

# Foundations of the Blended Isogeometric Discontinuous Galerkin (BIDG) Method

C. Michoski<sup>‡\*</sup>, J. Chan<sup>‡</sup>, L. Engvall<sup>\*</sup>, J.A. Evans<sup>\*</sup>

*Aerospace Engineering Sciences, Computational Mechanics and Geometry Laboratory (CMGLab)\**

*University of Colorado at Boulder, Boulder, CO 80302,*

*Institute for Computational Engineering and Sciences (ICES)<sup>†</sup>*

*University of Texas at Austin, Austin, TX 78712,*

*Computational and Applied Mathematics (CAAM)<sup>‡</sup>*

*Rice University, Houston, TX 77005*

## Abstract

A new discontinuous Galerkin (DG) method is introduced that seamlessly merges exact geometry with high-order solution accuracy. This new method is called the blended isogeometric discontinuous Galerkin (BIDG) method. The BIDG method contrasts with existing high-order accurate DG methods over curvilinear meshes (e.g. classical isoparametric DG methods) in that the underlying geometry is exactly preserved at every mesh refinement level, allowing for intricate and complicated real-world mesh design to be streamlined and automated using computer-aided design (CAD) software. The BIDG method is designed specifically for easy incorporation into existing code architecture. This paper discusses specific details of implementation using two examples: (1) the acoustic wave equations, and (2) Maxwell's equations. Basic tests of accuracy and stability are demonstrated, including optimal convergence, and supplemental theoretical results are provided in the appendix along with links to fully operational working code examples.

## Contents

<b>I. Introduction &amp; Background</b>	<b>2</b>
<b>II. The Classical Discontinuous Galerkin</b>	<b>6</b>
i. Basic Notation and Terminology . . . . .	6
ii. Classical Approximation Spaces and Choice of Basis Functions . . . . .	7
iii. Application to Acoustics . . . . .	8
iv. Application to Electromagnetics . . . . .	8

---

<sup>‡</sup>Corresponding author, [michoski@gmail.com](mailto:michoski@gmail.com)

<b>III. The Blended Isogeometric Discontinuous Galerkin Method</b>	<b>9</b>
i. Computer Aided Geometric Design: B-splines, NURBS, and T-splines . . . . .	9
ii. Rational Bézier Representational Geometries . . . . .	11
iii. Blending Algorithms and Approximation Spaces . . . . .	13
iv. Implementation and Incorporation . . . . .	16
a. Application to Acoustics . . . . .	16
b. Application to Electromagnetics . . . . .	17
<b>IV. Numerical Examples</b>	<b>18</b>
i. Application to Acoustics . . . . .	18
a) Unit Radius Resonant Drum: Convergence Study . . . . .	18
b) Acoustic Resonator . . . . .	20
ii. Application to Electromagnetics . . . . .	21
a) Unit Radius Electromagnetic Cavity: Eigenspectrum Analysis . . . . .	21
b) Unit Radius Electromagnetic Cavity: Convergence Study . . . . .	23
c) Cavity Magnetron . . . . .	24
<b>V. Summary and Conclusions</b>	<b>24</b>
<b>VI. Acknowledgements</b>	<b>25</b>
<b>VII. Appendix</b>	<b>26</b>

## I. Introduction & Background

High-order accurate discontinuous Galerkin (DG) finite element methods (FEM) are finding broad application in large-scale data intensive science and engineering problems [1, 16, 25, 34, 36, 43, 46, 59, 65]. Substantial benefits can be found in utilizing high-order accurate methods over their lower order counterparts. High-order methods not only induce greater solution accuracy, but they are also able to demonstrate greater computational efficiency, greater representational flexibility, as well as being able to be constructed to explicitly preserve derived physical features of solutions; features that are often essential to working physicists and engineers [10–12, 26, 29, 42, 48, 69, 70].

Though DG methods have gained increasing traction in large-scale application modeling and analysis, a shortcoming in the conventional DG methodology is the inability to *fully* recover complex underlying geometries in the meshing domain. Indeed, traditional meshing software has frequently only provided piecewise linear geometries, and has thus led to the need to support more realistic meshing approaches [27, 28, 39, 63]. Even so, linear elements can have the benefit of generating simple meshes for finite element analysis, but have the drawback of being incapable of representing many realistic physical domains (e.g. airplane wings, submarines propellers, wind turbines, coastline topographies, reactor cores) without performing excessive mesh refinement.

Improvements in mesh generation algorithms [35, 45, 71] are now providing curvilinear corrections to traditionally linear meshing routines, leading to the increasingly important question of how DGFEM algorithms can be constructed to exploit high-order accurate meshes. A number of clever approaches have been developed for supporting curvilinear meshes [21, 38, 50, 62, 67, 68], all of

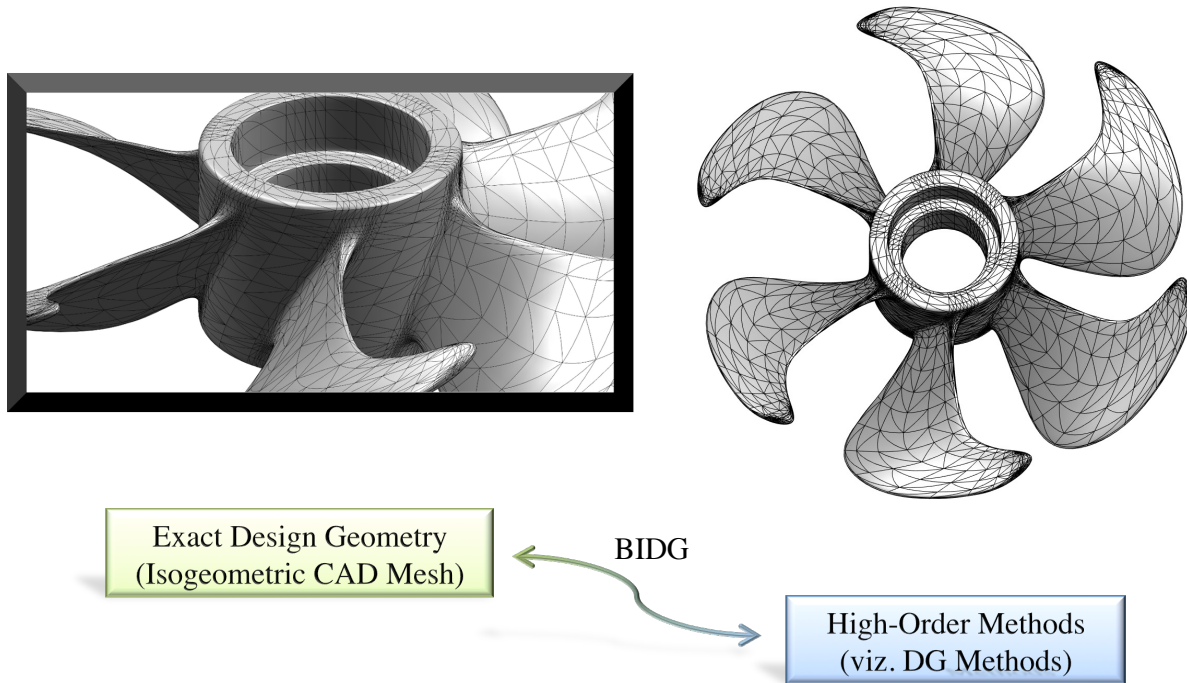


Figure 1: The blended isogeometric discontinuous Galerkin (BIDG) method *seamlessly* maps “exact design geometry” to high-order accurate discontinuous Galerkin methods. The TriGA software [19] takes CAD meshes (as shown in the top two images), and makes analysis/BIDG-suitable triangular/tetrahedral meshes for analysis, as indicated below.

which are invariably based on “node-snapping” techniques. Node-snapping techniques work by taking parameterized curves/surfaces, and systematically “snapping” support points to lie exactly on edges/faces of the parameterized curves/surfaces. The remaining interior points are then distorted over the element in a consistent way, so that many of the basic approximation properties of the domain interior are preserved.

While these node-snapping techniques have been shown to be able to — under many circumstances — recover optimal orders of convergence, they remain fundamentally reliant on the assumptions inherent to all isoparametric methods. In isoparametric methods curvilinear physical domains  $\Omega$  are locally approximated using piecewise polynomials, leading to a classical variational crime [60] along geometric boundaries, where the exact geometry along curvilinear boundary elements is *not* preserved in the isoparametric representation. This inability to exactly represent geometry has a number of far-reaching ramifications.

To briefly illustrate the inability of the isoparametric approach to exactly represent geometry, observe the geometric factors shown in figure 2. What is made clear upon close inspection, is that the classical isoparametric approach broadly fails to preserve exact curvilinear geometry, in that: (a) points on the domain no longer necessarily exactly lie on the parameterized curve, (b) volumes, normals, and variational derivatives at the boundary are no longer preserved, and (c) at support points local geometric continuity is reduced, lost, and/or broken. Loss of accuracy due to (a)-(c) are

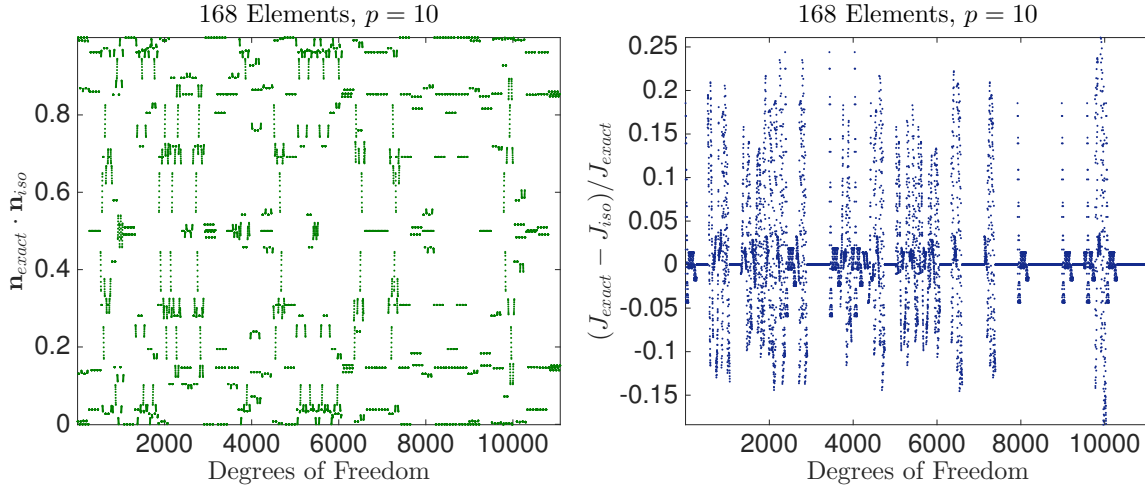


Figure 2: An exactly parameterized (i.e. isogeometric) unit circle with  $p = 10$  PKDO polynomial DG basis versus a  $p = 10$  fully isoparametric circle using a PKDO basis using the blend and warp method discussed in [30]. On the left is the scalar product of the normal vectors evaluated at the degrees of freedom. The degrees of freedom are 11088 as defined by the number of elements 168 times  $(p + 1)(p + 2)/2$ . On the right the variation in each degree of freedom of the Jacobian determinant between the exact (isogeometric) and isoparametric approaches. The mesh corresponds to the circle in section 4.

well-documented in fluid dynamics, where the formation of spurious boundary layers, for example, can have substantial impact on solution behavior [5].

However, the ramifications of fully encoding the exact geometry of a given discretization are considerably more wide-ranging than just a loss of solution accuracy. When the complete geometric content of the mesh becomes available, what one recovers along with it is the ability to develop automated supplemental tools for simulations that are traditionally time-intensive. For example, when the exact geometric content of a mesh is fully supported, then the analysis regime (i.e. DG method) is capable of fully and exactly coupling to the computer aided design (CAD) infrastructure, in that meshes and domains can be passed between CAD and DG algorithms directly. The ability to seamlessly interface analysis with design leads to the capability to automate analysis-suitable mesh generation for complicated geometries, as well as to automate exact geometry preserving mesh refinement strategies that have been traditionally difficult and expensive to reliably implement. For example, with an isoparametric mesh, one must communicate with the underlying CAD in order to preserve rates of convergence at each state of refinement.

Perhaps even more compelling is the capability of isogeometric methods to seamlessly incorporate broad classes of sophisticated design tools directly into the finite element analysis methodology itself. These design tools can be invaluable in the modern FEA setting, where simulations are frequently run in tandem with uncertainty quantification, sensitivity analysis, and parameter inference models, in order to identify optimal settings, determine limits in fault tolerances, and certify risk-aversion in engineering design approaches. For example, traditional structural shape optimization algorithms have suffered from too loose a link between exact CAD geometries, and the isopara-

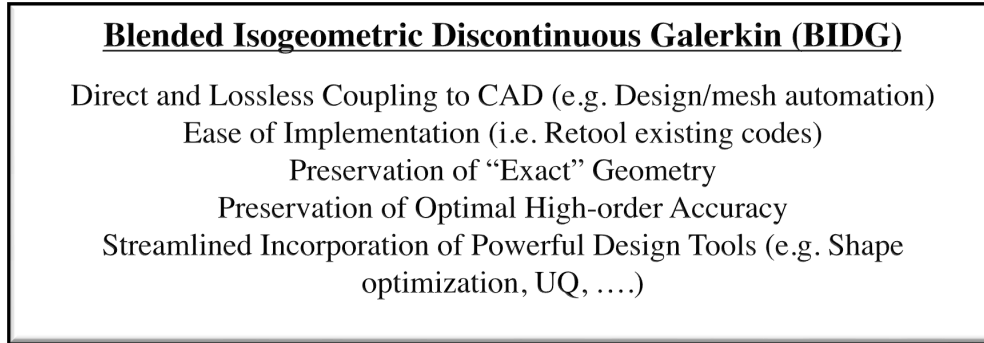


Figure 3: The salient features of the BIDG method.

metric approximations of them [64]. Even more broadly, in large-scale engineering and science applications such as turbulence [18, 44], structural vibrations [22], thermoelastoplasticity [4, 24], jet actuators [58], electromagnetism [13], coastal channels [68], etc., the sensitivity to geometry is being increasingly established in the sense that in these contexts changes in local representational geometries can have large impacts on the basic physics of the problem.

