A COMPARISON OF HIGH ORDER INTERPOLATION NODES FOR THE PYRAMID*

JESSE CHAN† AND T. WARBURTON‡

Abstract. The use of pyramid elements is crucial to the construction of efficient hex-dominant meshes [M. Bergot, G. Cohen, and M. Duruflé, *J. Sci. Comput.*, 42 (2010), pp. 345–381]. For conforming nodal finite element methods with mixed element types, it is advantageous for nodal distributions on the faces of the pyramid to match those on the faces and edges of hexahedra and tetrahedra. We adapt existing procedures for constructing optimized tetrahedral nodal sets for high order interpolation to the pyramid with constrained face nodes, including two generalizations of the explicit warp and blend construction of nodes on the tetrahedron [T. Warburton, *J. Engrg. Math.*, 56 (2006), pp. 247–262]. Comparisons between nodal sets show that the lowest Lebesgue constants are given by warp and blend nodes for order $N \leq 7$ and Fekete nodes for $N > 7$, though numerical experiments show little variation in the conditioning and accuracy of all surveyed nonequidistant points.

Key words. high order, interpolation, finite elements, Lebesgue constant

AMS subject classifications. 65D05, 65D15

DOI. 10.1137/141000105

1. Introduction. In recent years, high order finite element methods have been shown to have significant advantages over low order finite elements in a variety of areas. In particular, for smooth solutions, they tend to converge more rapidly under both order and mesh refinement, and for wave propagation, display less numerical dissipation under time-marching schemes than low order discretizations [12]. High order finite element stiffness and mass matrices are also typically block structured, allowing for efficient local computations in matrix-free applications.

Quadrilateral and hexahedral elements may offer significant benefits over triangular and tetrahedral elements in high order finite element methods as well—exploiting a tensor-product structure allows for simplified data structures, as well as fast, low-memory applications of operators in matrix-free methods. Additionally, high order discontinuous Galerkin (DG) methods benefit from the use of hexahedral elements by reducing the number of flux calculations and also reducing the number of degrees of freedom compared to tetrahedral elements, while still delivering the same global order of approximation. However, while tetrahedral mesh generation has developed to be able to mesh near-arbitrary geometries, it is difficult to mesh arbitrary geometries using purely hexahedral meshes. Often, it is at most possible to construct hex-dominant meshes, which contain primarily hexahedral elements, but also tetrahedral, wedge, and pyramid elements [11, 3, 1].

We are interested in developing H^1-conforming finite element methods using nodal (Lagrange) basis functions. For ease of conformity between elements, it is preferable

Downloaded 05/25/17 to 128.42.187.224. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

to have the same distribution of nodes on each type of element face. The nodal distribution may also be chosen in such a way that it minimizes the interpolation error for the given nodal basis via minimization of the Lebesgue constant. Defining such distributions for quadrilateral and hexahedral elements is typically done through a tensor-product construction, while construction of optimal distributions for triangular and tetrahedral elements is well explored [25, 24, 9, 17, 31, 28]. For the wedge element, a simple tensor-product extrusion of nodes on a triangular face allows for conformity with both tetrahedral and hexahedral elements with a reasonable Lebesgue constant. However, for the pyramid, optimized distributions of nodes have received less attention [10, 5, 3, 14], and are the focus of this article.

2. Finite element spaces and orthogonal bases on pyramids. Since pyramid and wedge elements are used primarily to link tetrahedral, hexahedral, and wedge elements together, it is appropriate to require that the trace space on triangular faces of the pyramid and tetrahedra are the same, and similarly for the quadrilateral faces of the pyramid and hexahedra. Additionally, we require that the edge trace spaces of the pyramid, tetrahedra, and hexahedrea are the same.

Standard trace spaces for tetrahedral and hexahedral finite element spaces contain polynomials of total degree or independent degree N, respectively; however, it has been shown by Bedrossian [2], Wieners [32], and Nigam and Phillips [21] that it is impossible to construct a basis with polynomial trace spaces on a pyramid with purely polynomial functions. Consequentially, conforming finite element spaces for pyramids necessarily contain rational functions.

Nigam and Phillips [21] give a comprehensive construction of arbitrary order H^1, $H(\mathrm{curl})$, $H(\mathrm{div})$, and L^2 finite element spaces for pyramids in terms of a monomial basis. Prior to this, L^2 orthogonal basis functions for the pyramid were used early on in hp and spectral finite element simulations by Warburton, Sherwin, and Karniadakis in [30, 26, 19], and partially orthogonalized high order finite element spaces were used in hp-adaptive conforming finite elements for Maxwell's equations with exact sequence by Zaglmayr [33] and Demkowicz et al. [11].

Bergot, Cohen, and Duruflé present an alternative L^2 orthogonal basis of optimal dimension on the pyramid in [3]. Comparisons are made with existing finite element spaces, and the basis of Warburton, Sherwin, and Karniadakis is shown to be suboptimal for higher order, while the basis of Nigam and Phillips is shown to be optimal, but of larger dimension than necessary.[1] Though the basis itself is rational, its traces are polynomial, satisfying our requirement that the trace spaces match with tetrahedral and hexahedral elements. We will use the basis of Bergot, Cohen, and Duruflé in our construction of nodal basis functions. For a reference pyramid (see Figure 1) with coordinates r, s, t such that

$$r, s \in [t - 1, 1 - t], \quad t \in [0, 1],$$

we define the basis functions $P_{ijk}(r, s, t)$ as

$$P_{ijk}(r, s, t) = P_i^{0,0}\left(\frac{r}{1-t}\right) P_j^{0,0}\left(\frac{s}{1-t}\right)(1-t)^c P_k^{2(c+1),0}(2t - 1),$$

where

$$c = \max(i, j), \quad 0 \le i, j \le N, \quad 0 \le k \le N - c,$$

---

[1]A new basis with smaller dimension is presented by Nigam and Phillips in [22]. The resulting approximation space matches that of Bergot, Cohen, and Duruflé. It should be noted that Zaglmayr's basis spans the same approximation space as well.
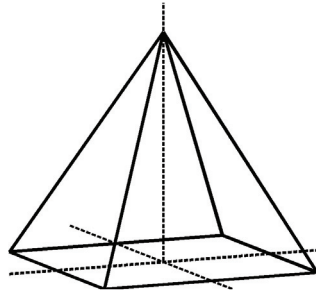
FIG. 1. *Reference pyramid on* $[-1,1]^2 \times [0,1]$.

and $P_n^{a,b}(x)$ is the Jacobi polynomial of order n, orthogonal with respect to the weight $(1-x)^a(1+x)^b$. For a given N, the above procedure produces N_p distinct basis functions, where

$$N_p = \frac{(N+1)(N+2)(2N+3)}{6}.$$

We may order these orthogonal functions arbitrarily as $\phi_j(r,s,t)$ from $j = 1, \ldots, N_p$.

Using the above basis functions, we may define the generalized Vandermonde matrix

$$V_{ij} = \phi_j(r_i, s_i, t_i), \quad i = 1, 2, \ldots, N_p,$$

where (r_i, s_i, t_i) is a set of nodal points contained inside or on the boundary of the reference pyramid. Then, a nodal basis may be constructed using elements of the inverse of the Vandermonde matrix

$$\ell_i(r,s,t) = \sum_{j=1}^{N_p} (V^{-1})_{ij}\phi_j(r,s,t).$$

3. Construction of optimized nodal sets. In the discrete setting, enforcing conformity for nodal finite element methods reduces to the matching of nodal degrees of freedom from one element face to another element face. This motivates the requirement that the distribution of surface nodes for both pyramids and wedges should match the distribution of face nodes on tetrahedra and hexahedra. Furthermore, it is assumed that the nodes on the edges of each element follow a Gauss–Legendre–Lobatto (GLL) distribution.

For quadrilateral and hexahedral elements, the simplest construction is to take a tensor product of one-dimensional (1D) GLL nodes. This choice of nodes is central in the formulation of spectral element methods (SEMs); combined with GLL quadrature and tensor product evaluation of operators, the result is a highly efficient method for tensor product meshes [23, 12, 19]. For wedges, the nodal distribution may also be easily constructed—assuming a given distribution of face nodes for the tetrahedra with GLL nodes on the edges, we can extrude this distribution in the direction orthogonal to the face, which results in a tensor product GLL structure on the quadrilateral face of the wedge.

For pyramids, the ideal nodal distribution is less clear. It is possible to simply combine an arbitrary surface and interior node distribution; for example, both Bergot,

Cohen, and Duruflé [3] and Gassner et al. [14] combine the electrostatic nodes of Hesthaven on the triangular faces of the pyramid with an appropriate number of tensor product GLL nodes in the interior and on the base, similar to the construction of a Stroud quadrature. However, it is possible to choose a more tailored distribution of interior nodes in order to optimize some measure of quality of the nodal set.

