2 - y^2), \end{aligned}$$

to check the performance of our polygonal splines of order 2 based on partitions obtained by repeated pentagonal refinements as in Figure 8. In Tables 1–8, we use Etime, Stime, MKtime for the times for evaluation, solution of the linear system, and generation of mass and stiffness matrices, respectively. These three times Etime, Stime, MKtime are measured in seconds. Let s_1 be the polygonal spline solution of u_1 . The RMSE (u_1) is

$$\text{RMSE}(u_1) = \sqrt{\frac{1}{N} \sum_{\substack{i,j=1 \\ (x_i, y_j) \in \Omega \setminus E}}^{1001} (s_1(x_i, y_j) - u_1(x_i, y_j))^2},$$

where $N = \#\{(x_i, y_i) \in \Omega \setminus E, i, j = 1, \dots, 1001\}$, where E is the collection of all edges of Δ . Our reason to exclude the points on E is because of the calculation of the gradient values of polygonal splines. We proceed similarly for the $\text{RMSE}(\nabla u_1)$. That is,

$$\begin{aligned} &\text{RMSE}(\nabla u_1) \\ &= \sqrt{\frac{1}{N} \sum_{\substack{i,j=1 \\ (x_i, y_j) \in \Omega \setminus E}}^{1001} (D_x s_1(x_i, y_j) - D_x u_1(x_i, y_j))^2 + (D_y s_1(x_i, y_j) - D_y u_1(x_i, y_j))^2}. \end{aligned}$$

It is interesting to know the rate of convergence. By Lemma 2, the size of Δ is reduced by $1/2$. The RMSE err_ℓ of u_i at refinement level $\ell > 0$ is defined by

