A SHORT NOTE ON A BERNSTEIN-BEZIER BASIS FOR THE PYRAMID

JESSE CHAN, T. WARBURTON

Abstract. We introduce a Bernstein-Bezier basis for the pyramid, whose restriction to the face reduces to the Bernstein-Bezier basis on the triangle or quadrilateral. The basis satisfies the standard positivity and partition of unity properties common to Bernstein polynomials, and spans the same space as non-polynomial pyramid bases in the literature [1, 2, 3, 4]. Procedures for differentiation and integration of these basis functions are also discussed.

1. Introduction. Bernstein-Bezier bases have long been ubiquitous in graphics and computer-aided design [5, 6], though they have recently received attention for their utility in the meshing of curved geometries and the numerical solution of partial differential equations [7, 8, 9, 10, 11]. Ainsworth and Kirby both noticed that the structure of Bernstein-Bezier polynomials on simplices allowed for reduced-complexity algorithms for the assembly and multiplication of finite element matrices [12, 13, 14], as well as for efficient Discontinuous Galerkin solvers [15]. These results may be summarized as follows: for a basis of order N, it is possible to assemble local discretization matrices in $O(1)$ work per entry, and to apply these matrices for $O(N^{d+1})$ cost in d dimensions. The hierarchical structure of Bernstein polynomials also facilitates the construction of structure-preserving vectorial basis functions [16, 17]. Additionally, their close connection to B-splines has been exploited to simplify the implementation of NURBS-based finite element methods [18].

Bernstein bases on tensor product elements (quadrilaterals and hexahedra) enjoy a natural tensor-product construction, while the construction of Bernstein bases on simplices rely on barycentric coordinates to generalize the construction in one space dimension. Bernstein-Bezier bases on the prism may similarly be expressed as the tensor product of triangular and one-dimensional basis functions. However, Bernstein-Bezier bases for the pyramid (which acts as a transitional piece to couple together hexahedral and tetrahedral elements [3, 19]) have received less attention. Recent work has considered the extension of such ideas to pyramids, but define Bernstein-Bezier pyramid bases (which are distinct from Bernstein-Bezier pyramid algorithms [20, 21]) by splitting the pyramid into two tetrahedra [22, 23]. The work presented here presents an alternative construction based on pyramid bases found in the literature [1, 2, 4], which are non-polynomial, but contain the space of polynomials of total degree N under vertex-based mappings. Quadrature and the computation of matrices for finite element methods are also discussed, and condition numbers are presented for mass and stiffness matrices on the reference element.

2. Bernstein-Bezier bases for triangles and quadrilaterals. The 1D Bernstein polynomials of degree N are defined as

$$B_i^N(r) = \binom{N}{i} r^i (1-r)^{N-i}, \qquad 0 \le i \le N,$$

for coordinate $r \in [-1, 1]$. To extend this definition to higher dimensional simplicial domains, Bernstein-Bezier basis functions are defined in terms of the barycentric coordinates. For example, given barycentric coordinates $\lambda_1, \lambda_2, \lambda_3$ for the triangle, triangular Bernstein polynomials [5] are defined as

$$B_{ijk}^N = C_{ijk}^N \lambda_1^i \lambda_2^j \lambda_3^k$$

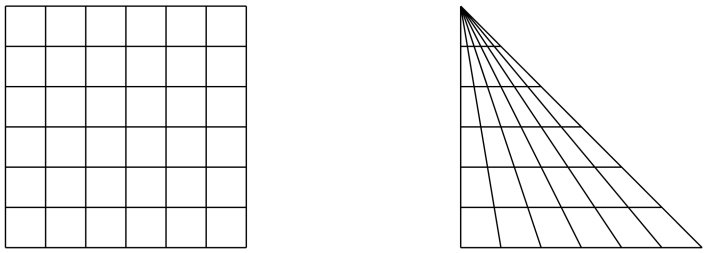


Fig. 2.1: An illustration of the Duffy transform mapping an equilateral grid on the quadrilateral to the triangle.