3.1. Metrics of quality and optimization strategies. A common aim in the construction of nodal sets is to minimize the Lebesgue constant of the nodal distribution. Given a set of N_p nodal points $\{x_1, \ldots, x_{N_p}\}$, the Lebesgue constant is defined in terms of the Lagrange interpolatory basis functions $\ell_i(\boldsymbol{x})$ as

$$\Lambda = \max_{\boldsymbol{x} \in K} \sum_{i=1}^{N_p} |\ell_i(\boldsymbol{x})|.$$

The Lebesgue constant bounds the interpolation error in the max norm such that

$$\|f - f_N\|_\infty \leq (\Lambda + 1) \|f - f_N^*\|_\infty,$$

where f_N is the order N interpolant of f, f_N^* is the best order N approximation to f, and Λ is the Lebesgue constant for the given interpolation points.

For the tetrahedron, nodal distributions which minimize the Lebesgue constant have been explored in great detail. Chen and Babǔska minimized the L^2 norm of the sum of the Lagrange basis—an L^2 analogue of the Lebesgue constant—in [9]. Hesthaven constructed points using an analogy to electrostatics; a nodal distribution is determined by finding stationary distributions of charges, which are in turn related to the zeros of specific Jacobi polynomials [17].

The warp and blend construction of nodes on triangles and tetrahedra involves defining a warp, or a displacement which shifts equispaced nodes to GLL nodes on an edge, and blending this warping function into the interior of faces and volumes using techniques from curvilinear mesh generation [31, 18]. The subsequent blends are linear in each barycentric coordinate, but an additional scaled quadratic warp term can be introduced in order to further optimize the Lebesgue constant over the resulting distribution of points.

Another metric for quality of a nodal distribution is the determinant of the Vandermonde matrix, and distributions which maximize the determinant of the Vandermonde matrix are referred to as Fekete nodes. The maximization of the determinant is also an attractive alternative to direct optimization of the Lebesgue constant since there exist analytic expressions for the determinant of the Vandermonde matrix, which can be exploited in algorithms to compute Fekete nodes. For example, Taylor, Wingate, and Vincent utilized analytic expressions for the determinant to develop a pseudotime steepest ascent algorithm which can be used to iteratively move from an arbitrary node set to the Fekete node set of a given order [28].

Other work has explored the optimization of interpolation points for more general shapes; Gassner et al. [14] give a general construction for a polygon using a barycentric mapping, while [15, 29, 20] give algorithms for the approximation of good interpolation points on general domains in multiple dimensions. Bos et al. construct nodal points using the concept of weakly admissible meshes (WAMs) [5], which are sequences of subsets A_N of an element over which an upper bound

$$\|p(x)\|_\infty \leq C(A_N)\|p(x)\|_{\infty, A_N} \quad \forall p \in P_N^d$$

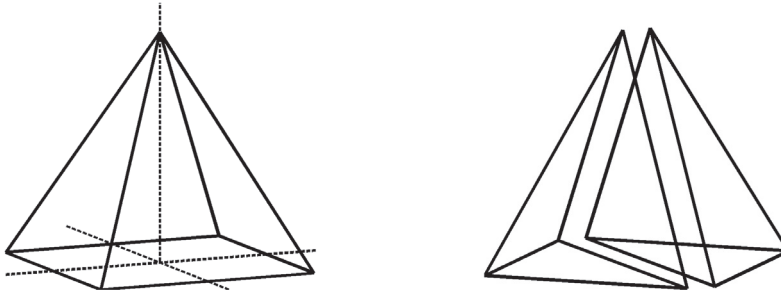can be shown. For Fekete points generated from WAMs, $C(A_N)$ can be used to bound

FIG. 2. *Reference pyramid on $[-1,1]^2 \times [0,1]$ and tetrahedral splitting.*

the Lebesgue constant, though the above bound only holds for polynomials of order N, while conforming finite element spaces on pyramids contain rational functions.

A more recent innovation is the application of numerical linear algebra techniques to compute sets of so-called "approximate Fekete" points [10]. Bos et al. [6] and Sommariva and Vianello [27] utilized similar ideas in the context of numerical linear algebra to compute "approximate Fekete" points. We note that these points do not refer to the approximations of Fekete points detailed by Taylor, Wingate, and Vincent, and instead relate to a discrete equivalent of Fekete points, which use the concept of WAM to characterize "approximate Fekete" and "discrete Leja" points. These points may then be computed using manipulations and factorizations of a Vandermonde matrix.

3.2. Face nodal distributions. For the following experiments, we enforce a fixed distribution of nodes on the faces of the pyramid. For quadrilateral faces, we choose SEM/tensor product GLL nodes for conformity with hexahedral elements. For triangular faces of the pyramid, we fix the face nodes to be identical to those of the warp and blend tetrahedra. As noted above, the choice of nodal distributions for the tetrahedron is less obvious. For consistency in the comparison of different pyramid nodal sets, we fix the triangular faces to be the warp and blend nodes for the tetrahedra, which give competitive Lebesgue constants, but maintain a simple and explicit construction for any order N.

We note that it may be possible to optimize nodal distributions over triangular faces to minimize the Lebesgue constants for tetrahedra, wedge, and pyramid elements simultaneously. The work described here does not consider this option, though we hope to explore this in the future.

3.3. Pyramid nodes by warp and blend. For conformity, we assume that the nodal distribution on both triangular and quadrilateral faces is fixed. For quadrilateral faces, we will choose SEM nodes for conformity with quadrilateral elements, and for triangular faces, we will choose the warp and blend optimized points of Warburton [31].

We propose a simple approach of splitting the pyramid into two tetrahedra, as in Figure 2, on which we construct nodal distributions using the warp and blend procedure. We will refer to this two-tetrahedron representation as a "duplex pyramid" (referring both to the tetrahedron as a three-dimensional (3D) simplex and the definition of "duplex" as having two parts). Supposing that the nodes along the shared face are only counted once, this produces $(N+1)(N+2)(2N+3)/6$ nodes, equal to the dimension N_p of the orthogonal basis.
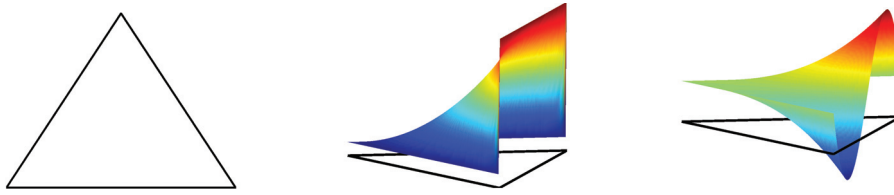
FIG. 3. *Reference equilateral triangle with xy coordinates (left), blending function (middle), and combined warp/blend (right) for a given edge.*

The idea of the duplex pyramid is not new; both Wieners [32] and Bluck and Walker [4] approached the construction of basis functions on the pyramid by dividing the pyramid into two tetrahedra and applying a conformity condition on the interface. The construction of nodes may be approached the same way—we will construct nodal sets by dividing the pyramid into two tetrahedra but constrain them to have the same locations on the shared face. However, unlike the splitting techniques of Wieners and Bluck and Walker, the choice of a nodal distribution is independent of the choice of finite element space, depending only on an orthogonal basis defined over the entire pyramid. To this end, we choose the orthogonal basis of Bergot, Cohen, and Duruflé as described in section 2.

3.3.1. The warp and blend procedure. The warp and blend procedure for a triangle is based on the warping function $w_{1D}(r)$, which maps 1D equidistant points r_{eq} to 1D GLL nodes r_{GLL} via the deformation

$$r_{GLL} = r_{eq} + w_{1D}(r_{eq}).$$

We define $w_{1D}(r)$ directly as the interpolating polynomial of $r_{GLL} - r_{eq}$. We may now displace equidistant nodes on a given triangle edge based on the displacement of GLL nodes over an edge. Consider edge 1; the blend procedure is then to extrapolate the displacement of nodes into the interior of the triangle by defining a blending function of the barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$, with the requirement that the blending function be one on edge 1 and zero on edges 2 and 3. Figure 3 shows the edge 1 blending function $b_1(\lambda_1, \lambda_2, \lambda_3)$.