$$(47) \quad \text{err}_\ell = O(1/2^\ell)^\alpha$$

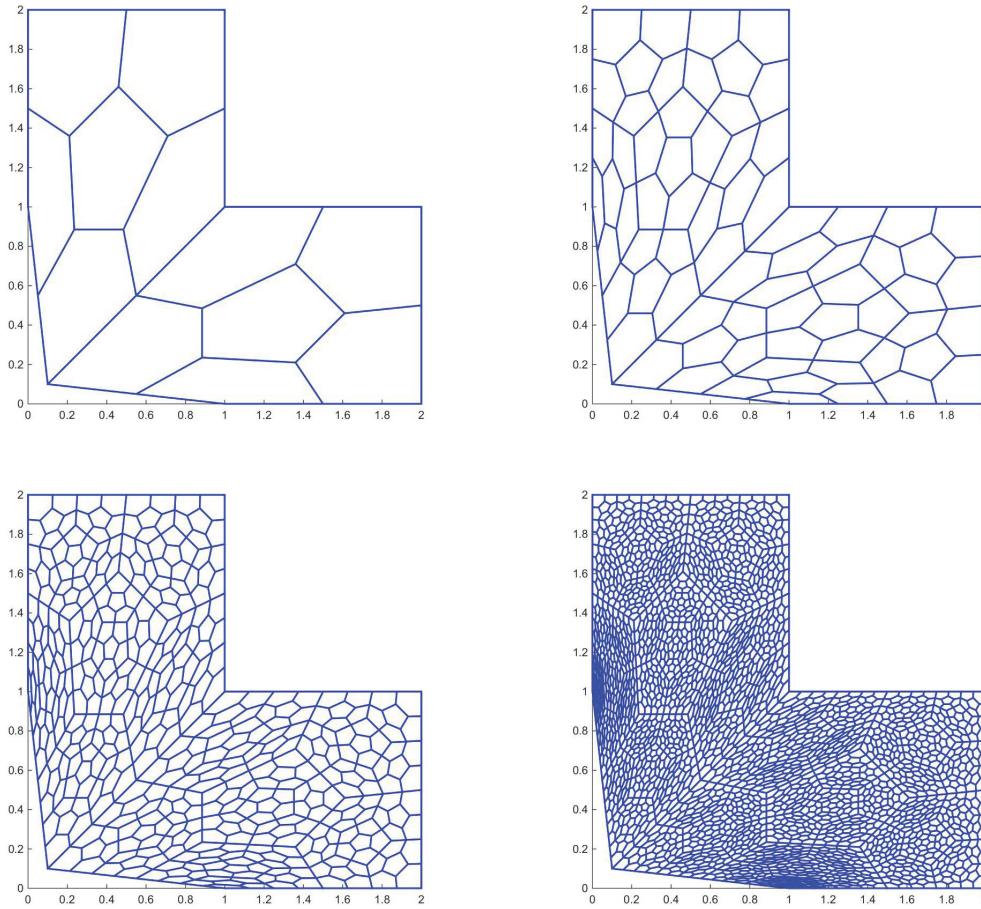


FIG. 8. A pentagon oartition (top-left) and its refinements.

for some $\alpha > 0$, where $i = 1, 2, 3, 4$. Similarly we can compute the rate of convergence of ∇u_i .

These can be compared with the convergence of the standard quadrature finite elements for these exact solution. To do so, we include a table for quadrature finite element for solution $u_i, i = 1, 2, 3, 4$ in Tables 2, 4, 6, and 8.

These numerical results in Tables 1–8 show that the polygonal spline of order 2 is able to approximate the solution very well with $\alpha \approx 3.4$ for function values and $\alpha \approx 2.4$ for gradient values, even though we are not able to prove the boundedness of the chunkiness of refined pentagon partitions.

The computational times for evaluation for the total 726,551 points located inside Ω , for solution of the associated linear systems and for generation of mass and stiffness matrices show that the polygonal splines are better for evaluation and solution while they are significantly worse than the traditional finite element method for the mass and stiffness matrices generation. More study is needed to speed up the generating mass and stiffness matrices. One possible approach is to use a GPU.

Remark 3. From Tables 1–7, we see that assembling mass and stiffness matrices costs a lot of time. This method may be useful if we use a computer with a GPU to