with positive integer indices i, j, k such that $i + j + k = N$ and

$$C_{ijk}^N = \frac{N\,!}{i\,!j\,!k\,!}.$$

For the unit right triangle with coordinates (r, s), these barycentric coordinates are given explicitly as

$$\lambda_1 = 1 - (r + s), \quad \lambda_2 = r, \quad \lambda_3 = s.$$

Assuming the Duffy transform

$$r = a(1 - b), \quad s = b,$$

which maps the unit quadrilateral with coordinates $(a, b) \in [0, 1]^2$ to the unit right triangle, we have that

$$\lambda_1 = (1 - a)(1 - b), \quad \lambda_2 = a(1 - b), \quad \lambda_3 = b.$$

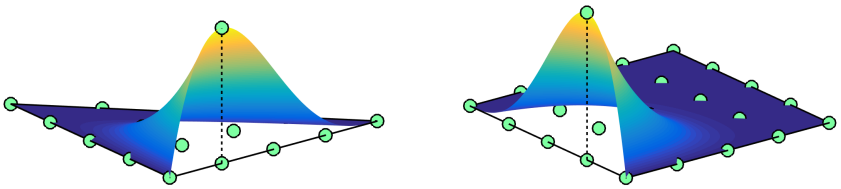
This gives an alternative definition for the Bezier triangle basis on the triangle

$$B_{ijk}^N = C_{ijk}^N a^j (1 - a)^i b^k (1 - b)^{i+j}.$$

Since $N = i + j + k$, $i + j = N - k$, and we may rewrite the above as

$$\begin{aligned} B_{ijk}^N(a, b) &= C_{ijk}^N a^j (1 - a)^i b^k (1 - b)^{N-k} \\ &= \frac{(N - k)\,!}{i\,!j\,!} a^j (1 - a)^i B_k^N(b) \\ &= B_i^{N-k}(a) B_k^N(b) = B_j^{N-k}(a) B_k^N(b) \end{aligned}$$

where $B_i^{N-k}(a)$ is the ith Bernstein polynomial of order $N - k$ and $B_k^N(b)$ is the kth Bernstein polynomial of order N. This decomposition is at the heart of the sum-factorization techniques used in [12, 13, 14].



(a) Triangular basis (b) Quadrilateral basis

Fig. 3.1: A Bernstein-Bezier basis function on a triangular and quadrilateral domain. Each basis function attains its maximum at a single equispaced point (in green).

3. A Bernstein-Bezier basis for the pyramid. We are interested in defining a high order Bernstein-Bezier basis for pyramidal elements, which are used primarily as transitional elements connecting tetrahedral and hexahedral elements. To facilitate conformity, it is desirable for basis functions on quadrilateral and triangular faces of the pyramid to coincide with tetrahedral and hexahedral basis functions on their respective faces.

Since the same barycentric approach is used to define Bernstein polynomials on a simplex of arbitrary dimension, the restriction of tetrahedral Bernstein-Bezier basis functions to a triangular face results in the triangular Bernstein-Bezier basis. Bernstein-Bezier basis functions on quadrilaterals or hexahedra are defined using a tensor product construction; for example, on a reference quadrilateral with coordinates (a, b),

$$B^N_{ij}(a, b) = B^N_i(a)B^N_j(b),$$

and for a reference hexahedron with coordinates (a, b, c), the Bernstein-Bezier basis is

$$B^N_{ijk}(a, b, c) = B^N_i(a)B^N_j(b)B^N_k(c).$$

Similarly to the tetrahedron, restricting hexahedral Bernstein-Bezier functions to a single face results in Bernstein-Bezier polynomials over the quadrilateral.

To extend the construction of the Bernstein-Bezier basis to the pyramid, we use a collapsed coordinate system [24]. We define the unit cube with coordinates $(a, b, c) \in [0, 1]^3$, such that the unit cube is mapped to the unit right pyramid through the transform

$$r = a(1 - c), \quad s = b(1 - c), \quad t = c. \tag{3.1}$$

The Bernstein-Bezier basis for the pyramid is defined on the unit cube as

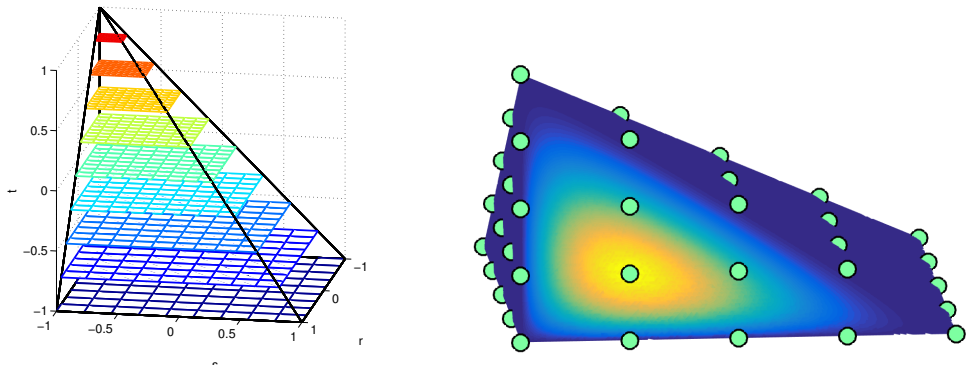
$$B_{ijk}(a, b, c) = B^{N-k}_i(a)B^{N-k}_j(b)B^N_k(c), \tag{3.2}$$

where the indices obey

$$0 \le k \le N, \quad 0 \le i, j \le N - k.$$

The total dimension of this space is $N_p = (N + 1)(N + 2)(2N + 3)/6$.