Noting that $1 - \lambda_1 - \lambda_2 = \lambda_3$, we may write b_1 as

$$b_1(\lambda_1, \lambda_2, \lambda_3) = \frac{4\lambda_1\lambda_2}{(2\lambda_1 + \lambda_3)(2\lambda_2 + \lambda_3)} = \frac{4\lambda_1\lambda_2}{1 - r_1^2},$$

where $r_1 = \lambda_2 - \lambda_1 \in [-1, 1]$ is the 1D coordinate along edge 1.

The denominator of b_1 is singular at points $r = \pm 1$; however, we note that these correspond to the displacement of vertex nodes by $w_{1D}(r)$, which we assume is zero for all nodal sets. This motivates the definition of equivalent nonsingular warping and blending functions $\tilde{b}_1(\lambda_1, \lambda_2, \lambda_3)$ and $\tilde{w}(r)$,

$$\tilde{b}_1 = 4\lambda_1\lambda_2, \qquad \tilde{w}(r) = \begin{cases} \frac{w_{1D}(r)}{1 - r^2}, & |r| < 1, \\ 0, & r = \pm 1. \end{cases}$$

The warp is applied along the edge tangent direction, while the blending carries the warp function into the interior of the triangle along the edge normal direction.
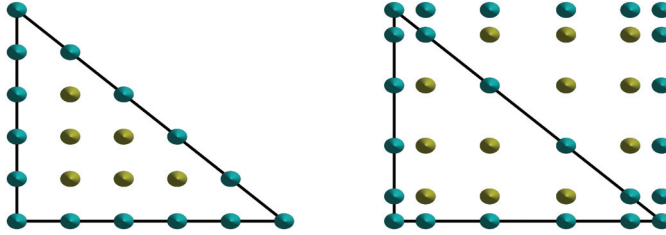
FIG. 4. *Reference triangle in rs coordinates with equispaced (left) and quadrilateral GLL nodes (right) for $N = 5$ overlaid.*

For edge 1, the warp thus affects the x position of the triangle nodes, and the blending function carries this x displacement into the interior of the element along the y direction. This results in the following expression for new nodal positions:

$$\left[\begin{array}{c} x \\ y \end{array} \right] = \left[\begin{array}{c} x_{\mathrm{eq}} \\ y_{\mathrm{eq}} \end{array} \right] + \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \tilde{w}(\lambda_2 - \lambda_1)\tilde{b}_1(\lambda_1, \lambda_2, \lambda_3).$$

The final step of the warp and blend procedure is to parametrize the blending function with a quadratic variation in order to increase the magnitude of the blend in the direction normal to the edge. For edge 1, this may be expressed using the modified blending function $\left(1 + (\alpha\lambda_3)^2\right)\tilde{b}_1$. As α is increased, the amount with which the warping function $\tilde{w}(r)$ is blended towards the opposite vertex increases. Taking the same blending function parameter α over all edges allows for a one-parameter family of nodal distributions, which may then be optimized over α to minimize the Lebesgue constant of the resulting nodal distribution.

For a tetrahedron, since each face is the affine image of an equilateral triangle, we may use the above procedure to define warping and blending functions to displace face nodes. A face blending function may then be used to define displacement formulas for nodes in the interior of the tetrahedron. This blend may also be optimized with some parameter β, which is arbitrarily taken to be the same as the face blend parameter α to retain the one-parameter nature of the optimization. Since this portion of the warp and blend procedure does not change for our extrapolations to the pyramid, we omit the details for brevity and refer the reader to [31, 18].

3.3.2. A duplex warp and blend procedure. To adapt the warp and blend procedure to the duplex pyramid, we redefine the warping and blending functions for the faces of each tetrahedron corresponding to the square base of the pyramid. For the rs reference right triangle with coordinates

$$r, s \in [-1, 1], \quad r + s \leq 1,$$

we require the warp and blend procedure to map equispaced nodes on a triangle to match half of the GLL nodes on a quadrilateral, as shown in Figure 4. We note that, since the GLL nodes are a tensor product in r and s coordinates, we may directly define the face warping function as a tensor product $w_{1\mathrm{D}}(r)w_{1\mathrm{D}}(s)$ as well. In addition, we assume that the tetrahedral face embedded in the base of the pyramid corresponds to a plane where t is constant, implying that the warp for the square face in the t coordinate is zero.

Let x, y, z denote coordinates on the equilateral tetrahedron, and let the number of nodes on a tetrahedron be denoted by $N_{p,T}$. The resulting warp and blend procedure
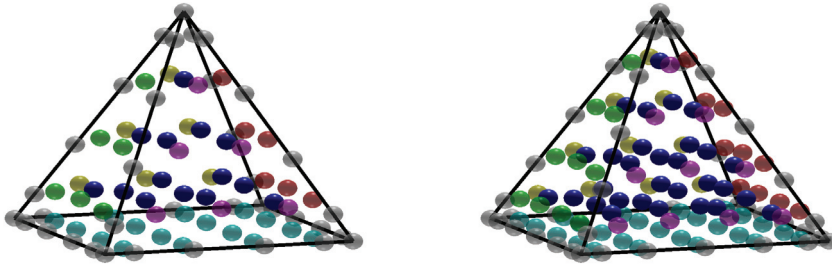
FIG. 5. *Duplex pyramid warp and blend nodes for $N = 5$ (left) and $N = 6$ (right) nodes for the pyramid.*

for a tetrahedral half of a pyramid is given as follows: for faces corresponding to the triangular faces of the pyramid, the warp and blend procedure is identical to that of the regular tetrahedron. For the face corresponding to the square base of the pyramid, we define a warp which maps the nodes to half of a GLL distribution on the right triangle, which is then blended into the interior as usual. The resulting nodes are then mapped to the two tetrahedral halves of the $[-1, 1]^2 \times [0, 1]$ reference pyramid using an affine transformation, such that the nodes on the base of each tetrahedron align with tensor product GLL nodes on the square base of the pyramid, and the nodes on the shared face of the two tetrahedra match. This procedure is given in more detail in Algorithm 1, and the nodal distribution for $N = 5, 6$ is shown in Figure 5.

ALGORITHM 1. WARP AND BLEND PROCEDURE ON ONE-HALF OF THE DUPLEX PYRAMID.

1: **procedure** DUPLEX WARP AND BLEND
2:     Initialize $\{x_i, y_i, z_i\}_{i=1}^{N_{p,T}}$ to equispaced nodes on the tetrahedron.
3:     **for** faces of the tetrahedron **do**
4:         **if** face is not the square face **then**
5:             Define face warp $w(x, y, z)$ as for the tetrahedron.
6:         **if** face is the square face **then**
7:             Define $w(x, y, z) = w_{1D}(r(x, y))w_{1D}(s(x, y))$.
8:         Blend face warp into interior.
9:         Evaluate blended warp, apply shifts to $\{x_i, y_i, z_i\}_{i=1}^{N_{p,T}}$.
10:     Map $\{x_i, y_i, z_i\}_{i=1}^{N_{p,T}}$ to $\{r_i, s_i, t_i\}_{i=1}^{2N_{p,T}}$ on each half of the duplex pyramid.
11:     Remove the $(N + 1)(N + 2)/2$ redundant nodes on the shared face.
12:     **return:** points $\{r_i, s_i, t_i\}_{i=1}^{N_p}$.

The warp and blend nodes for the triangle and tetrahedron were optimized over a parameter α, which controls the quadratic variation of the blending function. We adopt $\alpha = \alpha_{\text{opt}}$, the optimized value for the tetrahedron given in [31, 18].

The above procedure produces a distribution of nodes for the pyramid which reduces to optimized tetrahedral warp and blend nodes on triangular faces and hexahedral GLL nodes on the quadrilateral base, but is not rotationally symmetric in the interior of the pyramid. We will refer to these nodes as "duplex" in the numerical results.

3.4. An interpolatory warp and blend procedure. The duplex pyramid construction of nodes, while a viable procedure for determining a nodal distribution on the pyramid, is less elegant and more complicated than the original warp and blend

construction on the tetrahedron. However, it is possible to generalize the warp and blend procedure directly to the pyramid in another manner. We first illustrate the procedure on the triangle.

In one dimension, the warping function $w_{1D}(r)$ is constructed as the interpolating polynomial of $r_{GLL} - r_{eq}$, and represents a map from equispaced nodes to the difference between the GLL nodes and equispaced nodes. The extension to the two-dimensional (2D) triangle is discussed in section 3.3.1. The warping function for each individual edge is blended into the triangle in order to determine the displacement of nodes in the interior. The total displacement of the interior nodes may be determined by accumulating the displacements from the warping of each edge.