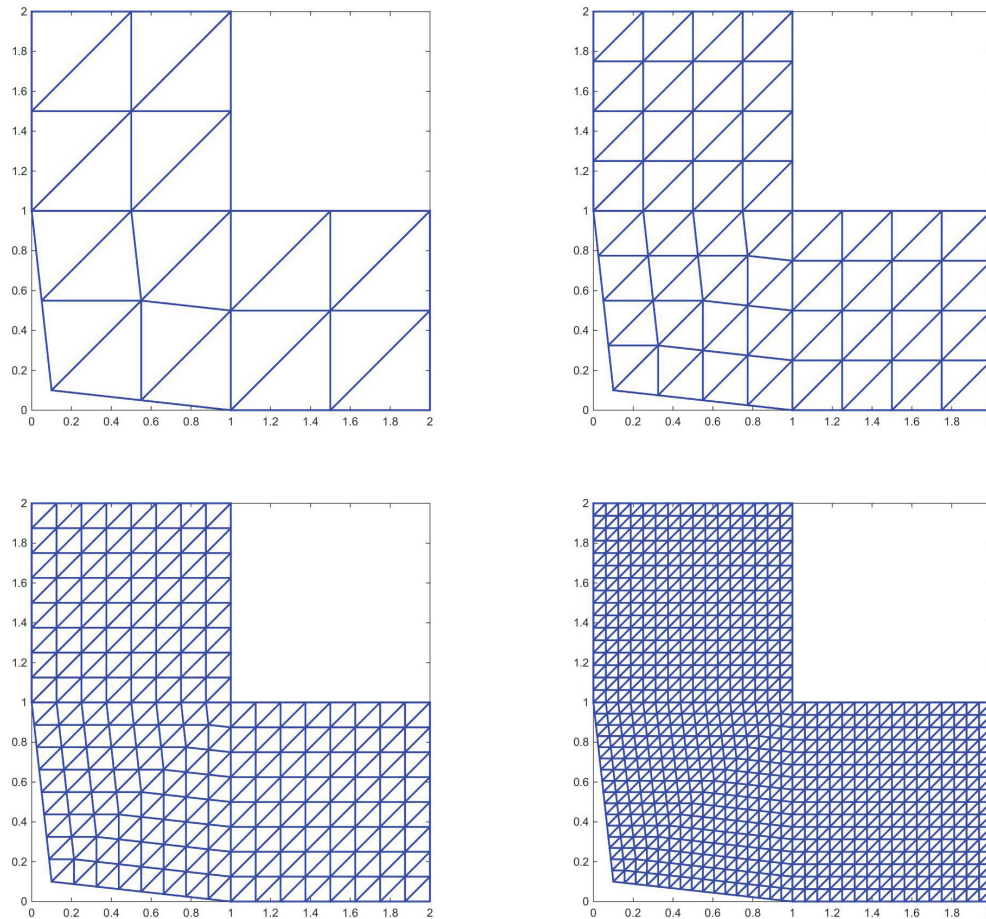


FIG. 9. A triangulation (top-left) and its refinements.

solve a nonlinear Poisson equation, e.g.,

$$(48) \quad \begin{cases} -\Delta u + f(u, x) = 0, & x \in \Omega \subset \mathbb{R}^2, \\ u = g, & x \in \partial\Omega, \end{cases}$$

where $f(u, x)$ is a nonlinear function of u , say $f(u, x) = u(1 - u)$, $f(u, x) = u^3$, $f(u, x) = \exp(u)$, and so on.

Example 8. In addition to the numerical results based on Wachspress coordinates, let us present some results using the mean value coordinates. We shall use u_3 and $u_5 = \sin(\pi x) \sin(\pi y)$ as testing solutions. Numerical results are listed in Table 9. It takes much longer than Wachspress coordinates to generate the mass and stiffness matrices as we have to use high order Gaussian quadrature to ensure sufficient accuracy. This shows the Wachspress coordinates are better than the mean value coordinates in this situation.

Example 9. In this example, we show polygonal splines of order 3 to solve the Poisson equation. For convenience, we only present numerical results from one testing function u_3 based on pentagonal refinements and quadrangulation refinements as shown in Figure 10. To compare, we also list the results from continuous finite

TABLE 9

Convergence of polygonal spline of order 2 over repeatedly refined pentagon partitions in Figure 8 based on the mean value coordinates.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
76	1.07749006	0.00	7.251159	0.00	2.91	0.003	3.69
392	0.04905039	4.46	1.007859	2.85	4.39	0.003	19.91
2224	0.00364113	3.75	0.203799	2.31	17.73	0.024	119.51
13088	0.00029163	3.64	0.038882	2.39	98.69	0.215	700.64
78016	0.00002405	3.60	0.007281	2.42	591.48	1.406	4309.75
DoF	RMSE (u_5)	Rrate	RMSE (∇u_5)	Rate	Etime	Stime	MKtime
76	0.04149946	0.00	0.339557	0.00	2.98	0.003	3.92
392	0.00230685	4.17	0.055688	2.61	4.41	0.003	19.96
2224	0.00017574	3.71	0.010330	2.43	17.72	0.025	117.79
13088	0.00001446	3.60	0.001929	2.42	98.63	0.186	699.16
78016	0.00000134	3.43	0.000461	2.07	589.98	1.365	4326.21