A recent renaissance in engineering design and analysis [52] has provided the fundamental tools necessary to solve the problem of exact representational geometry in high-order accurate DGFEM methods. Namely, the new field of Isogeometric Analysis (IGA) [31] introduced by T.J.R. Hughes, J.A. Cottrell and Y. Basilevs, has been developed to address the inconsistency between the communities of engineering design and analysis [8]. In the design community elaborate curvilinear meshes have traditionally been fashioned using computer aided design (CAD) technologies such as B-splines [53], and Bernstein-Bézier or NURBS patches, in order to parameterize sophisticated geometric bodies, such as airplanes, automobiles, biological organs, bridges, jets, reactors, and propellers (e.g. see figure 1). The process of converting a given CAD description into an analysis-suitable geometry is often expensive and error-prone, especially for large-scale engineering applications. In the engineering community, this has frequently been referred to as the “design-to-analysis bottleneck.” IGA bypasses the design-to-analysis bottleneck by directly employing the CAD description of geometry within the analysis framework. This is generally accomplished by adopting the geometric basis as the basis for analysis, thereby exactly preserving the CAD geometry.

In this paper the blended isogeometric discontinuous Galerkin (BIDG) method is introduced. The BIDG method is the Discontinuous Galerkin method formulated on elements that exactly preserve the geometries generated by CAD tools. The BIDG method seamlessly combines exact geometric design (CAD) with high-order accurate analysis (DGFEM), as schematically shown in figure 1. Some groundwork for BIDG-type methods is provided in [3, 6, 47, 55, 57], with patch-based methods recently being presented in [40, 41, 72] for elliptic problems. Also, please note the remark at the end of the appendix for a brief discussion of what is exactly meant by “isogeometric” throughout this paper.

The BIDG method is notable in that it is remarkably simple to implement, as all calculations are performed relative to the parametric space. Using this simplified approach, the BIDG method can be rapidly and easily incorporated into existing code architectures, by simply computing the

geometric factors in the parametric space, and then performing a straight-forward change of variables transformation upon integration. A basic overview of the salient features of the BIDG method is given in figure 3.

An outline of this paper is as follows. The basic notation of DG methods is presented in section 2, along with some discussion about the relevant approximation spaces, basis functions, and example problem formulations used throughout the paper. In section 3 a simple overview of relevant topics in IGA is given, followed by a detailed discussion of the blending algorithm used to seamlessly incorporate IGA into DGFEM; resulting in the BIDG method, with a followup discussion on how BIDG is derived, implemented and/or incorporated into existing codes. Finally, in section 4 two numerical examples are analyzed. The acoustic wave equations are first presented, followed by a form of Maxwell's equations, where both systems are modeled over several curvilinear domains. Comparisons with classical isoparametric DGFEM are performed, along with some spectral analysis and convergence studies. In section 5 some conclusions and future directions are presented. Finally, after some acknowledgements, the appendix in section 7 provides a theorem on the approximation properties of the BIDG method, along with some technical discussion.

## II. The Classical Discontinuous Galerkin

### i. Basic Notation and Terminology

The spatial discretization of an arbitrary physical domain  $\Omega$  can be performed relative to classical discontinuous Galerkin methods. This paper restricts to the two dimensional case in order to provide as many explicit details as possible. The follow-up paper will focus on large-scale applications and implementational features of the BIDG method as they arise in science and engineering in full dimension.

Consider the open set  $\Omega \subset \mathbb{R}^2$  with physical boundary  $\partial\Omega$ . Let  $\mathcal{T}_h$  denote the partition of the closure of a triangular mesh of  $\Omega$ , denoted  $\Omega_h$ , where the  $i$ th element is denoted  $\Omega_i$ , such that  $\mathcal{T}_h = \{\Omega_1, \Omega_2, \dots, \Omega_N\}$ , for  $N \in \mathbb{N}$  the number of elements in  $\Omega_h$  and  $i = 1, \dots, N$ . The discrete boundary domain is similarly denoted by  $\partial\Omega_h$ . Here and below, the mesh diameter  $h$  is chosen to satisfy  $h = \max_{ij}(d_{ij})$  for distance function  $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$  and face vertices  $\mathbf{x}_i, \mathbf{x}_j \in \partial\Omega_\ell$ .

The edge shared by two neighboring elements  $\Omega_i$  and  $\Omega_j$  is denoted by  $\Gamma_{ij}$ . These edges may correspond to interior element edges, or those located on the physical boundary  $\partial\Omega_h$  itself, in which case the outer boundary element corresponds instead to a prescribed boundary condition. The outward pointing normal along these edges is defined by  $\mathbf{n} = (n_x, n_y)$ . Then for some  $v$  evaluated on  $\partial\Omega_h$ , define the standard inter-element average and jump operators along a shared edge, respectively, as

$$\{v\} = \frac{1}{2}(v^+ + v^-), \quad \llbracket v \rrbracket = \mathbf{n}^- \cdot \mathbf{v}^- + \mathbf{n}^+ \cdot \mathbf{v}^+, \quad \llbracket \mathbf{v} \rrbracket = \mathbf{n}^- \cdot \mathbf{v}^- + \mathbf{n}^+ \cdot \mathbf{v}^+.$$

Here  $v^+$  indicates the function evaluation along the base element's edge, and  $v^-$  indicates the function evaluation along its neighbor's edge. To fully characterize the method, the matching condition:

$$[v] = \mathbf{n} \cdot \llbracket v \rrbracket = v^- - v^+,$$

is also required.

## ii. Classical Approximation Spaces and Choice of Basis Functions

Classical discontinuous Galerkin (DG) methods are broadly concerned with obtaining approximate representations of Lebesgue regular functions, e.g.  $v \in L^q(\Omega_i)$ . The most common way of achieving this is relative to the finite dimensional space of piecewise polynomial functions  $S_h^p$  over  $\Omega$  restricted to the local elements of  $\mathcal{T}_h$ ; formally defined by

$$S_h^p(\Omega_h, \mathcal{T}_h) = \{v : v|_{\Omega_i} \in \mathcal{P}^p(\Omega_i) \quad \forall \Omega_i \in \mathcal{T}_h\}^1.$$

Here  $\mathcal{P}^p(\Omega_i)$  denotes the space of degree (at most)  $p$  polynomials over  $\Omega_i$ .

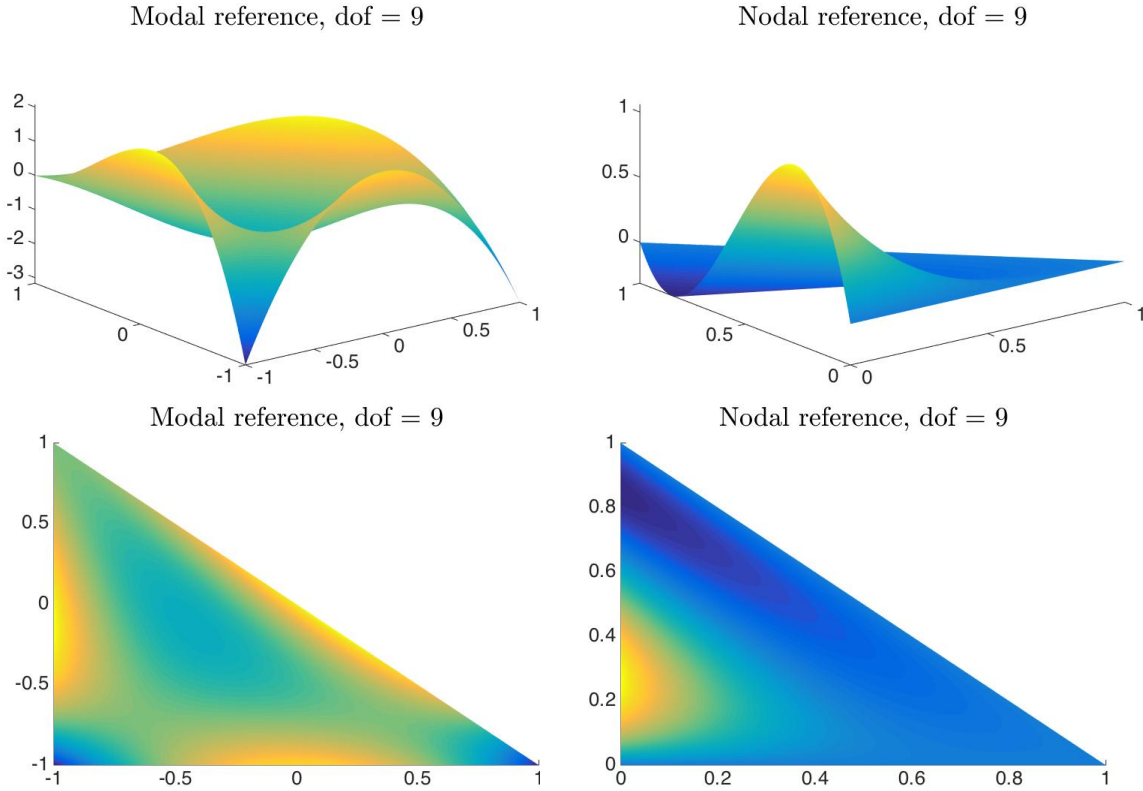


Figure 4: At  $p = 3$ , on the left is the ninth modal degree of freedom of the PKDO basis introduced by Proriol [51] and rediscovered by Koornwinder [37], Dubiner [17], and Owens [49], and on the right is the ninth degree of freedom in the nodal Lagrange basis as discussed in [66]

The DG basis can be cast relative to either modal or nodal basis functions. First consider the modal polynomial basis  $\tilde{N}_l$  indexed by  $l = 1, \dots, m_p$  the degrees of freedom prescribed by the local polynomial degree  $p$ . The approximate local representation of  $\zeta$  by  $\zeta_h^i$  over each finite element cell

<sup>1</sup>Note that DG methods are not required to be defined relative to the span of local polynomials. For example, an interesting candidate in the present context would be to define the space relative to the span of local rational functions.

$\Omega_i$  becomes:

$$\zeta_h^i(\mathbf{x}, t) = \sum_{l=1}^{m_p} \tilde{\zeta}_l^i(t) \tilde{N}_l^i(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_i. \quad (2.1)$$

Here  $l$  denotes the modal index, and  $\tilde{\zeta}_l^i(t)$  the modal unknowns.

In contrast to this modal form (2.1), an alternative way of representing  $\zeta$  is by way of the nodal representation, which is related by:

$$\zeta_h^i(\mathbf{x}, t) = \sum_{l=1}^{m_p} \tilde{\zeta}_l^i(t) \tilde{N}_l^i(\mathbf{x}) = \sum_{l=1}^{n_p} \zeta_l^i(\mathbf{x}_l^i, t) N_l^i(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_i, \quad (2.2)$$

where  $n_p$  denote the nodal degrees of freedom,  $N_l^i$  the nodal basis, and the nodal unknowns  $\zeta_l^i(\mathbf{x}_l^i, t)$ .

Moreover, transforming between the nodal and modal representations is now immediate, and can be made readily explicit by the associated Vandermonde matrix:  $\tilde{\mathbf{N}} = \mathbf{V}^\top \mathbf{N}$ , where  $\mathbf{N} = (N_1, \dots, N_{n_p})^\top$ ,  $\tilde{\mathbf{N}} = (\tilde{N}_1, \dots, \tilde{N}_{m_p})^\top$ , and  $\mathbf{V}$  is the elementwise Vandermonde matrix with entries,

$$V_{ij} = \tilde{N}_i(\mathbf{x}_j) \quad \text{for } i = 1, \dots, m_p, \text{ and } j = 1, \dots, n_p. \quad (2.3)$$

In figure 4 the modal and nodal basis function are shown over a linear reference element.

### iii. Application to Acoustics

As an example system, consider the first-order acoustic wave equations:

$$\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = 0, \quad \frac{\partial \mathbf{u}}{\partial t} + \nabla p = 0, \quad (2.4)$$

where  $\mathbf{u} = (u_x, u_y)$  is the velocity, and  $p$  the pressure.