In one dimension one may also define a direct map from equispaced nodal positions to GLL nodal positions $r_{GLL} = m_{1D}(r_{eq})$, where $m_{1D}(r) = r + w_{1D}(r)$ is the interpolating polynomial for the positions of the GLL nodes. We may also define linear vertex shape functions in r, s coordinates:

$$v_1(r,s) = -\frac{r+s}{2}, \quad v_2(r,s) = \frac{1+s}{2}, \quad v_3(r,s) = \frac{1+r}{2}.$$

Since v_1, v_2, v_3 are identical to the barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$, we may equivalently use vertex shape functions in lieu of barycentric coordinates in defining a linear blend of $m_{1D}(r)$ into the interior of the triangle. This will be useful in generalizing warp and blend to domains without barycentric coordinates.

Note that for the original warp and blend procedure, the blending for each edge is the product of two barycentric variables, which are each linear functions in the local coordinates r, s. As a result, the total displacement of the nodes in each coordinate r and s is the product of linear polynomials and an order N polynomial on the face. This may be exploited for an interpolatory warp and blend procedure based on the above properties of the map.

We define the edge basis functions for $j = 0, \ldots, N - 2$ using 1D Legendre polynomials $L_j(r)$:

$$e_{1,j}(r,s) = v_1(r,s)v_2(r,s)L_j(\xi_1), \quad \xi_1 = v_1(r,s) - v_2(r,s),$$
$$e_{2,j}(r,s) = v_2(r,s)v_3(r,s)L_j(\xi_2), \quad \xi_2 = v_2(r,s) - v_3(r,s),$$
$$e_{3,j}(r,s) = v_3(r,s)v_1(r,s)L_j(\xi_3), \quad \xi_3 = v_3(r,s) - v_1(r,s).$$

These edge basis functions may be used alongside vertex basis functions and interior (bubble) basis functions that vanish along the boundary to form a hierarchical basis defined over an element, and are commonly used in hp-adaptive finite element methods [11]. Here, we will discard interior bubble functions, and construct a basis consisting of only vertex and edge shape functions over the surface of the triangle.

Since the edge basis has cardinality $3N$, equal to the number of nodes on the surface of the triangle, we may use the above basis to interpolate a preset distribution of nodes on the boundary. The evaluation of these interpolants at equispaced nodes then determines the position of the new warp and blend nodes.

For the triangle, we wish to enforce a GLL nodal distribution over the edge for conformity with quadrilateral elements. Let $\{\phi_j(r,s)\}_{j=1}^{3N}$ be the basis consisting of vertex and edge shape functions. Using the coordinates of the $3N$ GLL points (vertex and edge nodes) over the surface of the triangle, we build a $3N \times 3N$ Vandermonde matrix, which can be used to interpolate, at equispaced points on the triangle surface, the positions of GLL nodes on the edges. This process explicitly constructs maps from equispaced nodal coordinates to GLL nodal coordinates on each edge, which are
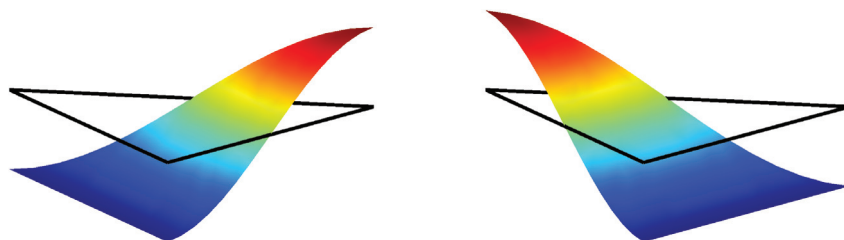
Fig. 6. *Interpolated maps from equispaced to warp and blend nodes for both r (left) and s (right) coordinates on the equilateral triangle.*

blended linearly into the interior. Interior nodal distributions are then determined as direct evaluations of this map (shown in Figure 6) at equispaced coordinates.

If we include an additional quadratic blending into the edge basis functions

$$e_{1,j}(r,s) = \left(1 + (\alpha v_3)^2\right) v_1(r,s)v_2(r,s)L_{j-1}(\xi_1), \quad \xi_1 = v_1(r,s) - v_2(r,s),$$

and similarly for $e_{2,j}, e_{3,j}$, then this process becomes identical to the original warp and blend procedure. In particular, if we set $\alpha = \alpha_{\mathrm{opt}}$, the optimized value reported for the triangle in [31], we recover exactly (to machine precision) the optimized warp and blend nodes on the triangle.

3.5. An interpolatory warp and blend procedure for the pyramid. Using the reference pyramid $[-1,1]^2 \times [0,1]$, we may define the vertex shape functions of Bedrosian [2]

$$v_1(r,s,t) = \frac{1}{4}\left(1 - r - s - t + \frac{rs}{1-t}\right), \quad v_2(r,s,t) = \frac{1}{4}\left(1 + r - s - t - \frac{rs}{1-t}\right),$$

$$v_3(r,s,t) = \frac{1}{4}\left(1 + r + s - t + \frac{rs}{1-t}\right), \quad v_4(r,s,t) = \frac{1}{4}\left(1 - r + s - t - \frac{rs}{1-t}\right),$$

$$v_5(r,s,t) = t.$$

Each vertex function vanishes at the other four vertices, and the traces of these shape functions are linear, though the functions themselves are rational. We may use these vertex functions to generalize the interpolatory warp and blend procedure to the pyramid. Mimicking the procedure for the triangle, we define a hierarchical basis on both edges and faces of the pyramid. The edge functions may be defined in a similar manner to the triangle; for an edge between vertices a and b, we may define $N-1$ edge functions

$$e_{ab,j}(r,s,t) = v_a(r,s,t)v_b(r,s,t)L_j(\xi_{ab}), \quad j = 0, \ldots, N-2,$$

where $L_j(\xi)$ is again the jth order 1D Legendre basis function and $\xi_{ab} = v_a(r,s,t) - v_b(r,s,t)$ is the local coordinate along the edge.

We may similarly define triangular face functions in terms of vertex shape functions due to the linearity of their traces. For a triangular face defined by vertices a, b, and c, we may define barycentric coordinates over the face in terms of vertex shape functions

$$\lambda_1(r,s,t) = v_a(r,s,t), \quad \lambda_2(r,s,t) = v_b(r,s,t), \quad \lambda_3(r,s,t) = v_c(r,s,t).$$

These may then be used to evaluate the orthogonal Dubiner basis on the triangle [13]. Let $D_j(\lambda_1, \lambda_2, \lambda_3)$ denote the jth Dubiner polynomial as a function of the barycentric coordinates; for $j = 0, \ldots, (N-1)(N-2)/2$, we may then define the triangular face functions through

$$f_{\text{tri},j}(r, s, t) = v_a(r, s, t)v_b(r, s, t)v_c(r, s, t)D_j(v_1, v_2, v_3).$$

For $0 \le j, k \le N-1$, the pyramid square face functions may also be defined using the vertex functions at the base and tensor products of 1D Legendre polynomials:

$$f_{\text{quad},jk}(r, s, t) = v_a(r, s, t)v_b(r, s, t)v_c(r, s, t)v_d(r, s, t)L_j(r)L_k(s).$$

Since both the number of surface nodes and total number of vertex, edge, and face basis functions are $N_p = 3N^2 + 2$, we may define the basis $\{\phi_j(r, s, t)\}_{j=1}^{N_p}$ as the collection of vertex, edge, and face functions and construct a square Vandermonde matrix V over the surface nodes.[2]

The construction of the map from equispaced nodes to warp and blend nodes on the face may then be expressed in coefficients of the basis $\{\phi_j(r, s, t)\}$. Assuming that the pyramid surface nodes are a combination of warp and blend nodes on triangular faces and tensor product GLL nodes on the quadrilateral faces, we may solve for the interpolant of these surface nodal values at equispaced nodes on the surface of the pyramid. These interpolants are regarded as maps whose evaluation at equispaced points on the surface and interior of the pyramid determine the position of the interpolatory warp and blend nodes. The procedure for computing these nodes is given in Algorithm 2, and we will refer to these nodes as "IWB" in the numerical results.

ALGORITHM 2. INTERPOLATORY WARP AND BLEND PROCEDURE FOR THE PYRAMID.