This construction of (3.2) is exactly what results from combining triangle Bernstein Bezier basis functions in the a, c and b, c coordinates. The Bernstein pyramid



(a) Cube-to-pyramid mapping (b) A Bernstein-Bezier polynomial on the pyramid

Fig. 3.2: Vertical slices of the cube mapped to the pyramid (left) and an example of a Bernstein-Bezier basis function on the pyramid (right), with equispaced points overlaid.

basis above satisfies the standard positivity and partition of unity properties, and the traces reduce to Bernstein-Bezier basis functions on the triangle or quadrilateral. This latter property simplifies the enforcement of conformity between pyramids and tetrahedral or hexahedral elements. Additionally, the Bernstein pyramid space spans the same space as the rational basis of Bergot, Cohen, and Durufle [1], and thus contains polynomials of total degree N under a vertex-based mapping of the unit right pyramid to physical space.

Lemma 3.1. *The Bernstein pyramid basis (3.2) satisfies a pointwise positivity and partition of unity property on the pyramid.*

Proof. Since each component of the Bernstein pyramid basis is pointwise positive on the cube, the basis is pointwise positive over the pyramid. We may also show that the partition of unity property for the Bernstein pyramid is preserved:

$$\sum_{k=0}^{N}\sum_{i=0}^{N-k}\sum_{j=0}^{N-k} B_{ijk} = \sum_{k=0}^{N} B_k(c) \sum_{i=0}^{N-k} B_i^{N-k}(a) \sum_{j=0}^{N-k} B_j^{N-k}(b) = 1$$

by the fact that B_k^N, B_i^{N-k}, B_j^{N-k} all satisfy a partition of unity property over $[0,1]$. □

Lemma 3.2. *The trace spaces of the Bernstein pyramid basis (3.2) are Bernstein polynomials on the faces.*

Proof. We may prove the first property on the unit cube by restricting a, b to either 0 or 1 for the triangular faces, and restricting $c = 0$ for the quadrilateral face. For the quadrilateral face, since $B_k(c) = 0$ for $k > 0$, the Bernstein basis is nonzero only if $k = 0$, resulting in a trace of the form

$$B_{ij0} = B_i^N(a) B_j^N(b),$$

which is exactly the tensor product Bernstein basis over the quadrilateral face. For the triangular faces, we take $a, b = 0, 1$. We show the trace space for $a = 0$; the other

faces are similar. If $a = 0$, the Bernstein basis is nonzero only for $i = 0$, and the nonzero Bernstein basis functions follow the form

$$B_{0jk} = B_j^{N-k}(b)B_k^N(c).$$

This is identical to the Bernstein basis on the triangle after a mapping to the unit quadrilateral. □

Lemma 3.3. *The Bernstein pyramid space is identical to the pyramid space of Bergot, Cohen, Durufle [1].*

Proof. This is simplest to show by showing equivalence with the pyramid basis presented in [4], which is also equivalent to the basis of Bergot, Cohen, and Durufle. This basis is constructed using Jacobi polynomials $P_i^{\alpha,\beta}$, which are orthogonal with respect to a weighted L^2 inner product, and Lagrange polynomials $\ell_i^k(a)$, defined using the set of $(k+1)$ Gauss-Legendre quadrature points. This "semi-nodal" basis is given as

$$\phi_{ijk}(a,b,c) = \ell_i^k(a)\ell_j^k(b)\left(\frac{1-c}{2}\right)^k P_{N-k}^{2k+3}(c),$$

or equivalently

$$\ell_i^{N-k}(a)\ell_j^{N-k}(b)\left(\frac{1-c}{2}\right)^{N-k} P_k^{2(N-k)+3}(c).$$

As the spaces spanned by each basis are of the same dimension, it simply remains to show that their span is identical. Since the functions in a, b

$$B_i^{N-k}(a)B_j^{N-k}(b), \qquad \ell_i^{N-k}(a)\ell_j^{N-k}(b), \qquad 0 \le i,j \le N-k$$

both span $Q_{N-k}(a,b)$, and the functions in c

$$\left(\frac{1-c}{2}\right)^{N-k} P_k^{2(N-k)+3}(c), \qquad B_k^N(c)$$

are both linearly independent homogeneous polynomials of total order N, the two bases span the same approximation space.