Recasting this equation into the strong form DG can be accomplished by multiplying (2.4) by the  $l$ th degree of freedom of the test function  $\varrho_l^i$ , and integrating by parts twice (see [30]):

$$\begin{aligned} \int_{\Omega_i} \varrho_l^i \frac{\partial p}{\partial t} d\mathbf{x} &= - \int_{\Omega_i} \varrho_l^i \nabla \cdot \mathbf{u} d\mathbf{x} + \int_{\partial\Omega_i} \mathbf{n} \cdot (\mathbf{u}^i - \mathbf{u}^{i*}) \varrho_l^i d\mathbf{x}, \\ \int_{\Omega_i} \varrho_l^i \frac{\partial \mathbf{u}}{\partial t} d\mathbf{x} &= \int_{\Omega_i} \varrho_l^i \nabla p d\mathbf{x} - \int_{\partial\Omega_i} \mathbf{n} (p^i - p^{i*}) \varrho_l^i d\mathbf{x}. \end{aligned} \quad (2.5)$$

The numerical fluxes can be determined for the linear hyperbolic system by solving the Riemann problem by way of the Rankine-Hugoniot conditions [61], yielding the following numerical fluxes:

$$\mathbf{n} \cdot (\mathbf{u}^i - \mathbf{u}^{i*}) = \alpha[p] - \llbracket \mathbf{u} \rrbracket, \quad \mathbf{n} (p^i - p^{i*}) = (\alpha \llbracket \mathbf{u} \rrbracket - [p]) \mathbf{n}. \quad (2.6)$$

### iv. Application to Electromagnetics

An additional example in electromagnetics is the normalized (unit-free) two dimensional transverse magnetic (TM) form of Maxwell's equations:

$$\frac{\partial B_x}{\partial t} = - \frac{\partial E_z}{\partial y}, \quad \frac{\partial B_y}{\partial t} = \frac{\partial E_z}{\partial x}, \quad \frac{\partial E_z}{\partial t} = \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}. \quad (2.7)$$



Here  $B_x$  and  $B_y$  are the two normalized components of the magnetic field  $\mathbf{B} = (B_x, B_y)$ ,  $E_z$  is the normalized electric field, and the variables  $\mathbf{x}$  and  $t$  are unit-free scalings of space and time.

Again the nodal DG form is arrived at by multiplying (2.7) by a test function  $\zeta_\ell^i$ , and integrating by parts twice to recover the strong DG form:

$$\begin{aligned}
 \int_{\Omega_i} \zeta_\ell^i \frac{\partial B_x}{\partial t} d\mathbf{x} &= - \int_{\Omega_i} \zeta_\ell^i \frac{\partial E_z}{\partial y} d\mathbf{x} + \int_{\partial\Omega_i} n_y (E_z^i - E_z^{i*}) \zeta_\ell^i d\mathbf{x}, \\
 \int_{\Omega_i} \zeta_\ell^i \frac{\partial B_y}{\partial t} d\mathbf{x} &= \int_{\Omega_i} \zeta_\ell^i \frac{\partial E_z}{\partial x} d\mathbf{x} - \int_{\partial\Omega_i} n_x (E_z^i - E_z^{i*}) \zeta_\ell^i d\mathbf{x}, \\
 \int_{\Omega_i} \zeta_\ell^i \frac{\partial E_z}{\partial t} d\mathbf{x} &= \int_{\Omega_i} \zeta_\ell^i \frac{\partial B_y}{\partial x} d\mathbf{x} - \int_{\Omega_i} \zeta_\ell^i \frac{\partial B_x}{\partial y} d\mathbf{x} \\
 &\quad - \int_{\partial\Omega_i} n_x (B_y^i - B_y^{i*}) \zeta_\ell^i d\mathbf{x} + \int_{\partial\Omega_i} n_y (B_x^i - B_x^{i*}) \zeta_\ell^i d\mathbf{x}.
 \end{aligned} \tag{2.8}$$

The remaining numerical fluxes are then directly derived (e.g. see [30]) to satisfy

$$\begin{aligned}
 n_y (E_z^i - E_z^{i*}) &= \frac{1}{2} (n_y [E_z] + \alpha (n_x [\mathbf{B}] - [B_x])), \\
 n_x (E_z^i - E_z^{i*}) &= \frac{1}{2} (n_x [E_z] + \alpha (n_y [\mathbf{B}] - [B_y])), \\
 n_y (B_x^i - B_x^{i*}) - n_x (B_y^i - B_y^{i*}) &= \frac{1}{2} (n_y [B_x] - n_x [B_y] - \alpha [E_z]).
 \end{aligned} \tag{2.9}$$

### III. The Blended Isogeometric Discontinuous Galerkin Method

#### i. Computer Aided Geometric Design: B-splines, NURBS, and T-splines

In isogeometric analysis, the geometric basis employed in a given CAD parameterization is adopted as the basis for analysis. This section provides a review of the basic CAD technologies typically adopted in an isogeometric setting.

Traditionally B-splines (or basis splines) have been used to develop analysis suitable representations of curvilinear elements. These representations are constructed from knot vectors and the Cox-de Boor recursion formula to develop basis functions that parameterize B-spline curves. The drawback of B-splines is their inability to exactly parameterize important geometric objects, such as conic sections.

However, B-splines projected from higher dimensional space  $\mathbb{R}^{d+1}$  to  $\mathbb{R}^d$  can be mapped in such a way as to exactly recover a much larger class of geometric curves, including piecewise conic sections. Using projective weighted mappings one can generate *nonuniform rational B-splines* (NURBS), which preserve all the nice properties of B-spline basis functions while substantially extending the supported class of parameterized curves.

These NURBS representations now exhibit a number of very nice additional features, making them the standard used for representing complex geometries in the computer graphics and CAD communities. Moreover, NURBS basis functions can be refined by simple knot insertion, leading to the generation of augmented knot vectors that are easy to generalize. An extension of the NURBS refinement strategies is further provided by T-splines [54], that allow for T-junctions, effectively making even local refinements of the surface geometry possible. T-splines are also becoming more popular in CAD programs for providing an elegant way of encoding complicated geometries.

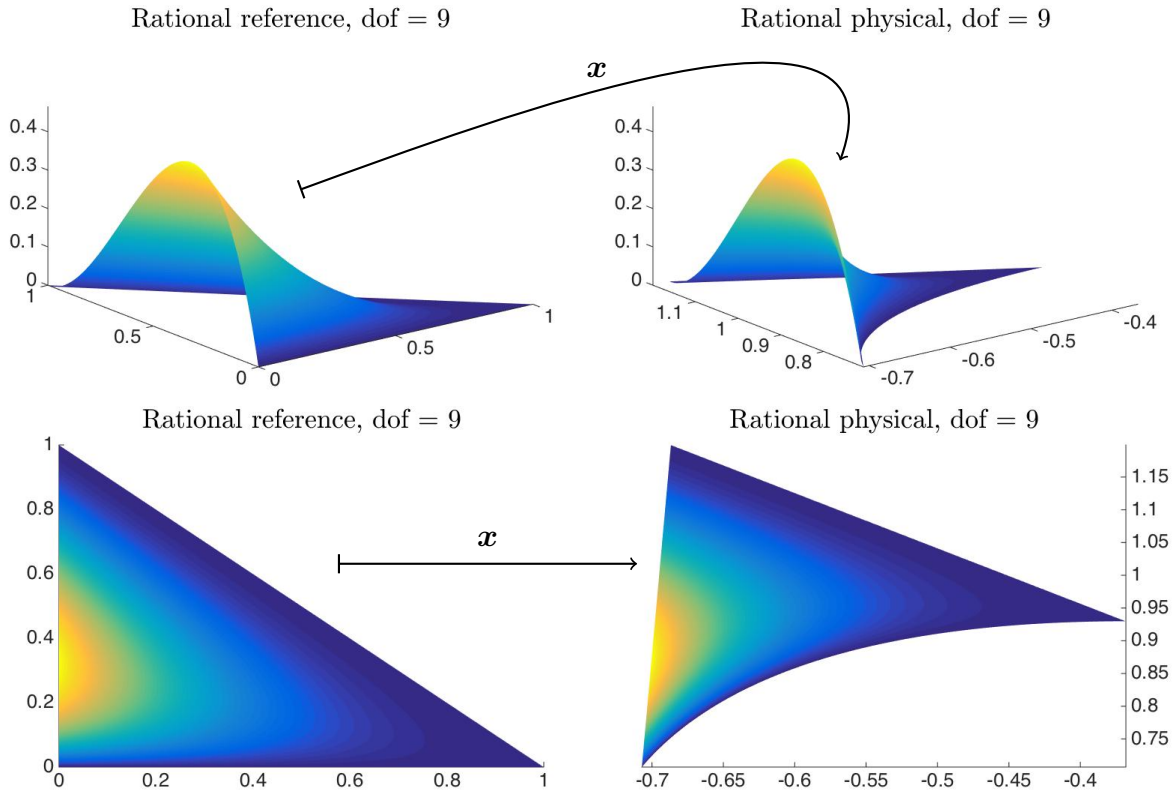


Figure 5: At  $\sigma = 3$ , on the left is the ninth degree of freedom of the rational Bézier basis evaluated on the master element  $\mathbf{M}$ , and on the right evaluated on the physical element  $\Omega_i$ .

Unfortunately, a problem that has served as a bottleneck in classical Isogeometric Analysis (IGA) has been the inability to automatically parameterize control volumes given the corresponding surface parameterization provided from the CAD mesh. Any admissible solution to this problem is required to preserve the exact geometry of the given surface parameterization inherited from the CAD, while simultaneously generating a mesh properly suited (e.g. conditioned) for immediate finite element analysis. Recent progress has identified the major limiting factor in achieving full surface-to-volume parameterizations to be the use of completely structured (i.e. quadrilateral and hexahedral elements) meshes; a problem that has now been addressed using meshes that include triangular and tetrahedral elements [19].

An important property of NURBS and T-splines in the context of isogeometric analysis is the ability to perform Bézier extraction [9]. Bézier extraction provides the capability of recovering a local Bernstein-Bézier representation of the geometry from the global NURBS or T-splines CAD. These Bernstein-Bézier representations are generally defined over rectangular elements, though recent progress now allows Bézier extraction to be extended to support unstructured isogeometric mesh generation [19], and enable automatic, geometrically exact tetrahedral mesh generation. The resulting local rational Bézier triangles, it turns out, provide all of the information needed to adapt

a classical DG method into a BIDG method.

## ii. Rational Bézier Representational Geometries

Because of Bézier extraction techniques [9], unstructured isogeometric meshes in two dimensions can be naturally defined in terms of rational Bézier triangles  $\mathbf{T}$ . Bézier triangles can be thought of as parameterized curves written in terms of the rational Bézier basis (i.e. shape) functions. These basis functions are quotients of classical Bernstein polynomials, where the Bernstein polynomials of degree  $\sigma$  are defined by,

$$B_{jkl}(\mathbb{\lambda}) = \frac{\sigma!}{j!k!l!} \lambda_1^j \lambda_2^k \lambda_3^l \quad (3.1)$$

written relative to the barycentric coordinate  $\mathbb{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ , with  $j, k$ , and  $l$  indices satisfying  $j + k + l = \sigma$  in each component.

The rational Bézier shape functions  $R_{jkl}$  can then be constructed as a quotient of Bernstein polynomials, with indexed weighting parameter  $w_{abc}$ , as

$$R_{jkl}(\mathbb{\lambda}) = \frac{B_{jkl}(\mathbb{\lambda})w_{jkl}}{\sum_{a+b+c=\sigma} B_{abc}(\mathbb{\lambda})w_{abc}} \quad (3.2)$$

for the indices  $j, k, l$  spanning zero to  $\sigma$ . The resulting Bézier triangle is then defined as:

$$\mathbf{T}(\mathbb{\lambda}) = \sum_{j+k+l=\sigma} R_{jkl}(\mathbb{\lambda}) \mathbf{P}_{jkl}, \quad (3.3)$$

where the  $\mathbf{P}_{jkl}$  are just the indexed control points of the control net.

These rational Bézier curves  $\mathbf{T}$  are able to represent conic sections exactly, and moreover retain the majority of the geometrically advantageous properties of Bézier curves, such as the end-point interpolation property, the prescribed tangent line property at end-points, the convex hull property (assuming the weights are positive), and so forth (see [23] for more details on Bézier curves).

In the context of the BIDG method, there are three spatial representations that are essential to “blending” IGA with DG. Recall that the discrete curvilinear physical domain  $\Omega_h$  refers to the physical domain or the CAD mesh of elements  $\Omega_i$ . Next, the standard DG approach computes on a reference (or master) element, denoted  $\mathbf{M}$ . And finally the barycentric element is denoted by  $\mathbf{B}$ , and will eventually serve as an intermediary between the reference and physical frames. For clarity, the coordinates of these elements are denoted by the physical coordinates  $\mathbf{x} \in \Omega_i$ , the reference coordinates  $\boldsymbol{\xi} \in \mathbf{M}$ , and the barycentric coordinates  $\mathbb{\lambda} \in \mathbf{B}$ , provided that  $\mathbf{x} = (x, y)$ ,  $\boldsymbol{\xi} = (\xi, \eta)$ , and  $\mathbb{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ .

The coordinate transformation from the reference to barycentric frame is performed using the standard affine mapping alongside the barycentric constraint,

$$\lambda_1 \xi_1 + \lambda_2 \xi_2 + \lambda_3 \xi_3 = \xi, \quad \lambda_1 \eta_1 + \lambda_2 \eta_2 + \lambda_3 \eta_3 = \eta, \quad \text{with constraint} \quad \lambda_1 + \lambda_2 + \lambda_3 = 1,$$

which reduces to solving:

$$\begin{bmatrix} \xi_1 & \xi_2 & \xi_3 \\ \eta_1 & \eta_2 & \eta_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} \xi \\ \eta \\ 1 \end{bmatrix}. \quad (3.4)$$

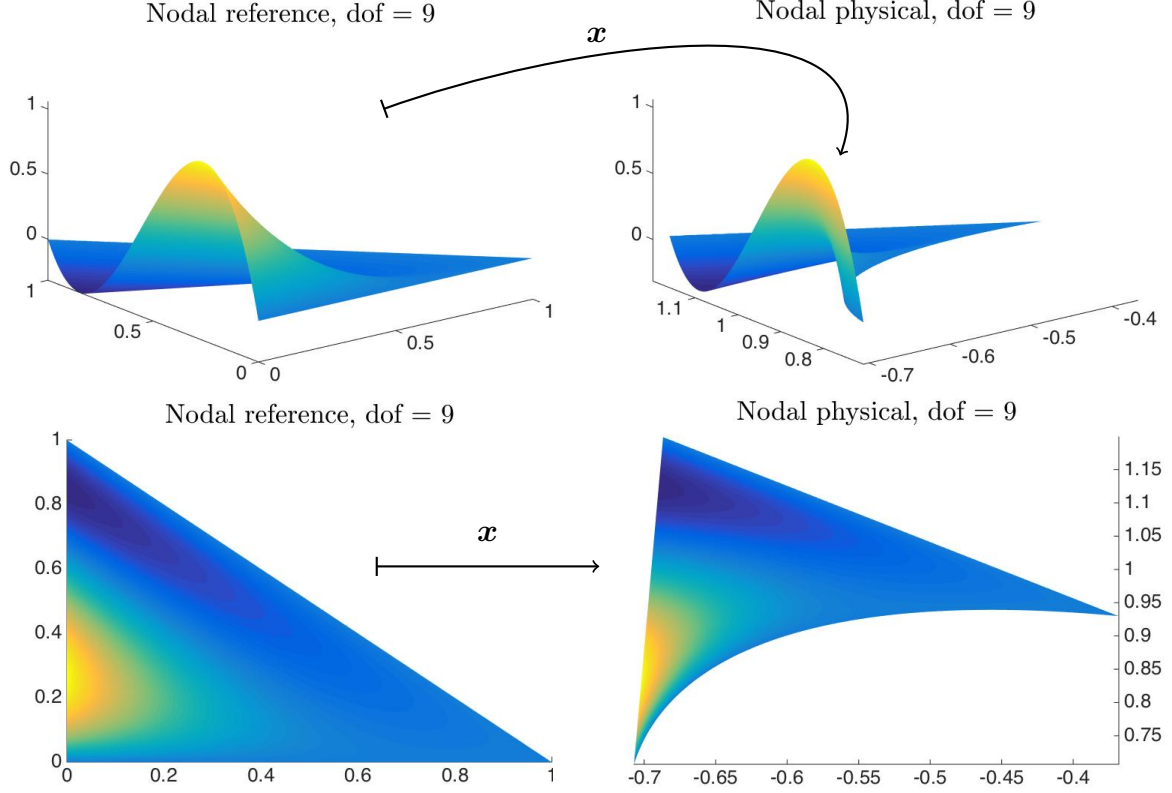


Figure 6: At  $p = 3$ , on the left is the ninth basis function of the nodal Lagrange basis on the reference triangle, and on the right is the ninth basis function in the nodal Lagrange basis on the curvilinear physical element.

Given this transformation it becomes clear that the barycentric coordinates can be viewed as a function of the reference coordinates,  $\lambda(\boldsymbol{\xi})$ .

Similarly, a Bézier triangle  $\mathbf{T}$  from (3.3) is just a reparameterization of the physical coordinate  $\mathbf{x}$ . As a consequence, it is possible to rewrite (3.3) relative to the physical coordinates as:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{j+k+l=\sigma} R_{jkl}(\lambda(\boldsymbol{\xi})) \mathbf{x}_{jkl}. \quad (3.5)$$

This mapping provides a transformation between physical and reference frames as shown in figure 5. In addition 3.5 now fully recovers the pushforward mapping connecting  $\mathbf{M}$  and  $\Omega_i$ , since by the chain rule,

$$\frac{\partial \mathbf{x}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \sum_{j+k+l=\sigma} \frac{\partial R_{jkl}(\lambda)}{\partial \boldsymbol{\xi}} \mathbf{x}_{jkl} = \sum_{j+k+l=\sigma} \frac{\partial R_{jkl}}{\partial \lambda} \frac{\partial \lambda}{\partial \boldsymbol{\xi}} \mathbf{x}_{jkl}, \quad (3.6)$$

where  $\partial \lambda / \partial \boldsymbol{\xi}$  is the Jacobian matrix computed directly from (3.4), and the  $\partial R_{jkl} / \partial \lambda$  are directly recovered from (3.1) and (3.2).

### iii. Blending Algorithms and Approximation Spaces

With the curvilinear representations from subsection 3.2 providing exact geometric mappings between the coordinates of the reference and physical frames, it is now possible to support exact mesh geometry in classical DG methods. In order to seamlessly incorporate the geometry with the classical DG solution however, blending techniques must be employed. These blending processes are extremely light weight algorithmically, ultimately allowing one to retain the basic machinery of their own DG algorithms, requiring only minor changes to the underlying support functions that draw from the additional geometric content of the mesh.

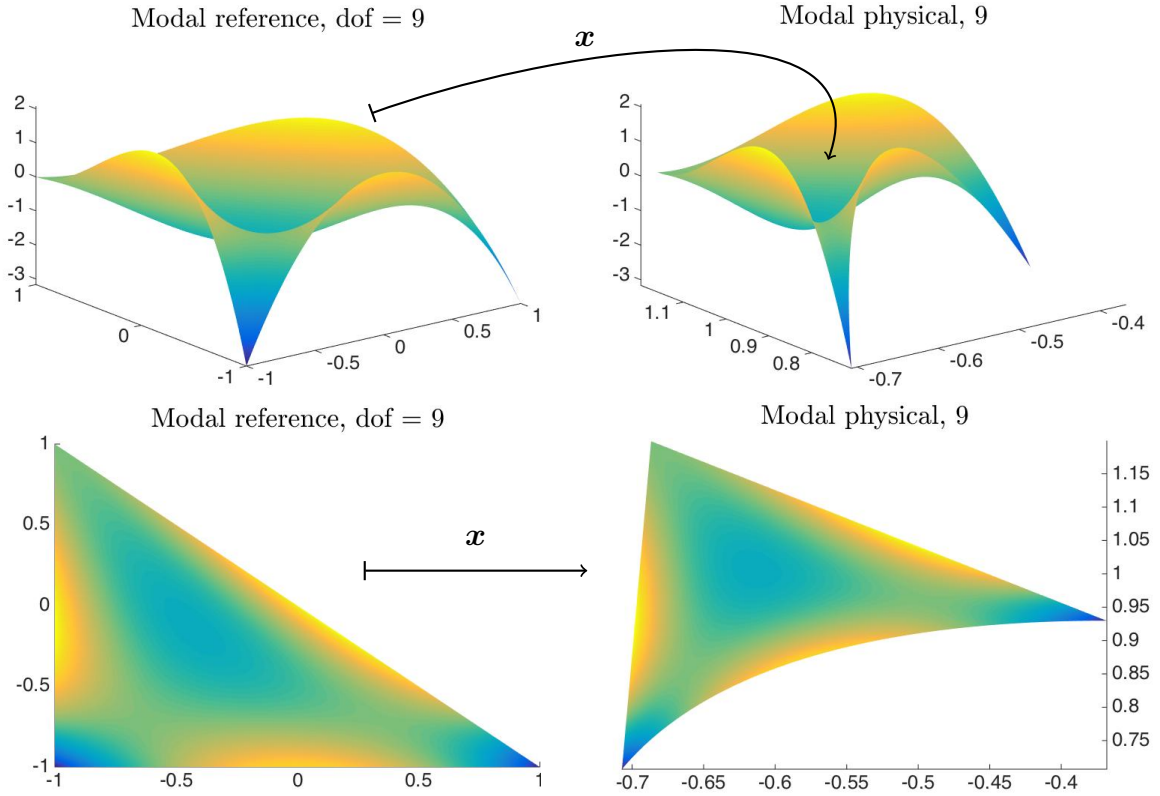


Figure 7: At  $p = 3$ , on the left is the ninth basis function of the modal normalized Jacobi basis on the reference element, and on the right is the ninth basis function in the nodal Lagrange basis on the physical element.

The blending algorithm primarily relies on the coordinate mapping (3.5) and (3.6), which now, in addition to transforming the geometric shape functions, also provides the coordinate transformation between the exact physical/geometric and reference elements for both the modal (see figure 7) and nodal (see figure 6) DG basis functions, as well as their first derivatives (all that is required in a standard mixed form DG approach [2], for example).

These rational maps (3.5) and (3.6) have a tremendous simplifying effect on the algorithmic design, allowing for classical DG algorithms to be performed in the usual way relative to a linear

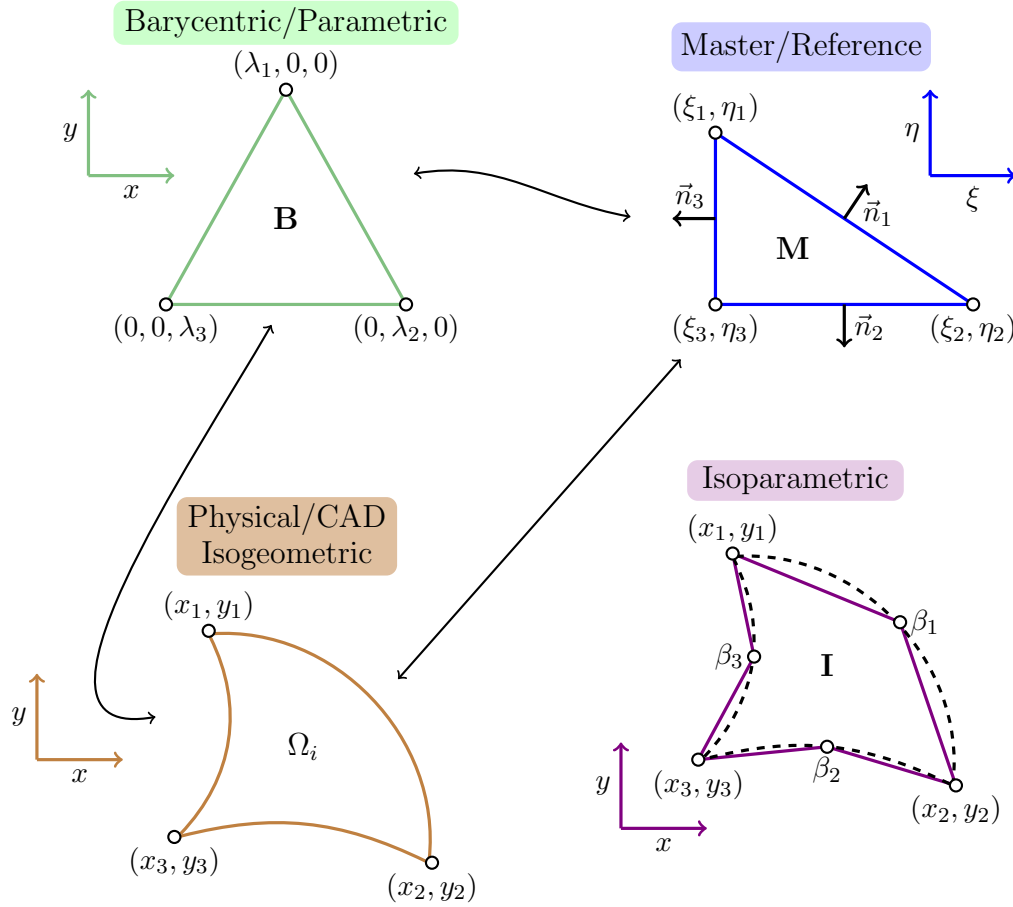


Figure 8: The linear reference element  $\mathbf{M}$ , the linear barycentric element  $\mathbf{B}$ , the physical/CAD element  $\Omega_i$ , can be seamlessly transformed for the BIDG method. This is in contrast to a classical isoparametric triangle  $\mathbf{I}$  supporting locally approximate geometry (e.g. by way of blend and warp techniques, etc. [30] and/or deformation inversions [50]).

reference element, even when computing over curvilinear/geometrically nontrivial meshes. In fact, this can now be performed using any DG basis one might like (e.g. a nodal basis, a modal basis, or some mixture of the two, and so on). The triangulation  $\mathcal{T}_h = \{\Omega_1, \Omega_2, \dots, \Omega_N\}$  is now some set of exact elements that are either linear or isogeometric, and the BIDG solution is computed (as before) relative to  $S_h^p$ , with the only additional feature being a composition due to the reparameterization with respect to the approximation space:

$$S_h^p(\Omega_h, \mathcal{T}_h) = \{v : v(\mathbf{x}(\boldsymbol{\xi}))|_{\mathbf{M}} \in \mathcal{P}^p \quad \forall \Omega_i \in \mathcal{T}_h\}.$$

This means that we recover nearly all of the nice approximation properties of classical DG methods, while simultaneously expanding support for exact curvilinear geometry.