1: **procedure** INTERPOLATORY WARP AND BLEND
2:     Initialize positions of equispaced nodes on the pyramid $r_i^{\text{eq}}, s_i^{\text{eq}}, t_i^{\text{eq}}$.
3:     Select $3N^2 + 2$ target node positions on the faces of the tetrahedron.
4:     Define the vertex, edge, and face basis functions $\{\phi_j(r, s, t)\}_{j=1}^{3N^2+2}$.
5:     Using ϕ_j, compute interpolants $m_r(r, s, t), m_s(r, s, t), m_t(r, s, t)$ of the target nodal positions.
6:     Evaluate $r_i = m_r(r_i^{\text{eq}}, s_i^{\text{eq}}, t_i^{\text{eq}}), \quad s_i = m_s(r_i^{\text{eq}}, s_i^{\text{eq}}, t_i^{\text{eq}}), \quad t_i = m_t(r_i^{\text{eq}}, s_i^{\text{eq}}, t_i^{\text{eq}}).$
7:     **return:** points $\{r_i, s_i, t_i\}_{i=1}^{N_p}$.

While this interpolatory procedure is identical to the original warp and blend procedure in two dimensions, they contain differences in three dimensions. The original warp and blend procedure modified node positions edge by edge: for a given triangular face, the warp from each edge of the triangle is blended into the face. Since this warp represents the displacement needed to move an equispaced node to a warp and blend node, these warps may be applied edge-by-edge, updating the positions of nodes on the face one edge at a time. For a given edge on a given face, this also defines a face warp, which is then blended into the interior via a face blending function which vanishes on all other faces. Conceptually, the warping of interior nodes is related to the warping of edge nodes only indirectly (through the face warp)—edge and interior nodes are decoupled from each other.

---

[2]We have constructed the above basis to closely mimic the blending functions used in the original warp and blend procedure. However, we note that any choice of hierarchical basis defined over vertices, edges, and faces—for example, the H^1-conforming basis described in Nigam and Phillips [21], Bergot, Cohen, and Duruflé [3], or [19, 11]—could also be used to construct maps from equispaced to warp and blend nodes.
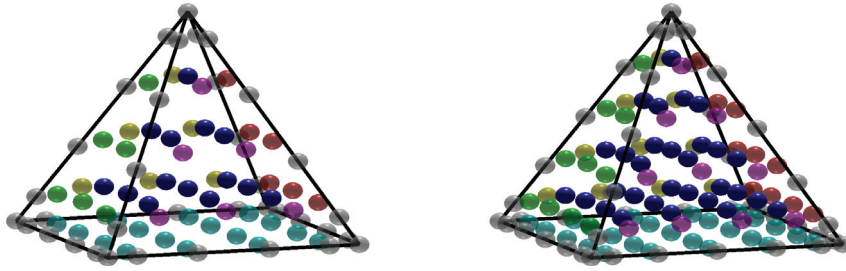
FIG. 7. *Interpolatory warp and blend nodes for $N = 5$ (left) and $N = 6$ (right) nodes for the pyramid.*

In contrast, in the interpolatory warp and blend procedure, the warping of edge nodes directly affects the interior node distribution, due to the fact that each edge function is blended into using one or two vertex shape functions, which are also nonzero in the interior of the pyramid. Thus, both face and edge nodes are coupled together in determining the position of interior nodes of the pyramid. The nodal distribution under this procedure is shown in Figure 7 for $N = 5, 6$.

3.6. Fekete points. Fekete points may be approximately constructed on the pyramid using the steepest ascent procedure from Taylor, Wingate, and Vincent, where nodal positions r_i, s_i, t_i are taken to be stationary distributions of the ODEs

$$\frac{\partial r_i}{\partial t} = \frac{\partial \ell_i}{\partial r}, \qquad \frac{\partial s_i}{\partial t} = \frac{\partial \ell_i}{\partial s}, \qquad \frac{\partial t_i}{\partial t} = \frac{\partial \ell_i}{\partial t}.$$

The steepest ascent procedure in Algorithm 3 is known to be sensitive to initial conditions [28], so we initialize the nodal positions at time $t = 0$ to a reasonable initial distribution by using the warp and blend points defined in section 3.3.1. We solve the resulting system of ODEs using fourth order Runge-Kutta, and terminate the steepest ascent procedure when the maximum change over all nodes in their r, s, or t positions is less than $10^{-10}/dt$ over a time step. Additionally, we do not update surface node positions, which fixes face distributions for conformity with other elements, which is done by setting the velocities $\frac{\partial r_i}{\partial t}$, $\frac{\partial s_i}{\partial t}$, and $\frac{\partial t_i}{\partial t} = 0$ for all nodes on the pyramid faces. The resulting nodes are referred to as "Fekete" in the numerical results, and the nodal distributions for $N = 5, 6$ are shown in Figure 8.

ALGORITHM 3. ODE STEEPEST ASCENT METHOD FOR APPROXIMATION OF FEKETE NODES [28].

1: **procedure** ODE STEEPEST ASCENT FEKETE
2:    Initialize positions of points r_i, s_i, t_i.
3:    Initialize $t = 0$, time step dt, and tolerance *tol*.
4:    **while** $t > 0$ and $\max_i \left|\frac{\partial r_i}{\partial t}\right| > tol$, $\max_i \left|\frac{\partial s_i}{\partial t}\right| > tol$, $\max_i \left|\frac{\partial t_i}{\partial t}\right| > tol$ **do**
5:        Compute, at current positions r_i, s_i, t_i,

$$\frac{\partial r_i}{\partial t} = \frac{\partial \ell_i}{\partial r}, \quad \frac{\partial s_i}{\partial t} = \frac{\partial \ell_i}{\partial s}, \quad \frac{\partial t_i}{\partial t} = \frac{\partial \ell_i}{\partial t}.$$

6:        Time march using LSERK-4 to determine new positions r_i, s_i, t_i.
7:    **return:** r_i, s_i, t_i.

While the Fekete nodes tend to deliver the lowest Lebesgue constant for large N, they become more computationally challenging to determine as N increases. In
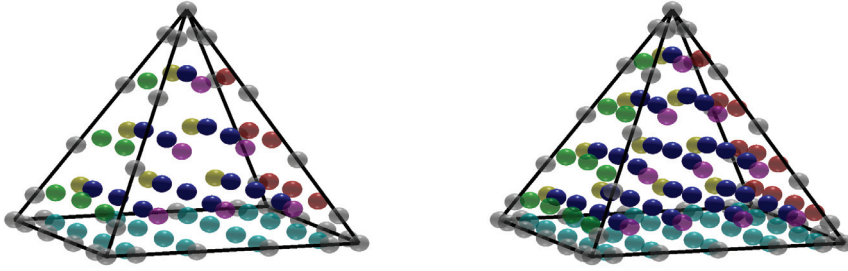
FIG. 8. *Fekete nodes for $N = 5$ (left) and $N = 6$ (right) nodes for the pyramid. Surface nodes (which have fixed nodal distributions) are shown as transparent spheres.*

contrast, apart from an optional 1D optimization of the nodal distribution, both the duplex and interpolatory warp and blend procedures give explicit (noniterative) constructions of pyramid nodes for any degree N.

3.7. "Approximate Fekete" points. Procedures to compute "approximate Fekete" points are given in [6, 27]. An advantage of these algorithms apart from their relatively fast speed is their flexibility; since the algorithms are built around tools from numerical linear algebra, they are generalizable to a larger variety of domains and bases. Most of these algorithms require only evaluations of some linearly independent basis at arbitrary points with which to generate a Vandermonde matrix, and are directly applicable to the pyramid. We consider two procedures, both of which may be used to determine a distribution of interpolation points lying inside the pyramid.

The first of these procedures is a greedy algorithm for maximizing the determinant of the Vandermonde matrix. Supposing that K is the pyramid, given a set of N_s sample points $\{\boldsymbol{x}_i\}_{i=1}^{N_s} \in K$, the following greedy algorithm chooses N_p indices i_k such that $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{N_p}}$ maximize the determinant of the Vandermonde matrix. The algorithm selects these points sequentially based on the column with maximum norm. The matrix is then updated by projecting out the component of the maximum norm column from all other columns. The procedure is detailed in Algorithm 4.

ALGORITHM 4. GREEDY SELECTION OF "APPROXIMATE FEKETE" POINTS FROM A SAMPLE SET [27].

1: **procedure** GREEDY "APPROXIMATE FEKETE"
2:      Select points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_s} \in K$.
3:      Construct $V_{ij} = \phi_j(\boldsymbol{x}_i)$, normalize the columns.
4:      **for** $k = 1, \ldots, N_p$ **do**
5:          Choose $i_k = \operatorname{argmax}_i \|V(:,i)\|$.
6:          **for** $j \neq i_k$ **do**
7:              Orthogonalize $V(:,j)$ with respect to $V(:,i_k)$.
8:      **return:** $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{N_p}}$.