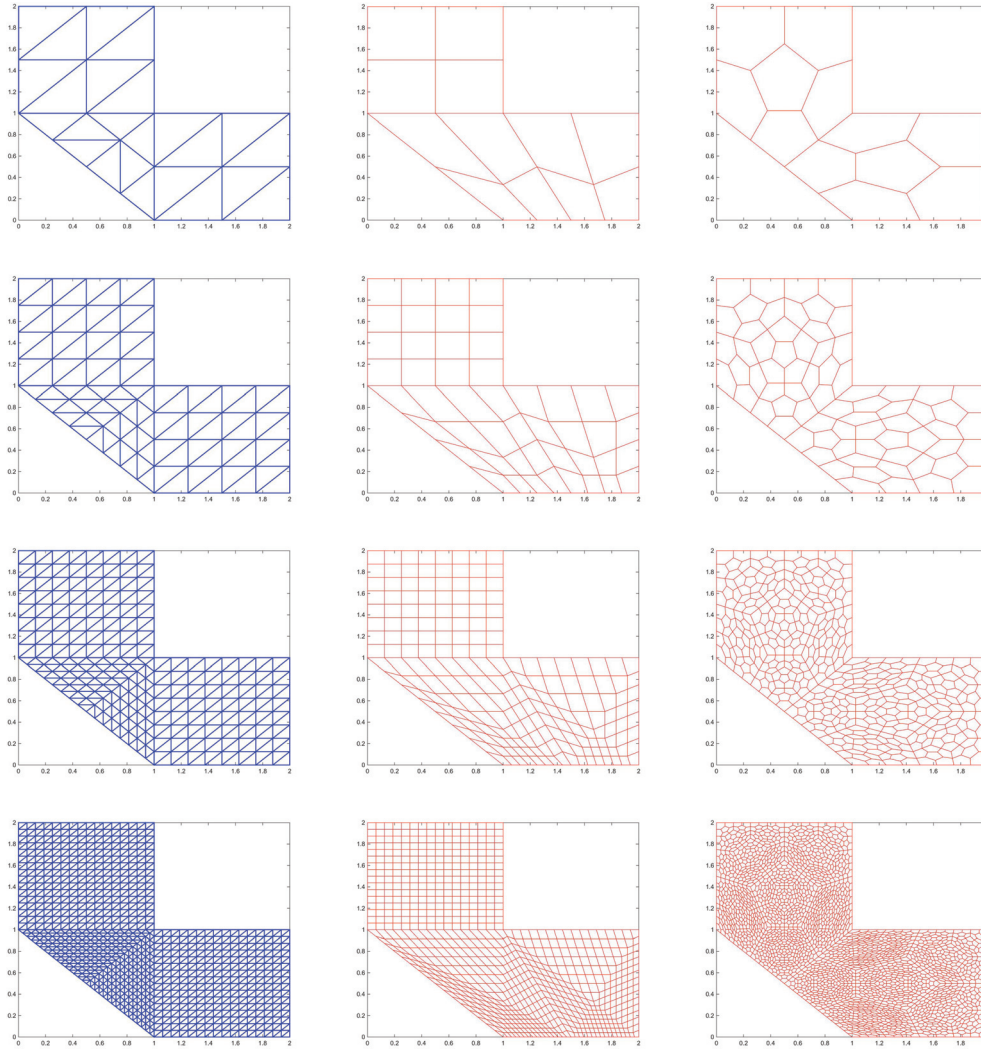


FIG. 10. Three different partitions of an L-domain Ω .

TABLE 10

Convergence of continuous finite elements of degree 3 over repeatedly refined triangulations in Figure 10.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
156	0.38339468	0.00	3.675201	0.00	9.64	0.02	0.18
576	0.01803492	4.41	0.668696	2.46	26.28	0.03	0.31
2208	0.00127945	3.82	0.091991	2.86	48.31	0.10	0.94
8640	0.00008248	3.96	0.011651	2.98	157.05	0.39	4.71
34176	0.00000516	4.00	0.001456	3.00	604.03	1.56	26.97
135936	0.00000032	4.01	0.000182	3.00	2387.03	7.45	100.05