In the appendix we show a technical outline of how the standard DG estimates apply to the BIDG method, and can be naturally recovered. The one subtlety that arises has to do with

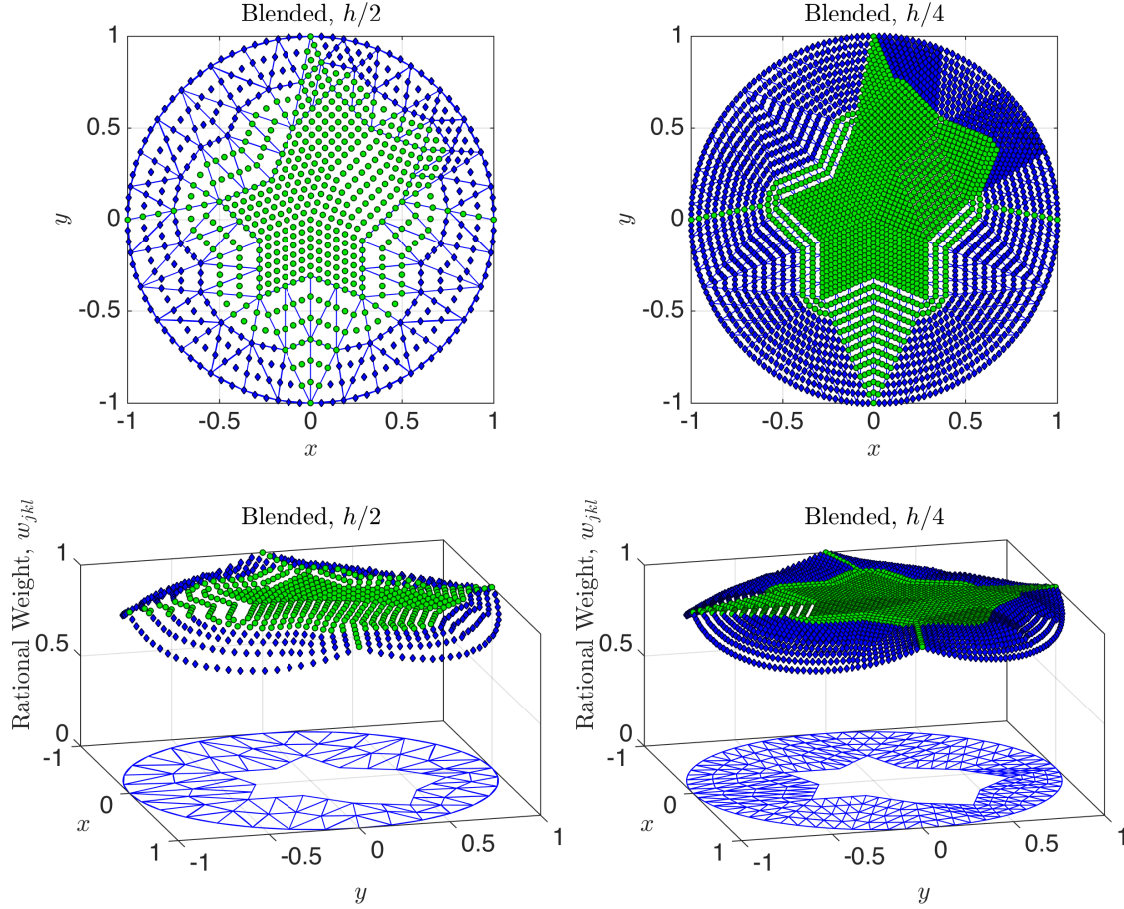


Figure 9: The BIDG blending algorithm using a standard Bézier refinement strategy. The control net is shown where  $\blacklozenge$  indicates weights less than one, and  $\bullet$  indicates control weights exactly equal to one. On the left is the  $h/2$  mesh, with 168 elements, 112 blended. On the right is the  $h/4$  refinement, with 672 total elements, 448 blended.

automating the generation of BIDG-suitable meshes. As discussed in detail in [19], automating the generation of appropriately conditioned Bézier meshes requires the utilization of Bézier extraction techniques from CAD T-splines and NURBS. Generally these meshes are generated at the coarsest level of refinement. The important ramification this has on the BIDG method comes at the level of adaptive mesh refinement. When refining an isogeometric rational Bézier mesh, it is essential to preserve a relative constraint on the variation of the control weights and/or control points relative to  $h$  in order to fully recover the approximation properties of the method (see the appendix for details). The practical impact this has, is that adaptive mesh refinement can naturally lead to the development of “blended layers” of curvilinear elements, which, in the continuum limit  $h \rightarrow 0$  tend to locally linear approximations. The existence and presence of these blended layers is important during mesh generation, and is explained in more detail in [19], and revisited in section 4 with a

numerical demonstration. To illustrate this behavior as it arises in the present context, observe figure 9 where the formation of these blended layers is apparent.

Finally, it is important to distinguish isogeometric DG from classical curvilinear DG methods. Classically, *superparametric* methods indicate that the local top degree of the shape function  $\sigma$  exceeds the local top degree of the test function  $p$ , *subparametric* methods imply  $p > \sigma$ , and  $\sigma = p$  indicate *isoparametric* methods. However, BIDG being an isogeometric method, the relationship between  $p$  and  $\sigma$  is more nuanced. In fact, if the control weights for a given Bézier element are not identical, then the given basis of test functions cannot replicate the basis of geometric basis functions for any given polynomial degree  $p$ . Consequently, BIDG is typically a subparametric method for any given degree  $p$ . However, in the unique setting of identical control weights, BIDG is a subparametric method for  $p > \sigma$ , an isoparametric method for  $p = \sigma$ , and a superparametric method for  $p < \sigma$ , just as in the classical curvilinear DG method. Nevertheless, all three settings in the BIDG method preserve exact geometry, which contrasts to classical subparametric, isoparametric, and superparametric methods as visualized in figure 8.

#### iv. Implementation and Incorporation

After providing a BIDG-suitable isogeometric rational Bézier mesh, the ability to easily implement and/or incorporate the BIDG algorithm into existing DG codes is remarkably straightforward. The basic transformations from section 3.2 provide the essential information. First the map (3.5) is used to transform coordinates, support points, and/or cubature points between the physical and reference meshes. Then the Jacobian mapping (3.6) is used to recover the remaining geometric factors of the transformation.

##### a. Application to Acoustics

To demonstrate the ease of implementing the BIDG algorithm, take the standard semidiscrete form of the first term of (2.5) from section 2.4, and consider the left hand side transformed to the reference element space,

$$\frac{d}{dt} \int_{\Omega_i} \varrho_\ell^i(\mathbf{x}) p_h(\mathbf{x}) d\mathbf{x} = \frac{d}{dt} \int_{\mathbf{M}} \varrho_\ell^i(\boldsymbol{\xi}) p_h(\boldsymbol{\xi}) J^i d\boldsymbol{\xi},$$

where  $J^i$  is the Jacobian determinant computed from (3.6) as:

$$J^i = \left( \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right). \quad (3.7)$$

In the same way, setting  $\mathbf{u}_h = (u_h, v_h)$ , the first term on the right in (2.5) transforms as,<sup>2</sup>

$$\int_{\Omega_i} \varrho_\ell^i(\mathbf{x}) \nabla \cdot \mathbf{u}_h d\mathbf{x} = \int_{\mathbf{M}} \varrho_\ell^i(\boldsymbol{\xi}) \left( u_j^i(t) \left( \frac{\partial \varrho_j^i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \varrho_j^i}{\partial \eta} \frac{\partial \eta}{\partial x} \right) + v_j^i(t) \left( \frac{\partial \varrho_j^i}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \varrho_j^i}{\partial \eta} \frac{\partial \eta}{\partial y} \right) \right) J^i d\boldsymbol{\xi}.$$

<sup>2</sup>Note that the  $\partial \boldsymbol{\xi} / \partial \mathbf{x}$  come directly from (3.6) due to the nice properties of the reparameterization.



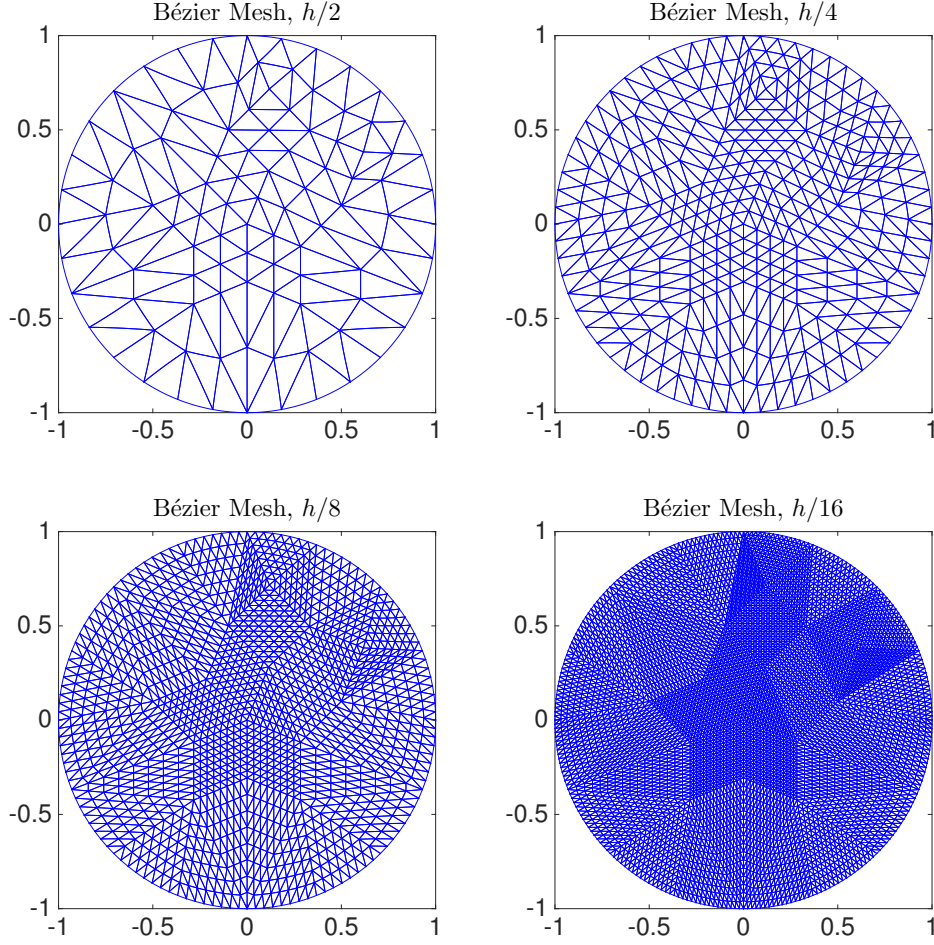


Figure 10: The exact isogeometric meshes for the unit circle, with refinements using uniform subdivision.

The remaining flux term easily follows then as:

$$\int_{\partial\Omega_i} (\alpha[p_h(\mathbf{x})] - \llbracket \mathbf{u}_h(\mathbf{x}) \rrbracket) \varrho_\ell^i(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{M}} (\alpha[p_h(\boldsymbol{\xi})] - \llbracket \mathbf{u}_h(\boldsymbol{\xi}) \rrbracket) \varrho_\ell^i(\boldsymbol{\xi}) J^i d\boldsymbol{\xi},$$

where the second equation in (2.5) can be expanded in the same way.

### b. Application to Electromagnetics

Restricting to the reference element is performed in essentially the same way, regardless of the application model. That is, the first term on the left of the first equation in (2.8) transforms as

$$\frac{d}{dt} \int_{\Omega_i} \zeta_\ell^i(\mathbf{x}) B_x^h(\mathbf{x}) d\mathbf{x} = \frac{d}{dt} \int_{\mathbf{M}} \zeta_\ell^i(\boldsymbol{\xi}) B_x^h(\boldsymbol{\xi}) J^i d\boldsymbol{\xi},$$

**Acoustic Wave Equation: Isoparametric Method (Warp & Blend [30])**

$p = 2$			$p = 3$			$p = 4$		
Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate
$h$	1.0421e-02	–	$h$	1.1549e-03	–	$h$	1.2979e-04	–
$h/2$	8.7315e-04	3.5771	$h/2$	6.9539e-05	4.0539	$h/2$	3.7843e-06	5.1000
$h/4$	8.9786e-05	3.2817	$h/4$	3.8486e-06	4.1754	$h/4$	1.1020e-07	5.1019

Table 1: Convergence behavior in the acoustic wave equation using the isoparametric approximate geometry with a nodal discontinuous Galerkin solution on the warped and blended analogues of the meshes shown in figure 10.

with the first term on the right

$$\int_{\Omega_i} \zeta_\ell^i(\mathbf{x}) E_{z,j}^i(t) \frac{\partial \zeta_j^i(\mathbf{x})}{\partial y} d\mathbf{x} = \int_{\mathbf{M}} \zeta_\ell^i(\boldsymbol{\xi}) E_{z,j}^i(t) \left( \frac{\partial \zeta_j^i}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \zeta_j^i}{\partial \eta} \frac{\partial \eta}{\partial y} \right) J^i d\boldsymbol{\xi},$$

and the flux

$$\begin{aligned} & \frac{1}{2} \int_{\partial\Omega_i} (n_y [E_z(\mathbf{x}) + \alpha(n_x \llbracket \mathbf{B}(\mathbf{x}) \rrbracket - [B_x(\mathbf{x})])] \zeta_\ell^i(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{2} \int_{\mathbf{M}} (n_y [E_z(\boldsymbol{\xi}) + \alpha(n_x \llbracket \mathbf{B}(\boldsymbol{\xi}) \rrbracket - [B_x(\boldsymbol{\xi})])] \zeta_\ell^i(\boldsymbol{\xi})) J^i d\boldsymbol{\xi}, \end{aligned}$$

where again the second and third equations follow directly.

## IV. Numerical Examples

### i. Application to Acoustics

The first example considered is the metallic membrane of a resonant drum solved using the acoustic wave equation (2.5), and given weakly imposed homogeneous boundary data

$$[p]|_{\partial\Omega_h} = 0, \quad \mathbf{u}|_{\partial\Omega_h} = 0. \quad (4.1)$$

#### a) Unit Radius Resonant Drum: Convergence Study

For the acoustic drum, the third resonant mode of the system is recovered from the Bessel beam solution of (2.5) over the unit cylinder in figure 10, given initial conditions

$$p|_{t=0} = J_0(\alpha_3 r) \cos(\alpha_3 t), \quad \mathbf{u}|_{t=0} = 0. \quad (4.2)$$

This solution is cast relative to cylindrical coordinates  $r = (x^2 + y^2)^{1/2}$ , with  $\alpha_3 = 8.653$  and  $J_0$  the Bessel function of the first kind.