□

As a result, the Bernstein pyramid contains the space of polynomials P^N on any vertex-mapped pyramid (i.e. a pyramid whose mapping from the reference to physical coordinates is given by a low-order interpolant of the vertex positions of the physical pyramid), and high order accurate approximations may be constructed on general vertex-mapped pyramids.

Unlike the orthogonal bases for pyramids constructed previously, the Bernstein basis absorbs the extra homogenizing factors of c^k or $(1-c)^k$ into the definition of the Bernstein-Bezier basis in the c direction, resulting in a very concise formula.

3.1. Quadrature and mass matrices. Under the collapsed coordinate system, integrals on a physical pyramid $\mathcal{P}$ are computed via the transformation

$$\int_{\mathcal{P}} u \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z = \int_{\widehat{\mathcal{P}}} uJ \,\mathrm{d}r\,\mathrm{d}s\,\mathrm{d}t = \int_0^1\int_0^1\int_0^1 uJ(1-c)^2 \,\mathrm{d}a\,\mathrm{d}b\,\mathrm{d}c.$$

where J is the determinant of the Jacobian of the mapping from reference pyramid $\widehat{\mathcal{P}}$ to $\mathcal{P}$. For vertex-mapped pyramids, J, as well as all change of variables factors $\frac{\partial rst}{\partial xyz}$, are bilinear in the a, b coordinates and constant in c [1, Lemma 3.5]. An appropriate quadrature for the pyramid may then be constructed using the tensor product of one-dimensional Gaussian quadratures in the a, b coordinates and Gauss-Jacobi quadratures with weights $(2, 0)$ in the c coordinate, which integrates exactly the above expression.

It is possible to use the collapsed-coordinate nature of the pyramid basis to reduce the cost of evaluating basis functions at quadrature points [24]. Another alternative is to compute the entries of the mass matrix using moment-based computations. The entries of the mass matrix are given as

(3.3)
$$M_{ijk,lmn} = \int_{\mathcal{P}} \phi_{ijk}\phi_{lmn} = \int_{\widehat{\mathcal{P}}} \phi_{ijk}\phi_{lmn} J = \int_c B_k^N(c) B_n^N(c)(1-c)^2 \,\mathrm{d}c \int_a \int_b B_i^{N-k}(a) B_l^{N-n}(a) B_j^{N-k}(b) B_m^{N-n}(b) J(a,b) \,\mathrm{d}a \,\mathrm{d}b.$$

Fast moment-based algorithms for Bernstein-Bezier bases [13] use that the product of two Bernstein polynomials is again a scaled Bernstein polynomial, and that the integrals of Bernstein polynomials are explicitly known in terms of ratios of binomial coefficients. By representing $(1-c)^2$ and $J(a, b)$ in one and two-dimensional Bernstein form, respectively, the integrals in (3.3) may be broken up into integrals over each coordinate and computed using moment-based techniques.

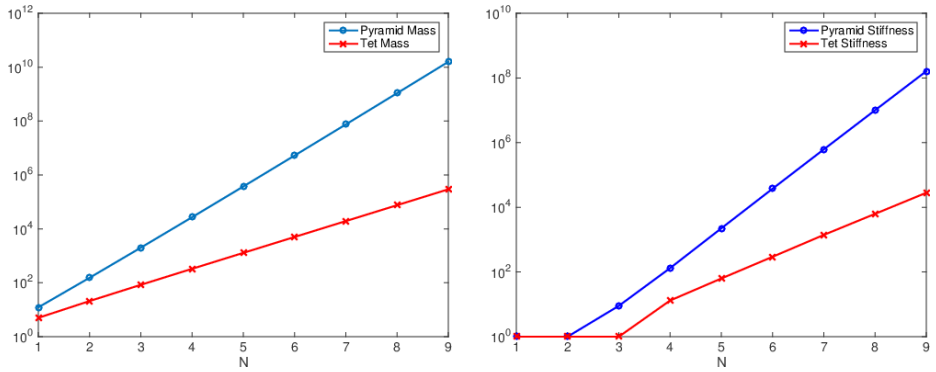
Additionally, since the Bernstein-Bezier basis is equivalent to the semi-nodal basis of [4], they are related by a linear transformation. This may be exploited for the low-storage inversion of local mass matrices. Since the semi-nodal basis yields a diagonal mass matrix for any vertex-mapped pyramid, the Bernstein-Bezier mass matrix may be inverted by transforming first to the semi-nodal basis, inverting a diagonal mass matrix, then transforming back. For a mesh consisting of many pyramids, this procedure requires less memory than directly factorizing or inverting the mass matrix, as all that is required is storage of change-of-basis matrices between semi-nodal and Bernstein-Bezier bases and the diagonal of the semi-nodal mass matrix over each pyramid.