TABLE 11

Convergence of polygonal spline of order 3 over repeatedly refined quadrilateral partitions in Figure 10.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
108	0.31682564		2.440209		2.77	0.001	0.75
384	0.00993331	5.00	0.326953	2.90	3.36	0.003	1.36
1440	0.00061742	4.01	0.038366	3.09	7.15	0.014	5.11
5568	0.00004165	3.89	0.004780	3.00	25.24	0.072	20.71
21888	0.00000266	3.97	0.000600	2.99	97.90	0.352	91.36
86784	0.00000017	3.99	0.000075	3.00	384.36	1.686	498.23

TABLE 12

Convergence of polygonal spline of order 3 over repeatedly refined pentagon partitions in Figure 10.

DoF	RMSE (u_3)	Rate	RMSE (∇u_3)	Rate	Etime	Stime	MKtime
126	0.40725861		3.057028		3.70	0.006	4.04
660	0.00766360	5.73	0.249972	3.61	4.94	0.008	19.01
3768	0.00025659	4.90	0.018809	3.73	18.02	0.052	113.43
22224	0.00001064	4.59	0.001575	3.58	100.49	0.378	697.02
132576	0.00000050	4.42	0.000140	3.49	613.10	2.588	4455.70

elements of degree 3 over refined triangulations. Our computer is able to find numerical solutions based on partitions of the L-domain in Figure 10 which are obtained by using the pentagonal refinement scheme 5 times while using quadrilateral refinement 6 times and triangulation refinement 6 times. From the numerical results in Tables 10, 11, and 12, we can see that polygonal splines can achieve a more accurate solution and gradients using a smaller number of degrees of freedom when using quadrilaterals although assembling the mass and stiffness matrices takes a much longer time than that for triangulation.

6. Conclusions and future research problems. We proposed to define a continuous polygonal spline space $S_d(\Delta)$ of order $d \geq 1$ over a mixed partition of convex polygons with an arbitrary number of sides which possesses a property of reproducing polynomials of degree d . Then we presented a construction of locally supported basis functions for $S_d(\Delta)$ over the mixed partition Δ of convex polygons for $d \geq 2$. For $d = 2$, our construction is simpler than the one in [23]. For $d \geq 3$, our construction is only based on Wachspress coordinates. We then explained how to use these spline functions to solve the Poisson equation numerically over a partition of quadrilaterals or pentagons. In particular, we found a refinement scheme for pentagons and showed that this refinement leads to a higher order of convergence rate.

There are many problems remaining open. For example, when $d \geq 3$, we are only able to construct polygonal splines of order d using Wachspress coordinates. It is interesting to see if one can use other GBCs to construct higher order splines. Also, we would like to know the approximation properties of our interpolation schemes. In particular, is it possible that the polygonal splines studied in this paper have higher approximation power than standard polynomial splines? Our numerical results suggest that the refinement scheme for pentagons may lead to higher accuracy than the uniform refinement of triangulations and quadrangulations. We would like to understand this phenomenon better.

In addition, we would like to extend our study to a higher dimensional setting. Currently we are working on constructing polyhedral splines over a three-dimensional (3D) partition of mixed polyhedra and using them for solving PDEs.

Finally, we would like to construct polygonal splines which are C^1 or smoother in the two-dimensional and 3D settings using GBCs in [12], [11], [26], and so on. These are all under investigation. Of course, we welcome any interested reader to join our efforts to understand polygonal splines better.