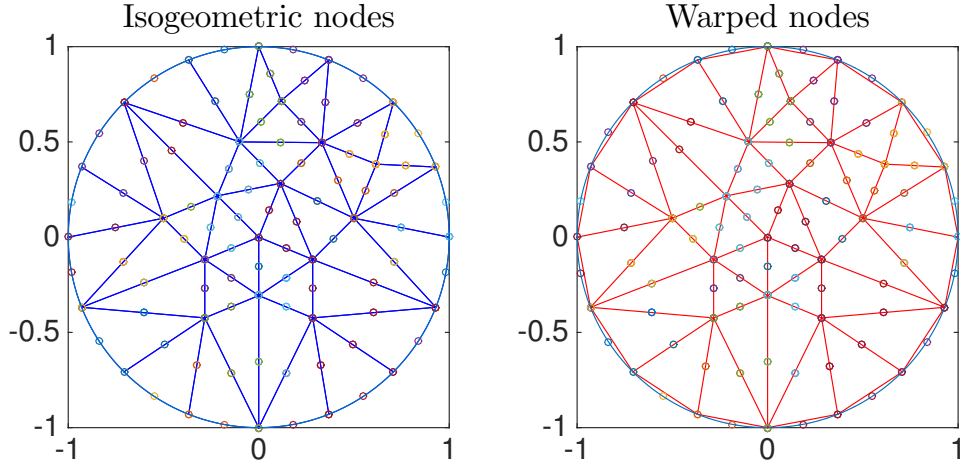


Figure 11: Distribution of nodal support points in the isogeometric Bézier mesh (left), versus the isoparametric blend and warp mesh that sends support points to lie on the true circle, but only preserves order  $p$  interpolative curves.

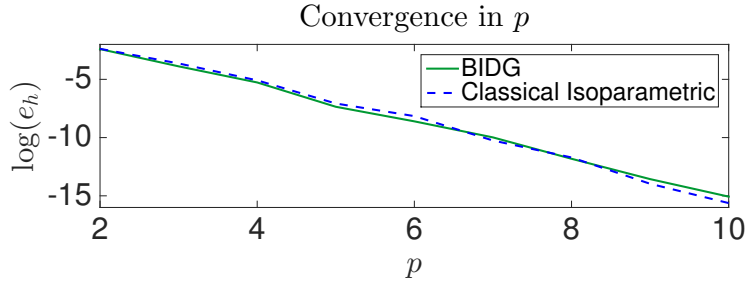


Figure 12: Convergence in BIDG versus classical isoparametric method as a function of degree  $p$ .

The convergence behavior relative to the isoparametric method for nodal DG is tested, using the blend and warp method for the unit cylinder [30], as shown in table 1. In table 3 the BIDG results can be compared. As expected, full convergence is observed in each case. In the warp and blend method, nodal support points are warped (i.e. node-snapped) to lie on the curved boundary of the mesh. Over this boundary layer, solutions are then fully integrated (viz. to exact cubature on order  $p$  isoparametric elements), however, the interpolation properties of the warp and blend method only preserve the order of the nodal basis  $p$  in the actual geometry, making it a classically isoparametric method. This behavior is shown in figure 11. The take-home message from figure 11 is that though the isoparametric approach provides a good analytic representation of the domain, it must be generated/preprocessed from a linear mesh at the outset, while the isogeometric mesh is already encoded before the computation begins, and can be done independent of the order  $p$  of the analysis basis.

The importance of the blending algorithm must also be emphasized. To illustrate this, meshes are constructed that do not implement a blended layer of elements, as shown in figure 13. In these

**Acoustic Wave Equation: Unblended IDG Method (Exact Geometry)**

$p = 2$			$p = 3$			$p = 4$		
Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate
$h$	9.7744e-03	–	$h$	2.0238e-03	–	$h$	1.6630e-04	–
$h/2$	2.1823e-03	2.1631	$h/2$	3.6218e-04	2.4823	$h/2$	3.9198e-05	2.0850
$h/4$	1.4992e-03	0.5416	$h/4$	3.8211e-05	3.2446	$h/4$	5.9905e-06	2.7100

Table 2: Convergence results for *unblended* IDG as compared with the appropriately blended algorithm in table 3.

**Acoustic Wave Equation: Blended Isogeometric DG Method (Exact Geometry)**

$p = 2$			$p = 3$			$p = 4$		
Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate
$h$	9.9746e-03	–	$h$	1.0853e-03	–	$h$	1.1696e-04	–
$h/2$	8.9056e-04	3.4855	$h/2$	6.2175e-05	4.1256	$h/2$	3.6692e-06	4.9944
$h/4$	9.4727e-05	3.2329	$h/4$	3.4280e-06	4.1809	$h/4$	1.1430e-07	5.0046

Table 3: Convergence results for BIDG as compared with the isoparametric results in table 1.

meshes, the control weights are allowed to transition within a single element, rather than over a smooth blended layer. The result of doing this leads to a dramatic reduction in the convergence rates of the method, as shown in table 2. This behavior is understood and fully expected from the underlying numerical theory, and is discussed and derived in detail in the appendix.

We additionally show the  $p$ -convergence behavior in figure 12. The most notable feature of the  $p$ -convergence graph is that in the BIDG method, the same mesh is used for every degree, while in the classical isoparametric method the physical domain must be remeshed each time the degree  $p$  is changed.

**b) Acoustic Resonator**

Finally, consider the same acoustic resonator from section 4.1.1, but solved over a more interesting geometric domain. The domain is still parameterized relative to the unit radius cylinder, though now with the addition of three additional circular “holes” into the interior of the domain. To avoid discontinuities in the initial state of the system, the following initial data is prescribed,

$$p|_{t=0} = \begin{cases} \sin(r\pi - 0.7)/0.3/10 & \text{for } r > 0.7 \\ 0 & \text{otherwise} \end{cases}, \quad \text{and} \quad \mathbf{u}|_{t=0} = 0,$$

with weakly imposed boundary (4.1) data as before, to elicit, for example, an idealized triple hang drum “gu” as pictured in figure 14.

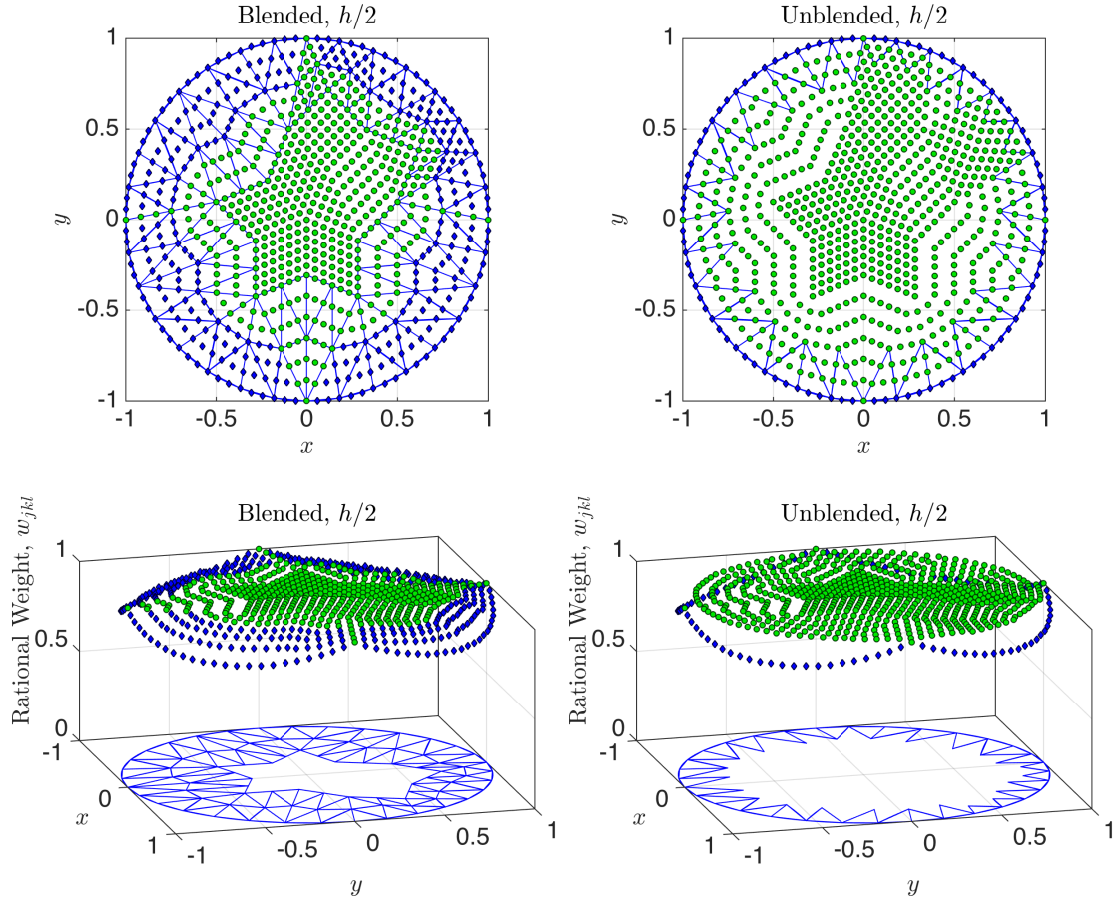


Figure 13: A blended versus an unblended mesh. The control net is shown where  $\blacklozenge$  indicates weights less than one, and  $\bullet$  indicates control weights exactly equal to one. Here we show two version of  $h/2$  mesh, with 168 elements. On the left the mesh is blended into a layer of 112, while on the right the control weights are allowed to transition within a single element.

## ii. Application to Electromagnetics

For our second numerical example, the transverse form of Maxwell's equations from section 2.4 is studied, specifying weakly imposed homogeneous boundary conditions

$$E_z|_{\partial\Omega_h} = 0, \quad [B_x]|_{\partial\Omega_h} = 0, \quad [B_y]|_{\partial\Omega_h} = 0, \quad (4.3)$$

with resonant solutions over the metallic cavity.

### a) Unit Radius Electromagnetic Cavity: Eigenspectrum Analysis

Consider the metallic resonant cavity shown in the meshes in figure 10, with initial conditions

$$B_x|_{t=0} = 0, \quad B_y|_{t=0} = 0, \quad E_z|_{t=0} = J_6(\alpha_6 r) \cos(6\theta) \cos(\alpha_6 t), \quad (4.4)$$

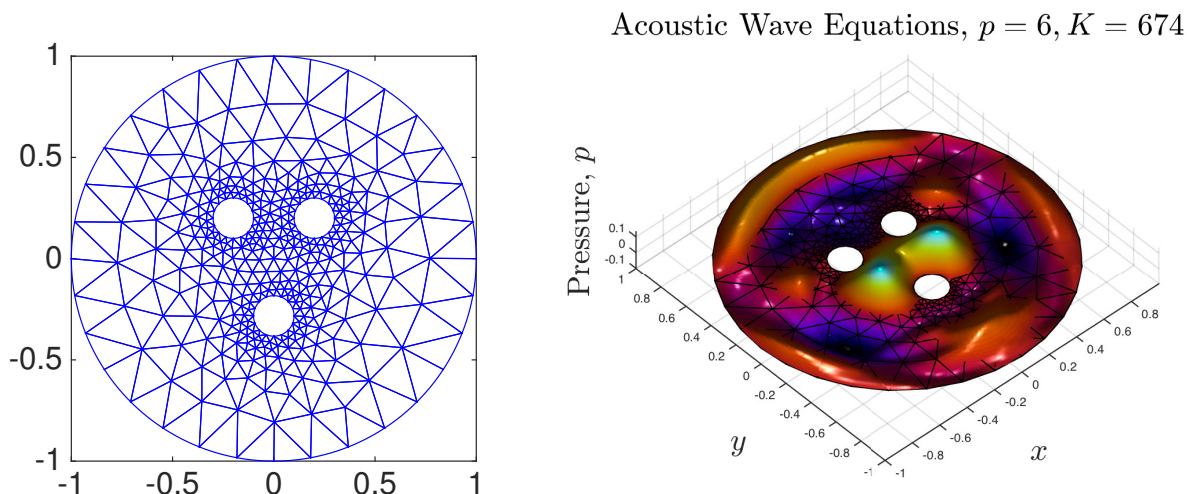


Figure 14: On the left is the isogeometric mesh, and on the right the BIDG solution to the third resonant mode propagating in the idealized triple hang drum “gu,” at  $p = 6$  after 2.5 resonant periods.

#### Maxwell’s Equations: Isoparametric Method (Warp & Blend [30])

$p = 2$			$p = 3$			$p = 4$		
Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate
$h$	8.4924e-03	–	$h$	1.1888e-03	–	$h$	1.7556e-04	–
$h/2$	8.8192e-04	3.2675	$h/2$	6.9461e-05	4.0972	$h/2$	4.3519e-06	5.3342
$h/4$	9.6328e-05	3.1946	$h/4$	4.3531e-06	3.9961	$h/4$	1.3075e-07	5.0567

Table 4: Convergence results for the isoparametric nodal discontinuous Galerkin solution on the warped and blended analogues of the meshes shown in the meshes of figure 10.

in cylindrical coordinates  $r = (x^2 + y^2)^{1/2}$ ,  $\theta = \text{atan2}(y, x)$ , with  $\alpha_6 = 13.589$ . Here  $J_6$  is the sixth Bessel function of the first kind, and  $E_z(t)$  is an exact resonant mode of the system.