Since we are interested in enforcing conformity between elements, we may introduce small modifications to fix the nodal distribution on pyramid faces. Suppose that we wish to force inclusion of the first N_b points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_b}$ in the index set $\{i_k\}_{k=1}^{N_p}$; we may then skip step 1 of the above algorithm for $k \leq N_b$. For the pyramid, we may force the inclusion of predetermined face nodes into the "approximate Fekete" set using this modification. We implement the above algorithm to determine "approximate

Fekete" nodes on the pyramid with constrained face distributions, and refer to the resulting points as "Greedy" in the numerical results.

The second procedure we consider is an iterative refinement method to determine "approximate Fekete" nodes based on the QR decomposition proposed by Sommariva and Vianello. Suppose that N_s denotes the number of sample points and N_p the dimension of an arbitrary basis $\{\phi_1, \ldots, \phi_{N_p}\}$, initialize V_0 as the $N_s \times N_p$ Vandermonde matrix, and P_0 as the $N_p \times N_p$ identity matrix. Algorithm 5 details this process.

ALGORITHM 5. ITERATIVE REFINEMENT SELECTION OF "APPROXIMATE FEKETE" POINTS FROM A SAMPLE SET [27].

1: **procedure** ITERATIVE REFINEMENT "APPROXIMATE FEKETE"
2:     Select points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_s} \in K$, construct $V_{ij} = \phi_j(\boldsymbol{x}_i)$.
3:     Initialize $V_0 = V$, $P_0 = I$.
4:     **for** $k = 0, \ldots, s-1$ **do**
5:         Compute the QR decomposition of $V_k = Q_k R_k$ and set

$$V_{k+1} = V_k R_k^{-1}, \quad P_{k+1} = P_k R_k^{-1}.$$

6:     Select $m_j \neq 0$.
7:     Set $\mu = (P_s^T)^{-1}m, \quad w = (V_s^T)^{-1}\mu$.
8:     Select N_p points i_k such that $w_{i_k} \neq 0$.
9:     **return:** $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{N_p}}$.

The iterative refinement method has the advantage of being broadly applicable; the basis may be arbitrary, and the initial Vandermonde matrix V_0 may be low rank, since the construction of V_s in the iterative refinement step will produce a nonsingular matrix.

If we wish to enforce a fixed distribution of nodes on the faces of the pyramid, we may modify the basis in the above algorithm to be zero over all nodes on the faces. We do so by taking the underdetermined Vandermonde matrix V_b,

$$V_{b,ij} = \phi_j(x_i), \quad j = 1, \ldots, N_p, \quad i = 1, \ldots, N_b,$$

where N_p is the dimension of the basis, $N_b < N_p$ is the number of points on the surface/boundary of the pyramid. We may then take the $N_p - N_b$ linearly dependent combinations of columns of this matrix corresponding to coefficients which define functions that are zero on the faces—in other words, an interior basis.[3]

Using V_i in lieu of V_0 in the above algorithm allows us to compute "approximate Fekete" points only in the interior of the pyramid. We refer to these resulting points as "QR" in the numerical results.

For both "approximate Fekete" procedures, we take the sample set for a given N to be Stroud-style equispaced points in the pyramid with $N^2 + 1$ points per edge. This choice is motivated by a theorem of Bos and Levenberg which states that, in one dimension, "approximate Fekete" points have the same asymptotic distribution as those of the true Fekete points for this choice of sample points [6]. Further increasing the number of sampling points was not significantly correlated with an improvement in the Lebesgue constant of the resulting "approximate Fekete" point set.

---

[3]Constructing V_i from a basis consisting purely of bubble functions supported only in the interior of the pyramid is another possibility.
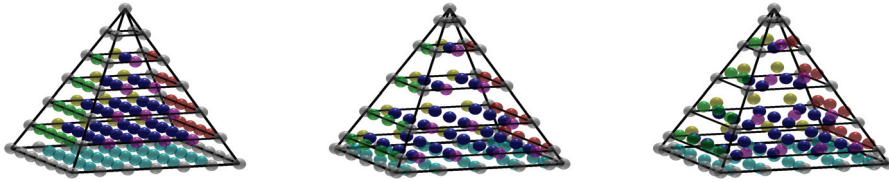
FIG. 9. *Equispaced (left), Stroud-type GLL (middle), and mixed warp and blend/GLL (right) pyramid nodes for $N = 6$.*

TABLE 1
Legend of abbreviations and summaries for different nodal sets.

Equi	Equispaced nodes (see Figure 9)
Conical	Stroud-type (see Figure 9)
Face	Tet face nodes, GLL interior nodes (see Figure 9, [3, 14])
Fekete	Steepest ascent ODE-based (see Figure 8, [28])
Greedy	Choosing from sampled points [27, 7]
QR	Iterative refinement using QR [27, 7]
Duplex	Two-tet warp and blend (see section 3.3.2)
IWB	Interpolatory warp and blend (see section 3.4)

4. Numerical experiments and comparisons with existing nodal sets.
In this section, we compare Fekete, "approximate Fekete," and warp and blend nodes
with other unoptimized nodal sets for the pyramid. We examine four metrics: the
Lebesgue constant, the determinant of the Vandermonde matrix, the condition num-
ber of the Vandermonde matrix, and interpolation error for two specific functions. In
all cases, the Vandermonde matrix is normalized by the L^2 norm of the corresponding
basis function, computed using quadrature [8].

Since there is no closed form expression through which to explicitly compute the
Lebesgue constant for a nodal set, we adaptively sample the Lebesgue function using
a random search [31]

$$L(\boldsymbol{x}) = \sum_{i=1}^{N_p} |\ell_i(\boldsymbol{x})|$$

and seek the Lebesgue constant as the maximum value $\Lambda = \max_{\boldsymbol{x} \in K} L(\boldsymbol{x})$.

The baseline comparison is with equispaced nodes on the pyramid; these can be
defined level by level, similar to the manner in which Stroud conical quadrature rules
are constructed [16]. Similarly, we may construct a Stroud-type conical GLL node
set by levels as well—the levels are placed according to a GLL distribution, and on
each level, nodes are arranged as a tensor product of GLL nodes. Both are shown in
Figure 9 for $N = 6$, and are referred to as "Equi" and "Conical" in the numerical
results.

Both Bergot, Cohen, and Duruflé [3] and Gassner et al. [14] used electrostatic
nodes on the faces and a Stroud-type GLL nodal distribution in the interior (shown
in Figure 9). For consistency in comparison with our optimized nodal sets, we will
mimic their choice of GLL interior nodes but substitute warp and blend nodes from
the tetrahedron for the electrostatic nodes on the faces. We refer to this distribution
as "Face" in the numerical results.

4.1. Lebesgue constants. We include Table 1, which describes references and
summaries of the various surveyed nodal sets. Table 2 gives Lebesgue constants for
different nodal distributions with $N = 3, \ldots, 10$. Since the nodal distribution for

TABLE 2

Values of the Lebesgue constant Λ for various nodal sets and polynomial orders N. The smallest Lebesgue constants for N ≥ 3 are in bold.

N	Equi	Conical	Face	Fekete	Greedy	QR	Duplex	IWB
3	3.15	2.83	**2.72**	2.73	2.80	2.80	2.73	2.75
4	5.94	4.29	4.22	4.13	4.19	4.19	**3.80**	3.90
5	11.87	6.84	6.93	5.53	6.33	6.03	**5.06**	5.11
6	25.13	10.10	10.67	7.35	8.51	8.29	**6.66**	7.23
7	56.66	14.20	15.42	9.71	12.82	13.63	**9.65**	9.75
8	136.40	20.43	22.16	**12.79**	18.85	21.43	14.65	14.22
9	350.23	31.14	34.28	**17.16**	22.84	33.31	23.39	20.82
10	954.08	48.38	54.27	**25.50**	42.85	37.23	40.12	32.16

TABLE 3

Condition numbers of the normalized Vandermonde matrix for various nodal sets and polynomial orders N.

N	Equi	Conical	Face	Fekete	Greedy	QR	Duplex	IWB
3	15.84	16.43	16.34	16.15	16.96	16.96	16.15	16.37
4	22.15	20.57	20.57	20.82	21.91	21.91	20.10	20.63
5	34.79	29.69	29.64	27.85	34.41	31.82	26.53	28.74
6	60.84	38.26	38.48	40.61	48.40	43.17	36.69	43.53
7	123.46	53.02	53.77	63.91	51.75	76.87	52.20	67.54
8	301.65	80.85	83.67	107.05	82.14	114.65	78.92	110.58
9	810.06	131.13	135.18	188.56	105.39	191.26	124.46	187.81
10	2346.19	222.97	234.38	345.23	226.88	205.85	202.36	330.31

$N = 1, 2$ is the same for all distributions, we focus on the lowest Lebesgue constants for $N > 2$, which are bolded for reference.