REFERENCES

- [1] D. N. ARNOLD AND G. AWANOU, *The serendipity family of finite elements*, Found. Comput. Math., 11 (2011), pp. 337–344.
- [2] G. AWANOU, M.-J. LAI, M. J., AND P. WENSTON, *The Multivariate Spline Method for Numerical Solution of Partial Differential Equations, Wavelets and Splines*, G. Chen and M.-J. Lai, eds., Nashboro Press, pp. 24–74, 2006.
- [3] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Springer Verlag, New York, 1994.
- [4] F. BREZZI AND M. FORTIN, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
- [5] L. BEIRÃO DA VEIGA, K. LIPNIKOV, AND G. MANZINI, *Arbitrary-order nodal mimetic discretizations of elliptic problems on polygonal meshes*, SIAM J. Numer. Anal., 49 (2011), pp. 1737–1760.
- [6] L. BEIRÃO DA VEIGA, F. BREZZI, A. CANGIANI, G. MANZINI, L. D. MARINI, AND A. RUSSO, *Basic principles of virtual element methods*, Math. Models Methods Appl. Sci., 23 (2013), pp. 199–214.
- [7] P. CIARLET, *The Finite Element Method for Elliptic Problems*, North Holland, Amsterdam, 1978.
- [8] G. FARIN, *Triangular Bernstein-Bézier patches*, Comput. Aided Geom. Design, 3 (1986), pp. 83–127.
- [9] M. FLOATER, *Mean value coordinates*, Comput. Aided Geom. Des., 20 (2003), pp. 19–27.
- [10] M. FLOATER, *Generalized barycentric coordinates and applications*, Acta Numer., 24 (2015), pp. 161–214.
- [11] M. FLOATER, K. HORMANN, AND G. KÓS, *A general construction of barycentric coordinates over convex polygons*, Adv. Comput. Math., 24 (2006), pp. 311–331.
- [12] M. S. FLOATER, G. KOS, AND M. REIMERS, *Mean value coordinates in 3D*, Comput. Aided Geom. Design, 22 (2005), pp. 623–631.
- [13] X.-L. HU, D.-F. HAN, AND M.-J. LAI, *Bivariate splines of various degrees for numerical solution of partial differential equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1338–1354.
- [14] M.-J. LAI AND L. L. SCHUMAKER, *Spline Functions on Triangulations*, Cambridge University Press, Cambridge, 2007.
- [15] M.-J. LAI AND L. L. SCHUMAKER, *A domain decomposition method for computing bivariate spline fits of scattered data*, SIAM J. Numer. Anal., 47 (2009), pp. 911–928.
- [16] M.-J. LAI AND G. SLAVOV, *On Recursive Refinement of Convex Polygons*, manuscript.
- [17] M.-J. LAI AND P. WENSTON, *Bivariate splines for fluid flows*, Comput. & Fluids, 33 (2004), pp. 1047–1073.
- [18] C. T. LOOP AND T. D. DEROSE, *A multisided generalization of Bézier surfaces*, ACM Trans. Graph., 8 (1989), pp. 204–234.
- [19] G. MANZINI, A. RUSSO, AND N. SUKUMAR, *New perspectives on polygonal and polyhedral finite element methods*, Math. Models Methods Appl. Sci., 24 (2014), pp. 1665–1699.

- [20] H. PRAUTZSCH, W. BOEHM, AND M. PALUSZNY, *Bézier and b-Spline Techniques*, Springer, New York, 2002.
- [21] L. RAMSHAW, *Blossoms are polar forms*, *Comput. Aided Geom. Design*, 6 (1989), pp. 323–358.
- [22] A. RAND, A. GILLETTE, AND C. BAJAJ, *Interpolation error estimates for mean value coordinates over convex polygons*, *Adv. Comput. Math.*, 39 (2013), pp. 327–347.
- [23] A. RAND, A. GILLETTE, AND C. BAJAJ, *Quadratic serendipity finite elements on polygons using generalized barycentric coordinates*, *Math. Comp.*, 83 (2014), pp. 2691–2716.
- [24] B. SZABÓ AND I. BABUSKA, *Finite Element Analysis*, John Wiley & Sons, New York, 1991.
- [25] E. L. WACHSPRESS, *A Rational Finite Element Basis*, *Math. Sci. Eng.* 114, Academic, New York, 1975.
- [26] E. L. WACHSPRESS, *Barycentric coordinates for polytopes*, *Comput. Math. Appl.*, 61 (2011), pp. 3319–3321.