Comparing results against the high-order accurate isoparametric solutions discussed in [30] utilizing the classical warp and blend technique again, provides the benchmark comparison study. Since the test case is linear, the algorithm is naturally optimized by integrating linear elements using exact quadrature-free methods, and the curvilinear elements using full cubature for both the BIDG and the isoparametric blend and warp method. As a qualitative stability metric, we compare the eigenspectrum of the spatial operator in the isoparametric method using warp and blend techniques, to that of the BIDG method. The comparisons indicate good stability features in the BIDG algorithm. Namely, the eigenspectrum shown in figure 15 display the expected stability properties to machine precision, which is that the eigenvalues remain non-positive along the real axis.

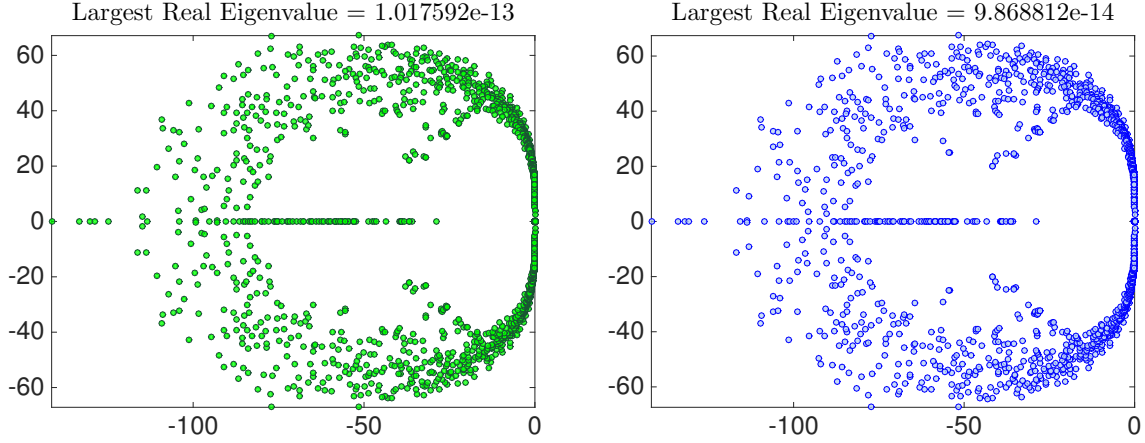


Figure 15: Eigenspectrum of the spatial operator of (2.8). On the left  $\bullet$  are BIDG eigenvalues, and on the right  $\circ$  are isoparametric warped eigenvalues, both at  $p = 4$  over the mesh from figure 11.

### Maxwell's Equations: Blended Isogeometric DG Method (Exact Geometry)

Superparametric, $p = 2$			Isoparametric, $p = 3$			Subparametric, $p = 4$		
Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate	Mesh	$L^2$ error	Rate
$h$	8.3660e-03	–	$h$	1.1632e-03	–	$h$	1.5384e-04	–
$h/2$	8.6698e-04	3.2705	$h/2$	6.5533e-05	4.1497	$h/2$	4.2802e-06	5.1676
$h/4$	9.4590e-05	3.1962	$h/4$	4.2065e-06	3.9615	$h/4$	1.2464e-07	5.1019

Table 5: Convergence results for the blended isogeometric discontinuous Galerkin (BIDG) solution on the meshes in figure 10.

### b) Unit Radius Electromagnetic Cavity: Convergence Study

Using the same initial boundary problem from section a, the convergence rates for isoparametric warp and blend technique is shown in table 4, and for the BIDG solution is shown in table 5. Again optimal rates are observed, where the BIDG solution is shown in figure 16, alongside the absolute error. This is remarkable to note, particularly since the BIDG solutions are able to exactly recover the geometry of the design mesh, while still maintaining the optimal high-order accurate behavior, and doing so while preserving the computationally efficient quadrature-free implementation on many elements.

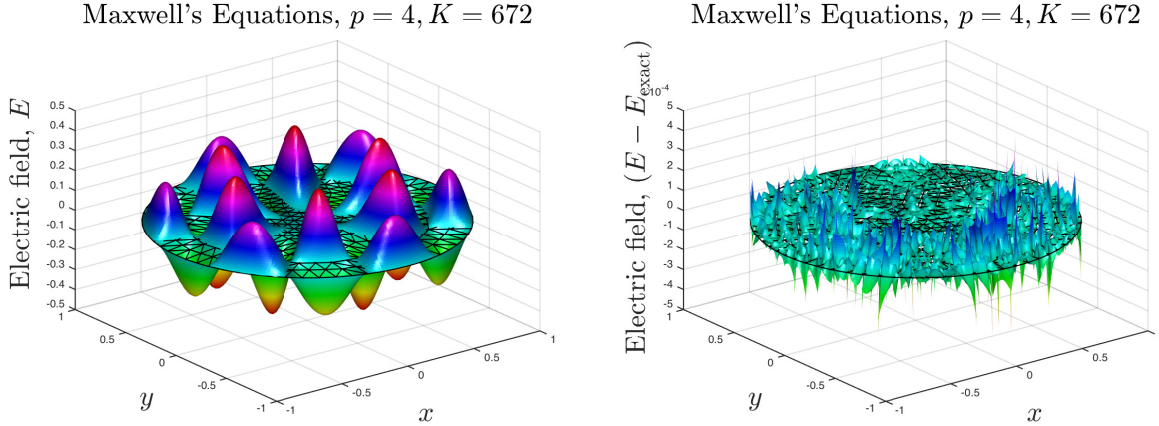


Figure 16: The left is the electric field  $E_z$  at endtime  $T = 1.0$  using the BIDG solution superimposed over the isogeometric mesh, while the right shows the absolute error  $(E_z - E_{z,\text{exact}})$  at the same timestep.

### c) Cavity Magnetron

Again, testing the geometric tolerances in the solution over the more complicated domain, the stator of a standard cavity magnetron is tested. The initial data is given by

$$E_z|_{t=0} = \begin{cases} \sin(r\pi - 9.2)/0.8/b & \text{for } r > 9.2 \\ 0 & \text{otherwise} \end{cases}, \quad \text{and} \quad B_x|_{t=0} = B_y|_{t=0} = 0,$$

with  $b = 200$ , and the weakly imposed boundary (4.3) data used. As shown in figure 17 the solutions run stable, preserving expected CFL constraints into high degrees of  $p$ . Note that while the blend and warp technique is also stable at  $p = 7$  for this example, albeit in the reduced order geometry, we fully expect that by preserving the exact geometric transforms along with the geometric continuity of the meshes, that BIDG solutions should demonstrate improved stability by reducing high frequency eigenmode pollution, as demonstrated exhaustively in works such as [20, 32, 33], and the references therein. We further conjecture that this expected behavior can be made rigorous and precise, in the form of sharp error estimates.

## V. Summary and Conclusions

In the paper a new computational method called the blended isogeometric discontinuous Galerkin (BIDG) method has been developed. The BIDG method enables: (1) seamless coupling to CAD mesh automation tools, (2) preservation of exact geometry of curvilinear domains, (3) preservation of optimal high-order accuracy, (4) easy incorporation into existing codes, and (5) easy incorporation of shape sensitivity and optimization tools.

This work has presented the basic approximation spaces needed for the BIDG method, and shown how to implement classical DG methods using an algorithm that blends the geometrically



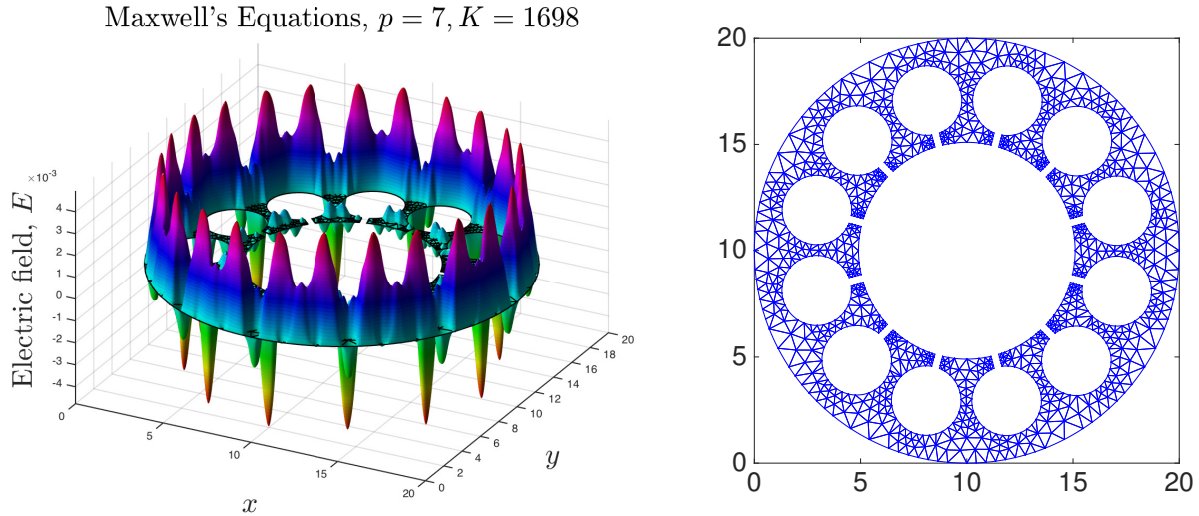


Figure 17: The left is the BIDG electric field  $E_z$  at  $p = 7$  with a blended layer of 1698 elements in the stator of a cavity magnetron at  $T = 20$ , and the right is the isogeometric mesh.

exact rational NURBS from isogeometric analysis into this regime. The basic transformation required to implement the BIDG method can be summarized as a reparameterization of the physical coordinate, along with the corresponding Jacobian derivative. Numerical tests have been run on both the acoustic wave equation, as well as a form of Maxwell's equations, showing optimal convergence results, along with favorable spectral behavior. In the appendix the reader will find additional results demonstrating how the underlying numerical theory can be provided to recover classical-type estimates.

Future efforts will be focused on extending the BIDG methods to include large-scale HPC examples in full dimension on substantially more complicated real-world geometries and nonlinear application models. A particularly attractive application problem for BIDG methods are those of shape-optimization studies over fluid-structure interaction (FSI) models, such as turbulence minimization in aerospace engineering and magnetically confined plasma.

## VI. Acknowledgements

C. Michoski and J.A. Evans were supported by the Air Force Office of Scientific Research under Grant No. FA9550-14-1-0113. L. Engvall was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1144469. J. Chan is supported partially by the Rice University CAAM Department Pfeiffer Postdoctoral Fellowship.

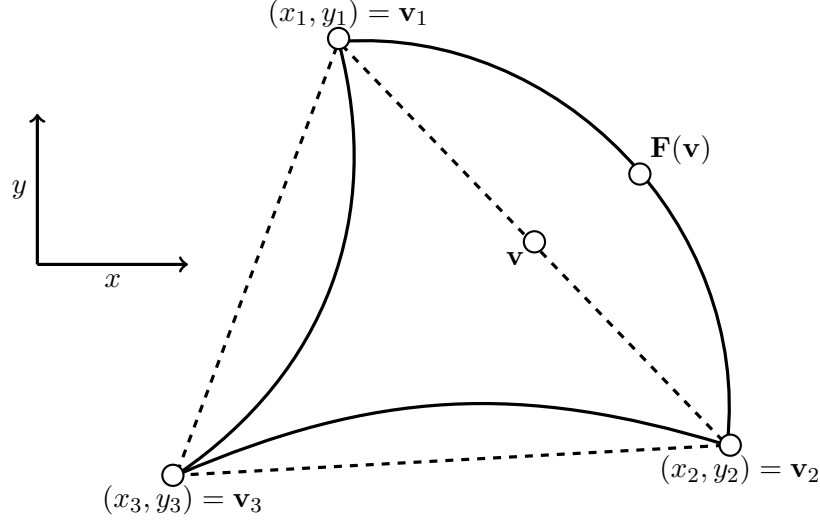


Figure 18: Here we show the warping function  $\mathbf{F}$  that takes points  $\mathbf{v}$  on a linear physical element  $\mathbf{V}$ , to  $\mathbf{F}(\mathbf{v})$  on a curvilinear physical element  $\Omega_i$ .

## VII. Appendix

In this appendix we present an *a priori* approximation estimate for the approximation spaces utilized in the blended isogeometric discontinuous Galerkin (BIDG) method. This estimate sharpens previously obtained results in the curvilinear discontinuous Galerkin and isogeometric communities [7, 67]. In particular, our estimate clearly delineates the effects of linear mesh regularity and the nonlinear mapping.

In this direction, let us introduce the notion of a warping function that characterizes the map from a physical linear triangle  $\mathbf{V}$  defined by the three vertex control points  $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$  to the physical coordinate triangle  $\Omega_i$ . More precisely, we denote  $\mathbf{F}(\mathbf{v}) = \mathbf{x}(\boldsymbol{\xi}(\boldsymbol{\lambda}(\mathbf{v})))$ , where  $\mathbf{v} \in \mathbf{V}$  and the barycentric coordinates are linear functions of  $\mathbf{v}$ , viz.  $\lambda_1 = \lambda_1(\mathbf{v})$ ,  $\lambda_2(\mathbf{v})$ , and  $\lambda_3(\mathbf{v})$  and figure 18. The mappings  $\mathbf{x} : \mathbf{M} \mapsto \Omega_i$  and  $\boldsymbol{\xi} : \mathbf{B} \mapsto \mathbf{M}$  are defined as in (3.4) and (3.5), respectively.