3.2. Conditioning of mass matrices. The numerical sensitivity of a nonsingular matrix to perturbations can be estimated using the matrix condition number

$$\kappa(A) = \frac{\sigma_1}{\sigma_{N_p}},$$

where σ_1, σ_{N_p} are the largest and smallest singular values of A, respectively.

Figure 3.3 reports computed condition numbers of the reference mass and stiffness matrices for both pyramidal and tetrahedral Bernstein-Bezier bases at various orders of approximation. Dirichlet boundary conditions are enforced to ensure that the stiffness matrix is nonsingular. The condition number of each matrix grows exponentially in the order N, as expected for Bernstein-Bezier type bases. However, the growth of the condition numbers for the pyramid is more rapid than the growth of the condition number for the tetrahedra, and may need to be addressed in order to maintain numerical accuracy at high orders of approximation.



(a) Mass matrix (b) Stiffness matrix

Fig. 3.3: Condition numbers of reference tetrahedral and pyramidal Bernstein-Bezier mass and stiffness matrices at various orders of approximation N.

3.3. Derivative matrices. To compute derivatives with respect to reference coordinates r, s, t, we use the chain rule, involving factors of the transform (3.1)

$$\frac{\partial a}{\partial r} = \frac{\partial b}{\partial s} = \frac{1}{1-c}, \qquad \frac{\partial a}{\partial t} = \frac{r}{(1-t)^2} = \frac{a}{1-c}$$
$$\frac{\partial b}{\partial t} = \frac{s}{(1-t)^2} = \frac{b}{1-c}.$$

If $k = N$, the derivative is zero. For $k < N$, we have

$$\frac{\partial B^N_{ijk}}{\partial r} = \frac{\partial B^{N-k}_i(a)}{\partial a} B^{N-k}_j(b) \frac{B^N_k(c)}{1-c}$$
$$\frac{\partial B^N_{ijk}}{\partial s} = B^{N-k}_i(a) \frac{\partial B^{N-k}_j(b)}{\partial b} \frac{B^N_k(c)}{1-c}$$
$$\frac{\partial B^N_{ijk}}{\partial t} = \frac{\partial B^N_{ijk}}{\partial a}\left(\frac{a}{1-c}\right) + \frac{\partial B^N_{ijk}}{\partial b}\left(\frac{b}{1-c}\right) + \frac{\partial B^N_{ijk}}{\partial c}.$$

The additional factors of a, b and $1/(1-c)$ may be absorbed into the Bernstein polynomials, and properties of one-dimensional Bernstein-Bezier polynomials may be used to rewrite derivatives in terms of Bernstein polynomials. For example, the expression for $\frac{\partial B^N_{ijk}}{\partial r}$ simplifies to

$$\frac{\partial B^{N-k}_i(a)}{\partial a} B^{N-k}_j(b) \frac{B^N_k(c)}{1-c} = \begin{cases} \frac{\partial B^{N-k}_i(a)}{\partial a} B^{N-k}_j(b) \frac{\binom{N}{k}}{\binom{N-1}{k}} B^{N-1}_k(c), & k < N \\ 0, & k = N. \end{cases}.$$

Using one-dimensional formulas for derivatives of Bernstein polynomials, the derivative with respect to a can be written as a short linear combination

$$\frac{\partial B^{N-k}_i(a)}{\partial a} = (N-k)\left(B^{N-k-1}_{i-1}(a) - B^{N-k-1}_i(a)\right).$$

From a linear algebraic perspective, it can be convenient to define a derivative matrices D_r which map expansion coefficients of a function to expansion coefficients of its derivative. In [25], these matrices were shown to be sparse. For the pyramid, it is possible to recover a similar sparsity, though this requires a more nuanced definition of the derivative matrix as a mapping.

The standard derivative matrix is defined as $\boldsymbol{D}_r : V_h \to V_h$, such that it maps the discrete approximation space to itself. However, $\boldsymbol{D}_r$ is not sparse under the Bernstein-Bezier basis for the pyramid. To introduce sparsity, we first define the auxiliary Bernstein-Bezier basis $\tilde{B}^N_{ijk}$

$$\tilde{B}^N_{ijk} = B^{N-k}_i(a) B^{N-k}_j(b) B^{N-1}_k(c), \qquad 0 \le k \le N-1, \quad 0 \le i,j \le k.$$