For $N = 3$, the face nodes return the lowest Lebesgue constant by .01. The duplex pyramid nodes contain the lowest Lebesgue constants for $3 < N \leq 6$, while similarly to the triangle, Fekete nodes outperform other nodal sets for high N. While this happens for $N > 10$ on the triangle, it occurs earlier at $N > 7$ on the pyramid.

An interesting observation is that the face distribution, which chooses face and interior nodal distributions independently of each other, results in a Lebesgue constant which grows faster for large N than the Lebesgue constant for the conical GLL distribution, implying that the interior distribution of nodes plays a significant role in minimizing the Lebesgue constant.

4.2. Determinant and conditioning of the Vandermonde matrix. In this section, we discuss the magnitude of determinants and condition numbers of the Vandermonde matrix for various nodal sets.

Since the determinants of the normalized Vandermonde matrix are too large to represent numerically, we arbitrarily scaled the Vandermonde matrix to prevent numerical overflow. As expected, the largest magnitude determinants of the Vandermonde matrix are produced by the Fekete points. The "approximate Fekete" points do remarkably well, producing determinants that are within a factor of the magnitude of the Fekete determinant. The determinants of all other sets behave roughly the same, decreasing at a steady rate as N increases.

We also compare the condition numbers of the normalized Vandermonde matrix for different nodal sets at various N in Table 3. Since nodal basis functions are typically constructed through the inversion of a Vandermonde matrix, poor conditioning can result in the loss of accuracy when constructing interpolants.

TABLE 4
Interpolation errors for f_1 (see (4.1)).

N	Equi	Conical	Face	Fekete	Greedy	QR	Duplex	IWB
4	3.2e-2	2.17e-2	2.13e-2	2.11e-2	2.11e-2	2.17e-2	2.14e-2	2.20-2
5	8.6e-3	4.95e-3	4.93e-3	5.03e-3	4.86e-3	4.98e-3	4.64e-3	4.70e-3
6	1.5e-3	6.55e-4	6.46e-4	6.53e-4	6.25e-4	6.05e-4	6.12e-4	6.51e-4
7	2.2e-4	8.27e-5	6.99e-5	7.64e-5	7.16e-5	7.12e-5	7.56e-5	6.93e-5
8	3.5e-5	1.27e-5	1.04e-5	1.05e-5	1.06e-5	1.02e-5	9.61e-6	1.05e-5
9	3.1e-6	8.32e-7	6.82e-7	4.55e-7	6.85e-7	6.65e-7	6.45e-7	5.99e-7
10	4.8e-7	1.69e-7	1.18e-7	9.23e-8	1.31e-7	1.31e-7	1.27e-7	1.26e-7

TABLE 5
Interpolation errors for f_2 (see (4.2).

N	Equi	Conical	Face	Fekete	Greedy	QR	Duplex	IWB
4	5.2e-3	5.95e-3	6.04e-3	4.86e-3	4.77e-3	4.77e-3	3.91e-3	3.58e-3
5	3.8e-3	5.82e-3	5.82e-3	5.82e-3	5.82e-3	5.82e-3	5.82e-3	5.82e-3
6	9.0e-4	6.65e-4	6.55e-4	5.90e-4	5.46e-4	5.51e-4	5.13e-4	4.70e-4
7	5.9e-4	5.84e-4	5.84e-4	5.84e-4	7.17e-4	5.84e-4	5.84e-4	5.84e-4
8	1.7e-4	7.34e-5	7.41e-5	6.36e-5	5.87e-5	7.78e-5	5.60e-5	4.38e-5
9	1.2e-4	5.86e-5	5.86e-5	5.86e-5	5.86e-5	5.86e-5	5.86e-5	5.86e-5
10	3.4e-5	8.22e-6	7.98e-6	6.75e-6	8.76e-6	6.68e-6	6.41e-6	5.36e-6

Overall, the duplex nodes produce the most well-conditioned matrices, followed closely by the two "approximate Fekete" node sets. The Fekete and interpolatory warp and blend nodes surprisingly do the most poorly apart from equispaced nodes; however, even in these cases, the condition number is relatively small and should not introduce numerical issues.

4.3. Interpolation errors. In this section, we compute interpolation errors in the max norm for two functions using our constructed nodal sets. We iterate towards the max norm error $\|f - f_N\|$ using an adaptive sampling, similar to the manner in which the Lebesgue constant is computed. Tables 4 and 5 shows interpolation errors for two functions: a smooth analytic function f_1 and a Runge-type function f_2:

$$(4.1) \qquad f_1(r, s, t) = (r + 1)(s + 1)(t + 1) \cosh(r + s + t - 1),$$

$$(4.2) \qquad f_2(r, s, t) = \frac{1}{1 + (r^2 + s^2 + t^2)/2}.$$

We can observe from the results in Table 4 that for $N = 10$, the Fekete nodes give back the lowest interpolation error for the smooth analytic function, which is consistent with the Fekete nodes having the lowest Lebesgue constant at high N. Equispaced and conical/GLL nodes deliver larger interpolation errors, especially as N increases. However, the face node set (warp and blend faces with GLL interior nodes) does quite well for even N, which may be specific to the specific function f_1.

For the Runge-type function f_2, we see behavior more reflective of the Lebesgue constant of the resulting nodal set: the face nodal distribution results in a higher error than all other optimized node sets. For f_2, the interpolatory warp and blend nodes for the pyramid actually give back the lowest interpolation error for high N, despite their Lebesgue constant being larger than that of the Fekete nodes. For both the greedy and QR-based "approximate Fekete" points, the error appears to oscillate around the error of the true Fekete points.

We note that the reported interpolation errors differ by a very small margin over each point set. We note that this may be due to our choice of interpolated functions—the extremal function which achieves the upper bound given by the Lebesgue constant would differ between each nodal set, implying that some nodal sets may return lower interpolation errors for specific functions independently of the Lebesgue constant. Additionally, since the Lebesgue constant is only an upper bound on the interpolation error, the reported numerical errors may behave better than the Lebesgue constants would indicate. For comparison, the computed interpolation errors for various point sets over the triangle and tetrahedron are also very similar [31].

Finally, with the exception of the equispaced nodes, the Lebesgue constants and condition numbers for each point set are roughly the same order of magnitude up to $N = 10$. Thus, for low orders of approximation, we expect most point sets to behave similarly. The choice of point set may then be chosen based on other metrics (for example, the Fekete and "approximate Fekete" nodes do not admit an explicit construction, the conical point set does not obey a given face distribution, etc.).

At much higher orders of approximation, the rate of growth of the Lebesgue constant may result in more prominent distinctions between nodal sets. While numerical evidence suggest that Fekete nodes will deliver the smallest Lebesgue constants at high N, their cost of construction makes them difficult to determine at high orders. "Approximate Fekete" points may be an attractive alternative due to their relatively low cost and the asymptotic properties of their point distribution. Additionally, among explicitly constructed node sets, the Lebesgue constant of interpolatory warp and blend nodes grows the most slowly.

4.4. Optimization of the duplex/interpolatory warp and blend nodes. The final step in construction of the original warp and blend nodes on the triangle was the addition of a quadratic variation to the interior blend, creating a one-parameter family of nodal distributions. For the tetrahedron, a similar quadratic term is added, both to the edge blending functions and the face blending functions. A 1D optimization problem is then solved for the value of this parameter which minimizes the Lebesgue constant of the resulting distribution.

Both the duplex and interpolatory warp and blend procedures for the pyramid may also be optimized in a similar fashion. For the duplex construction, one such option is to optimize the edge/face blending parameter associated with the shared tetrahedral face inside the pyramid. For the interpolatory warp and blend procedure, we may also add a quadratic variation to the edge and face functions.[4] However, in our numerical experiments, neither optimization procedure improved the Lebesgue constant of the resulting nodal distribution significantly. Table 6 shows the difference in Lebesgue constant after optimization of α for the triangle, tetrahedron, duplex pyramid, and interpolatory warp and blend pyramid.[5] While the effect of optimization is noticable for both the 2D triangle and the 3D tetrahedron, the effect of optimization for both warp and blend pyramid nodes is much less pronounced.

Since this optimization depends completely on the choice of quadratic variation, we are currently investigating other parametrizations of blending functions to improve optimization of Lebesgue constant.