Then, introducing the vector spaces V_h and $\tilde{V}_h$

$$V_h = \text{span}\left\{B^N_{ijk}\right\}, \qquad \tilde{V}_h = \text{span}\left\{\tilde{B}^N_{ijk}\right\},$$

an "auxiliary" derivative matrix $\tilde{\boldsymbol{D}}_r$ can be defined as

$$\tilde{\boldsymbol{D}}_r : V_h \to \tilde{V}_h$$

such that

$$\begin{aligned} u(a,b,c) &= \sum_{k=0}^{N} \sum_{i,j=0}^{k} \boldsymbol{u}_{ijk} B^N_{ijk}(a,b,c) \\ \frac{\partial u(a,b,c)}{\partial r} &= \sum_{k=0}^{N-1} \sum_{i,j=0}^{k} \left(\tilde{\boldsymbol{D}}_r \boldsymbol{u}\right)_{ijk} \tilde{B}^N_{ijk}(a,b,c), \end{aligned}$$

where it is implicitly understood that

$$(a,b,c) = \left(a(r,s,t), b(r,s,t), c(r,s,t)\right).$$

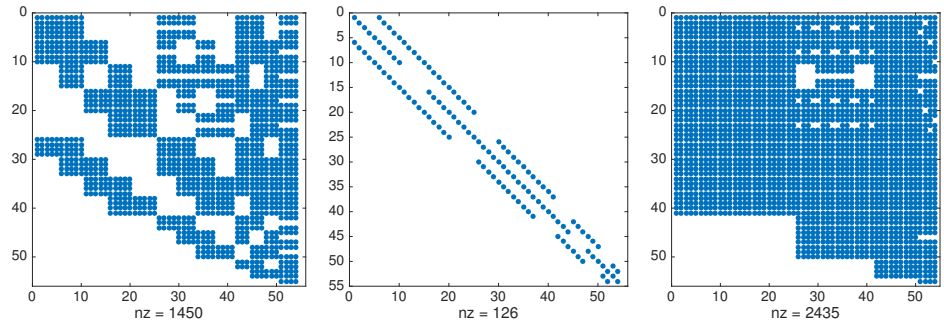
The two derivative matrices $\boldsymbol{D}_r$ and $\tilde{\boldsymbol{D}}_r$ are related through $\boldsymbol{D}_r = \tilde{\boldsymbol{E}}\tilde{\boldsymbol{D}}_r$, where $\tilde{\boldsymbol{E}} : \tilde{V}_h \to V_h$ is an "auxiliary" degree elevation matrix such that

$$\begin{aligned} u(a,b,c) &= \sum_{k=0}^{N} \sum_{i,j=0}^{k} \boldsymbol{u}_{ijk} B^N_{ijk}(a,b,c) \\ u(a,b,c) &= \sum_{k=0}^{N-1} \sum_{i,j=0}^{k} \left(\tilde{\boldsymbol{E}} \boldsymbol{u}\right)_{ijk} \tilde{B}^N_{ijk}(a,b,c). \end{aligned}$$

While we have not derived exact expressions for entries of $\tilde{\boldsymbol{E}}$, it is straightforward to compute numerically (for example, as a projection of $\tilde{V}_h$ onto V_h). The sparsity pattern of each of these matrices is shown in Figure 3.4. Derivatives with respect to s are computed in a similar fashion.

Defining sparse derivative matrices with respect to t is slightly more involved. The chain rule implies

$$\frac{\partial B^N_{ijk}}{\partial t} = \frac{\partial B^N_{ijk}}{\partial a}\left(\frac{a}{1-c}\right) + \frac{\partial B^N_{ijk}}{\partial b}\left(\frac{b}{1-c}\right) + \frac{\partial B^N_{ijk}}{\partial c}.$$



(a) $\boldsymbol{D}_r$ (b) $\tilde{\boldsymbol{D}}_r$ (c) $\tilde{\boldsymbol{E}}$

Fig. 3.4: Sparsity patterns of derivative matrices $\boldsymbol{D}_r$, $\tilde{\boldsymbol{D}}_r$ and auxiliary degree elevation matrix $\tilde{\boldsymbol{E}}$ for $N = 4$.

Examining $\frac{\partial B_{ijk}^N}{\partial a}\left(\frac{a}{1-c}\right)$, we have

$$\begin{aligned}\frac{\partial B_{ijk}^N}{\partial a}\left(\frac{a}{1-c}\right) &= a\frac{\partial B_i^{N-k}(a)}{\partial a}B_j^{N-k}(b)\frac{B_k^N(c)}{1-c} \\ &= (N-k)\left(aB_{i-1}^{N-k-1}(a) - aB_i^{N-k-1}(a)\right)B_j^{N-k}(b)\frac{\binom{N}{k}}{\binom{N-1}{k}}B_k^{N-1}(c).\end{aligned}$$

The extra factor of a may be combined with a lower degree Bernstein polynomial, resulting in a higher degree Bernstein polynomial

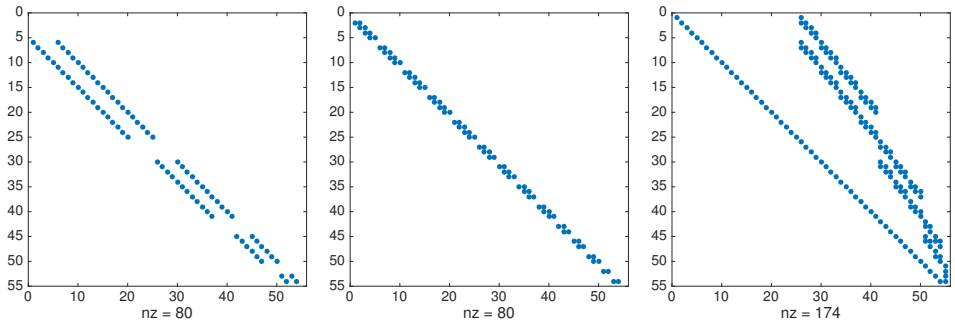
$$aB_i^{N-k-1}(a) = \binom{N-k-1}{i}a^{i+1}(1-a)^{N-k-(i+1)} = \frac{\binom{N-k-1}{i}}{\binom{N-k}{i+1}}B_{i+1}^{N-k}(a).$$

This sparsifies the matrices $\tilde{\boldsymbol{D}}_t^a, \tilde{\boldsymbol{D}}_t^b, \tilde{\boldsymbol{D}}_t^c$, which are defined implicitly through

$$\begin{aligned}u(a,b,c) &= \sum_{k=0}^{N}\sum_{i,j=0}^{k}\boldsymbol{u}_{ijk}B_{ijk}^N(a,b,c) \\ a\frac{\partial u(a,b,c)}{\partial a} &= \sum_{k=0}^{N-1}\sum_{i,j=0}^{k}\left(\tilde{\boldsymbol{D}}_t^a\boldsymbol{u}\right)_{ijk}\tilde{B}_{ijk}^N(a,b,c) \\ b\frac{\partial u(a,b,c)}{\partial b} &= \sum_{k=0}^{N-1}\sum_{i,j=0}^{k}\left(\tilde{\boldsymbol{D}}_t^b\boldsymbol{u}\right)_{ijk}\tilde{B}_{ijk}^N(a,b,c) \\ \frac{\partial u(a,b,c)}{\partial c} &= \sum_{k=0}^{N-1}\sum_{i,j=0}^{k}\left(\tilde{\boldsymbol{D}}_t^c\boldsymbol{u}\right)_{ijk}\tilde{B}_{ijk}^N(a,b,c).\end{aligned}$$

The t-derivative matrix $\tilde{D}_t$ can be defined as the sum of these matrices

$$\tilde{\boldsymbol{D}}_t = \tilde{\boldsymbol{D}}_t^a + \tilde{\boldsymbol{D}}_t^b + \tilde{\boldsymbol{D}}_t^c.$$



(a) $\tilde{\boldsymbol{D}}_t^a$ (b) $\tilde{\boldsymbol{D}}_t^b$ (c) $\tilde{\boldsymbol{D}}_t^c$

Fig. 3.5: Sparsity patterns of derivative matrices $\tilde{\boldsymbol{D}}_t^a, \tilde{\boldsymbol{D}}_t^b, \tilde{\boldsymbol{D}}_t^c$ for $N = 4$.

Straightforward calculations similar to those done in [25] can be used to show that the matrices $\tilde{\boldsymbol{D}}_t^a, \tilde{\boldsymbol{D}}_t^b, \tilde{\boldsymbol{D}}_t^c$ contain a fixed maximum number of nonzeros per row independent of N, allowing for efficient calculation of derivatives.

We note that computing physical derivatives requires multiplying reference derivatives by geometric factors, which are non-constant for general pyramids. If these geometric factors are represented in Bernstein form, this multiplication may be done efficiently using the moment-based algorithms described in [13] and Section 3.1 applied to the auxiliary basis $\tilde{B}_{ijk}^N$.

While the derivative matrices derived in this section allow for efficient computation of derivatives, they do so by treating the derivative as a map between two different spaces. Numerical methods such as time-domain Discontinuous Galerkin [26] or pseudo-spectral methods are often formulated in terms of derivative matrices mapping from V_h to itself. Unfortunately, since the auxiliary degree elevation matrix is dense, it is presently unclear how to efficiently apply such derivative matrices in this setting.