---

[4]For edges associated with one or more triangular faces, we multiply the blending function by $1 + \alpha v_c^2$, where for each bordering triangular face, v_c is the vertex that does not lie on the edge. For triangular faces, we multiply the blending function by $1 + \alpha p_e^2$, where $p_e(r, s, t)$ is the plane which is zero on the triangular face and 1 at the opposite edge, while for the quadrilateral face, we multiply the blending function by $1 + \alpha v_5^2$, where v_5 is the vertex function at the tip of the pyramid.

[5]The unoptimized duplex blending parameter is taken to be α instead of zero.

TABLE 6

Original and optimized Lebesgue constants for the triangle, tetrahedron, duplex pyramid, and interpolatory warp and blend pyramid. The left columns show the unoptimized Lebesgue constant, and the right columns show the Lebesgue constants after optimization of the construction in the previous column.

	Triangle		Tetrahedron		Duplex pyramid		Pyramid WB	
N	No opt	Opt	No opt	Opt	No opt	Opt	No opt	Opt
3	3.12	3.12	2.93	2.93	2.73	2.73	2.75	2.74
4	3.82	3.70	4.07	4.07	3.80	3.80	3.90	3.77
5	4.55	4.27	5.36	5.32	5.06	5.06	5.11	5.10
6	5.69	4.96	7.38	7.01	6.66	6.61	7.23	7.00
7	7.02	5.74	9.82	9.21	9.65	9.48	9.75	9.74
8	9.16	6.67	13.75	12.54	14.65	14.32	14.22	14.20
9	11.83	7.90	18.85	17.02	23.39	23.11	20.82	20.76
10	16.06	9.36	27.02	24.40	40.12	39.73	32.16	32.00

5. Conclusions. We have compared several methods for the construction of nodal sets on pyramids for conforming finite element methods. Both explicit and iterative procedures are considered—previously existing algorithms for computing both Fekete and "approximate Fekete" points are adapted to the pyramid, and a new duplex pyramid warp and blend procedure is introduced. Furthermore, a new interpolatory warp and blend procedure is developed and applied to the pyramid. Similarly to the triangle, warp-and-blend-based nodal sets deliver lower Lebesgue constants for moderate values of N, while the iteratively determined Fekete nodes give lower Lebesgue constants for $N \geq 7$. The condition number of the Vandermonde matrix and computed interpolation error for two smooth functions are also reported. For each nodal set (except for equilateral nodes) the Vandermonde matrix is well conditioned (up to $N = 10$) and interpolation errors are roughly the same order of magnitude, suggesting only small practical differences between each point distribution.

A directory containing files for the relevant nodal distributions is available for download on Github at https://github.com/tcew/nodes. Both MATLAB `.mat` files and script files with which to generate the given nodal sets are provided.

REFERENCES

[1] T. CARRIER BAUDOUIN, J.-F. REMACLE, E. MARCHANDISE, F. HENROTTE, AND C. GEUZAINE, *A frontal approach to hex-dominant mesh generation*, Adv. Model. Simul. Eng. Sci., 1 (2014), 8

[2] G BEDROSIAN, *Shape functions and integration formulas for three-dimensional finite element analysis*, Internat. J. Numer. Methods Engrg., 35 (1992), pp. 95–108.

[3] M. BERGOT, G. COHEN, AND M. DURUFLÉ, *Higher-order finite elements for hybrid meshes using new nodal pyramidal elements*, J. Sci. Comput., 42 (2010), pp. 345–381.

[4] M. J. BLUCK AND S. P. WALKER, *Polynomial basis functions on pyramidal elements*, Comm. Numer. Methods Eng., 24 (2008), pp. 1827–1837.

[5] L. BOS, J.-P. CALVI, N. LEVENBERG, A. SOMMARIVA, AND M. VIANELLO, *Geometric weakly admissible meshes, discrete least squares approximations and approximate Fekete points*, Math. Comp., 80 (2011), pp. 1623–1638.

[6] L. BOS, S. DE MARCHI, A. SOMMARIVA, AND M. VIANELLO, *Computing multivariate Fekete and Leja points by numerical linear algebra*, SIAM J. Numer. Anal., 48 (2010), pp. 1984–1999.

[7] M. BRIANI, A. SOMMARIVA, AND M. VIANELLO, *Computing Fekete and Lebesgue points: Simplex, square, disk*, J. Comput. Appl. Math., 236 (2012), pp. 2477–2486.

[8] J. Burkardt, *STROUD: Numerical integration in m dimensions*, https://people.sc.fsu.edu/~jburkardt/m_src/pyramid_rule/pyramid_rule.html (2014).

[9] Q. Chen and I. Babuška, *The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron*, Comput. Methods Appl. Mech. Engrg., 137 (1996), pp. 89–94.

[10] S. De Marchia and M. Vianelloa, *Polynomial approximation on pyramids, cones and solids of rotation*, in Proceedings of Dolomites Workshop on Constructive Approximation and Applications, 2012, Vol. 6 (2013), pp. 20–26.

[11] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, and A. Zdunek, *Computing with hp-Adaptive Finite Elements, Vol. 2. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman & Hall/CRC, Boca Raton, FL, 2007.

[12] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge Monogr. Appl. Comput. Math. 9, Cambridge University Press, Cambridge, 2002.

[13] M. Dubiner, *Spectral methods on triangles and other domains*, J. Sci. Comput., 6 (1991), pp. 345–390.

[14] G. J. Gassner, F. Lörcher, C.-D. Munz, and J. S. Hesthaven, *Polymorphic nodal elements and their application in discontinuous Galerkin methods*, J. Comput. Phys., 228 (2009), pp. 1573–1590.

[15] M. Gunzburger and A. L. Teckentrup, *Optimal Point Sets for Total Degree Polynomial Interpolation in Moderate Dimensions*, preprint, ArXiv:1407.3291, 2014.

[16] P. C. Hammer, O. J. Marlowe, and A. H. Stroud, *Numerical integration over simplexes and cones*, Mathematical Tables and Other Aids to Computation, 10 (1956), pp. 130–137.

[17] J. S. Hesthaven, *From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex*, SIAM J. Numer. Anal., 35 (1998), pp. 655–676.

[18] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Texts Appl. Math. 54, Springer, New York, 2008.

[19] G. Karniadakis and S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, Oxford, 2013.

[20] A. Narayan and D. Xiu, *Constructing nested nodal sets for multivariate polynomial interpolation*, SIAM J. Sci. Comput., 35 (2013), pp. A2293–A2315.

[21] N. Nigam and J. Phillips, *High-order conforming finite elements on pyramids*, IMA J. Numer. Anal., 32 (2012), pp. 448–483.

[22] N. Nigam and J. Phillips, *Numerical integration for high order pyramidal finite elements*, ESAIM Math. Model. Numer. Anal., 46 (2012), pp. 239–263.

[23] A. T. Patera, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phys., 54 (1984), pp. 468–488.

[24] F. Rapetti, A. Sommariva, and M. Vianello, *On the generation of symmetric Lebesgue-like points in the triangle*, J. Comput. Appl. Math., 236 (2012), pp. 4925–4932.

[25] M. J. Roth, *Nodal Configurations and Voronoi Tessellations for Triangular Spectral Elements*, Ph.D. thesis, University of Victoria, Victoria, Canada, 2005.

[26] S. J. Sherwin, T. C. E. Warburton, and G. Em Karniadakis, *Spectral/hp methods for elliptic problems on hybrid grids*, Contemp. Math., 218 (1998), pp. 191–216.

[27] A. Sommariva and M. Vianello, *Computing approximate Fekete points by {QR} factorizations of Vandermonde matrices*, Comput. Math. Appl., 57 (2009), pp. 1324–1336.

[28] M. A. Taylor, B. A. Wingate, and R. E. Vincent, *An algorithm for computing Fekete points in the triangle*, SIAM J. Numer. Anal., 38 (2000), pp. 1707–1720.

[29] M. Van Barel, M. Humet, and L. Sorber, *Approximating optimal point configurations for multivariate polynomial interpolation*, Electron. Trans. Numer. Anal., 42 (2014), pp. 41–63.

[30] T. Warburton, *Spectral/hp Methods on Polymorphic Multidomains: Algorithms and Applications*, Ph.D. thesis, Brown University, Providence, RI, 1999.

[31] T. Warburton, *An explicit construction of interpolation nodes on the simplex*, J. Engrg. Math., 56 (2006), pp. 247–262.

[32] C. Wieners, *Conforming Discretizations on Tetrahedrons, Pyramids, Prisms and Hexahedrons*, preprint Bericht 97/15, University of Stuttgart, 1997.

[33] S. Zaglmayr, *High Order Finite Element Methods for Electromagnetic Field Computation*, Ph.D. thesis, Johannes Kepler University, Linz, Austria, 2006.