4. Acknowledgements. JC would like to thank John Evans for informative discussions and for posing the question of how to construct a Bernstein-Bezier basis on the pyramid. Both authors would like to acknowledge the support of NSF (award number DMS-1216674) in this research.

REFERENCES

[1] Morgane Bergot, Gary Cohen, and Marc Duruflé. Higher-order finite elements for hybrid meshes using new nodal pyramidal elements. *Journal of Scientific Computing*, 42(3):345–381, 2010.

[2] Nilima Nigam and Joel Phillips. High-order conforming finite elements on pyramids. *IMA Journal of Numerical Analysis*, 32(2):448–483, 2012.

[3] Morgane Bergot and Marc Duruflé. Higher-order discontinuous Galerkin method for pyramidal elements using orthogonal bases. *Numerical Methods for Partial Differential Equations*, 29(1):144–169, 2013.

[4] Jesse Chan and T Warburton. Orthogonal bases for vertex-mapped pyramids. *arXiv preprint arXiv:1502.07703*, 2015. Accepted to SISC.

[5] Gerald Farin. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3(2):83–127, 1986.

[6] Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

[7] F Hindenlang, G Gassner, T Bolemann, and CD Munz. Unstructured high order grids and their application in discontinuous Galerkin methods. In *Conference Proceedings, V European Conference on Computational Fluid Dynamics ECCOMAS CFD*, pages 1–8, 2010.

[8] Amaury Johnen, J-F Remacle, and Christophe Geuzaine. Geometrical validity of high-order triangular finite elements. *Engineering with Computers*, 30(3):375–382, 2014.

[9] Christophe Geuzaine, Amaury Johnen, Jonathan Lambrechts, J-F Remacle, and Thomas Toulorge. The generation of valid curvilinear meshes. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pages 15–39. Springer, 2015.

[10] C Michoski, J Chan, L Engvall, and JA Evans. Foundations of the Blended Isogeometric Discontinuous Galerkin (BIDG) method. 2015.

[11] Amaury Johnen and Christophe Geuzaine. Geometrical validity of curvilinear pyramidal finite elements. *Journal of Computational Physics*, 299(C):124–129, 2015.

[12] Robert C Kirby. Fast simplicial finite element algorithms using Bernstein polynomials. *Numerische Mathematik*, 117(4):631–652, 2011.

[13] Mark Ainsworth, Gaelle Andriamaro, and Oleg Davydov. Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures. *SIAM Journal on Scientific Computing*, 33(6):3087–3109, 2011.

[14] Robert C Kirby and Kieu Tri Thinh. Fast simplicial quadrature-based finite element operators using Bernstein polynomials. *Numerische Mathematik*, 121(2):261–279, 2012.

[15] Robert C Kirby. Efficient discontinuous Galerkin finite element methods via bernstein polynomials. *arXiv preprint arXiv:1504.03990*, 2015.

[16] Robert C Kirby. Low-complexity finite element algorithms for the de Rham complex on simplices. *SIAM Journal on Scientific Computing*, 36(2):A846–A868, 2014.

[17] Mark Ainsworth, Gaelle Andriamaro, and Oleg Davydov. A Bernstein–Bézier Basis for arbitrary order Raviart-Thomas finite elements. *Constructive Approximation*, 41(1):1–22, 2015.

[18] Michael J Borden, Michael A Scott, John A Evans, and Thomas JR Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1-5):15–47, 2011.

[19] Jesse Chan, Zheng Wang, Axel Modave, Jean-Francois Remacle, and T Warburton. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *arXiv preprint arXiv:1507.02557*, 2015.

[20] Ron Goldman. *Pyramid algorithms: A dynamic programming approach to curves and surfaces for geometric modeling*. Morgan Kaufmann, 2002.

[21] Mark Ainsworth. Pyramid algorithms for Bernstein–Bézier finite elements of high, nonuniform order in any dimension. *SIAM Journal on Scientific Computing*, 36(2):A543–A569, 2014.

[22] Juan Chen, Chong-Jun Li, and Wan-Ji Chen. A 3D pyramid spline element. *Acta Mechanica Sinica*, 27(6):986–993, 2011.

[23] Mark Ainsworth, Oleg Davydov, and Larry L. Schumaker. Bernstein-Bezier finite elements on tetrahedral-hexahedral-pyramidal partitions. 2015.

[24] George Karniadakis and Spencer J Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.

[25] Jesse Chan and T Warburton. GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave problems. *arXiv preprint arXiv:1512.06025*, 2015.

[26] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.