Given the concept of a warping function, we have the following result:

**Theorem 7.1** (Extended Bramble-Hilbert for curvilinear triangles). *Over each physical element  $\Omega_i \subseteq \Omega_h$ , there exists a constant  $C_{shape}$  independent of the mesh size  $h$  and the warping function  $\mathbf{F}$  such that for all  $u \in W^{j,2}(\Omega_i)$ , there exists a mapped polynomial  $u_h$  of degree  $p$  (i.e.,  $u_h(\mathbf{F}(\mathbf{v})) \in \mathcal{P}^p(\mathbf{V})$ ) satisfying the estimate:*

$$\|u - u_h\|_{L^2(\Omega_i)} \leq C_{shape} h^{p+1} \|\det \nabla \mathbf{F}^{1/2}\|_{L^\infty(\mathbf{V})} \|\det \nabla \mathbf{F}^{-1/2}\|_{L^\infty(\Omega_i)} \sum_{j=0}^{p+1} \alpha_{j,p+1}(\nabla \mathbf{F}) |u|_{W^{j,2}(\Omega_i)} \quad (7.1)$$

where  $\nabla \mathbf{F} = \nabla_{\mathbf{v}} \mathbf{F}$  and

$$\alpha_{j,m}(\nabla \mathbf{F}) = \sum_{\substack{i_1+i_2+\dots+i_m=j \\ i_1+2i_2+\dots+mi_m=m}} \|\nabla \mathbf{F}\|_{L^\infty(\Omega_i)}^{i_1} \|\nabla^2 \mathbf{F}\|_{L^\infty(\Omega_i)}^{i_2} \dots \|\nabla^m \mathbf{F}\|_{L^\infty(\Omega_i)}^{i_m}.$$

*Proof.* By definition we have,

$$\int_{\Omega_i} u^2 d\mathbf{x} = \int_{\mathbf{V}} (u \circ \mathbf{F})^2 \det \nabla \mathbf{F} d\mathbf{v}.$$

Therefore,  $\|u\|_{L^2(\Omega_i)} = \|\det \nabla \mathbf{F}^{1/2} u \circ \mathbf{F}\|_{L^2(\mathbf{V})}$ , and as a consequence:

$$\|u\|_{L^2(\Omega_i)} \leq \|\det \nabla \mathbf{F}^{1/2}\|_{L^\infty(\mathbf{V})} \|u \circ \mathbf{F}\|_{L^2(\mathbf{V})}.$$

Consequently, for each mapped polynomial  $u_h$  of degree  $p$ , it follows that:

$$\|u - u_h\|_{L^2(\Omega_i)} \leq \|\det \nabla \mathbf{F}^{1/2}\|_{L^\infty(\mathbf{V})} \|u \circ \mathbf{F} - u_h \circ \mathbf{F}\|_{L^2(\mathbf{V})}.$$

By the classical Bramble-Hilbert lemma [15], there exists a polynomial  $u_h \circ \mathbf{F}$  of degree  $p$  on the physical linear triangle satisfying:

$$\|u \circ \mathbf{F} - u_h \circ \mathbf{F}\|_{L^2(\mathbf{V})} \leq C_1 h^{p+1} |u \circ \mathbf{F}|_{W^{p+1,2}(\mathbf{V})}$$

where  $C_1$  is a constant that only depends on the polynomial degree  $p$  and the shape regularity of  $\mathbf{V}$  (i.e., the constant depends on the shape but not the size of  $\mathbf{V}$ ). We must now bound the seminorm  $|u \circ \mathbf{F}|_{W^{p+1,2}(\mathbf{V})}$  appearing in the above estimate by an analogous norm over the physical coordinate triangle  $\Omega_i$ . We may easily obtain control of the  $H^1$ -seminorm using the estimate:

$$\begin{aligned} |u \circ \mathbf{F}|_{H^1(\mathbf{V})} &= \left( \int_{\mathbf{V}} \nabla_{\mathbf{v}} (u \circ \mathbf{F}) \cdot \nabla_{\mathbf{v}} (u \circ \mathbf{F}) d\mathbf{v} \right)^{1/2} \\ &= \left( \int_{\Omega_i} \nabla_{\mathbf{v}} \mathbf{F} \nabla_{\mathbf{x}} u \cdot \nabla_{\mathbf{v}} \mathbf{F} \nabla_{\mathbf{x}} u \det \nabla \mathbf{F}^{-1} d\mathbf{x} \right)^{1/2} \\ &\leq \|\nabla \mathbf{F}\|_{L^\infty(\Omega_i)} \|\det \nabla \mathbf{F}^{-1/2}\|_{L^\infty(\Omega_i)} |u|_{H^1(\Omega_i)}. \end{aligned}$$

To obtain control of higher-order seminorms, we simply recurse on the previous estimate, as is done in [7, 14], resulting in the estimate:

$$|u \circ \mathbf{F}|_{W^{p+1,2}(\mathbf{V})} \leq C_2 \|\det \nabla \mathbf{F}\|_{L^\infty(\Omega_i)} \sum_{j=0}^{p+1} \alpha_{j,p+1}(\nabla \mathbf{F}) |u|_{W^{j,2}(\Omega_i)}$$

where  $C_2$  again is a constant that only depends on the polynomial degree  $p$  and the shape regularity of  $\mathbf{V}$ . We thus arrive with the optimal bound (7.1):

$$\|u - u_h\|_{L^2(\Omega_i)} \leq C_{shape} h^{p+1} \|\det \nabla \mathbf{F}^{1/2}\|_{L^\infty(\mathbf{V})} \|\det \nabla \mathbf{F}^{-1/2}\|_{L^\infty(\Omega_i)} \sum_{j=0}^{p+1} \alpha_{j,p+1}(\nabla \mathbf{F}) |u|_{W^{j,2}(\Omega_i)}$$

where  $C_{shape} = C_1 C_2$ .

□

The above result indicates that if  $\|\nabla^k \mathbf{F}\|_{L^\infty(\Omega_i)} = O(1)$  for  $k = 1, \dots, p+1$  and the determinant Jacobian is bounded from above and below, then the BIDG method exhibits a similar convergence rate to that seen for affine elements. It is easily shown that if nested refinements are performed, then  $\nabla \mathbf{F} \rightarrow \mathbb{I}$  in the limit of mesh refinement, and hence one may expect to see optimal convergence. That being said, it is instructive to examine the effect of mesh warping in the pre-asymptotic limit as well as in the setting of non-nested refinements. In these cases closer inspection of the Jacobian matrix,  $\nabla \mathbf{F} = \nabla_{\xi} \mathbf{x} \nabla_{\lambda} \xi \nabla_{v} \lambda$  is required.

Notice that the term  $\nabla_{v} \lambda$  gives a measure of the combined size and shape regularity of the physical linear element, and it is easily seen to be of order  $h^{-1} C_{shape}$  where  $C_{shape}$  is a constant which depends on the shape of the physical linear element but not its size. Meanwhile, the term  $\nabla_{\lambda} \xi$  is the same for all considered element types and is independent of the mesh size  $h$ . Therefore, to ensure that  $\|\nabla \mathbf{F}\| = O(1)$ , it must hold that  $\|\nabla_{\xi} \mathbf{x}\| = O(h)$ . The question then arises: how can one ensure that this equivalence holds?

To examine this question in detail, we expand the derivative of the parametric mapping. In what follows, let us denote the Bézier weighting function as

$$w = \sum_{i+j+k=\sigma} B_{ijk}(\lambda) w_{ijk}.$$

It follows that in each parametric direction:

$$\frac{\partial \mathbf{x}}{\partial \xi_i} = \left( \frac{1}{w} \frac{\partial (\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl})}{\partial \xi_i} \right) - \left( \frac{\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl}}{w^2} \frac{\partial w}{\partial \xi_i} \right).$$

By exploiting the derivative relationship for Bernstein polynomials, we can establish the following relationships for the left and right terms of the above expression:

$$\frac{\partial (\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl})}{\partial \xi_1} = \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k+1,l} \mathbf{P}_{j,k+1,l} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l})$$

$$\frac{\partial (\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl})}{\partial \xi_2} = \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k,l+1} \mathbf{P}_{j,k,l+1} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l})$$

$$\frac{\partial w}{\partial \xi_1} = \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k+1,l} - w_{j+1,k,l})$$

$$\frac{\partial w}{\partial \xi_2} = \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k,l+1} - w_{j+1,k,l})$$

Therefore, we have that:

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \xi_1} &= \left( \frac{1}{w} \right) \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k+1,l} \mathbf{P}_{j,k+1,l} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l}) \\ &\quad - \left( \frac{\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl}}{w^2} \right) \sum_{j+k+l=p-1} p B_{jkl} (w_{j,k+1,l} - w_{j+1,k,l}) \end{aligned}$$

and:

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \xi_2} &= \left(\frac{1}{w}\right) \sum_{j+k+l=p-1} p B_{jkl}(w_{j,k,l+1} \mathbf{P}_{j,k,l+1} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l}) \\ &\quad - \left(\frac{\sum B_{jkl}(\lambda) w_{jkl} \mathbf{P}_{jkl}}{w^2}\right) \sum_{j+k+l=p-1} p B_{jkl}(w_{j,k,l+1} - w_{j+1,k,l}). \end{aligned}$$

As Bernstein polynomials are bounded above by one and below by zero, we thus have that:

$$\left| \frac{\partial \mathbf{x}}{\partial \xi_1} \right| \lesssim \sum_{j+k+l=p-1} |w_{j,k+1,l} \mathbf{P}_{j,k+1,l} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l}| + \sum_{j+k+l=p-1} |w_{j,k+1,l} - w_{j+1,k,l}|$$

and:

$$\left| \frac{\partial \mathbf{x}}{\partial \xi_2} \right| \lesssim \sum_{j+k+l=p-1} |w_{j,k,l+1} \mathbf{P}_{j,k,l+1} - w_{j+1,k,l} \mathbf{P}_{j+1,k,l}| + \sum_{j+k+l=p-1} |w_{j,k,l+1} - w_{j+1,k,l}|.$$

Therefore, all that is needed to guarantee the property  $\|\nabla_{\xi} \mathbf{x}\| = O(h)$  is that the projective control points  $\mathcal{P}_{jkl} = \{w_{jkl} \mathbf{P}_{jkl}, w_{jkl}\}^T$  lie in a convex polyhedron of radius  $O(h)$ .

Similar analyses can be conducted to determine sufficient conditions to guarantee that higher-order derivatives satisfy  $\|\nabla^k \mathbf{F}\|_{L^\infty(\Omega_i)} = O(1)$  and that the determinant Jacobian is bounded above and below, but such analyses are beyond the scope of the current paper and will be the subject of a forthcoming publication.

**Remark:** Throughout the paper, we utilize the term “isogeometric” to describe our proposed discretization technology. We define a method to be isogeometric if the geometry emanating from a Computer Aided Design package is also used as the geometry for finite element analysis. The first instances of isogeometric analysis were based on the use of Non-Uniform Rational B-splines (NURBS) for both the geometry and analysis, though isogeometric discretization methods based on the use of subdivision surfaces, T-splines, hierarchical B-splines, and T-splines have also been shown to be very powerful in the computational analysis of complex systems. Herein, we rely on the use of rational Bernstein-Bézier surfaces, another commonly employed technology in Computer Aided Design, to parametrize the geometry, but to reduce the computational cost of our new method, we employ standard polynomial approximation spaces for the representation of solution fields. As we employ different function spaces to represent the geometric mapping and solution field, we refer to our methodology as “blended.” It should be remarked that the use of “blending” to match a given mesh to a geometrically complex boundary is not new and is indeed common in high-order finite element analysis. Many such methods, including the recently introduced NURBS-enhanced finite element method [55, 56], employ transfinite interpolation to obtain such a matching. However, BIDG differs from such methods in an important way — methods based on transfinite interpolation only employ a rational Bézier mapping at the domain boundary while BIDG employs rational Bézier mappings throughout the domain. While either approach is valid, we argue that the BIDG approach is easier to implement in existing DG codes. First of all, as each element in the domain is mapped in the same manner with BIDG, one only need implement shape function routines based on rational Bernstein basis functions in order to transform a classical DG code into a BIDG code. Second, by

avoiding the use of transfinite interpolation, standard cubature integration rules may be employed with BIDG. This is not the case with NURBS-enhanced finite element methods, where specialized shape function routines and integration rules must be introduced to treat elements touching the boundary, and the situation is particularly complicated, wherein several “corner” cases must be considered (i.e. does the given element have a node, edge, or face on the geometric boundary?).

## References

- [1] C. Altmann, A. Beck, A. Birkefeld, F. Hindenlang, M. Staudenmaier, G. Gassner, and C.-D. Munz. Discontinuous Galerkin for high performance computational fluid dynamics (hpcdg). In W. E. Nagel, D. B. Kröner, and M. M. Resch, editors, *High Performance Computing in Science and Engineering '11*, pages 277–288. Springer Berlin Heidelberg, 2012.
- [2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.
- [3] F. Auricchio, L. B. da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully “locking-free” isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):160–172, 2007.
- [4] D. Balagangadhar and S. Roy. Design sensitivity analysis and optimization of steady fluid-thermal systems. *Computer Methods in Applied Mechanics and Engineering*, 190(42):5465 – 5479, 2001.
- [5] T. Barth. *Simplified numerical methods for gasdynamic systems on triangulated domains*. PhD thesis, Stanford University, 1998.
- [6] F. Bassi, L. Botti, A. Colombo, and S. Rebay. Agglomeration based discontinuous Galerkin discretization of the Euler and Navier-Stokes equations. *Computers & Fluids*, 61(SI):77–85, May 30 2012.
- [7] Y. Bazilevs, L. Beirão DA Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.
- [8] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229–263, 2010. Computational Geometry and Analysis.
- [9] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. 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.
- [10] Y. Cheng, A. J. Christlieb, and X. Zhong. Energy-conserving discontinuous Galerkin methods for the Vlasov-Ampere system. *Journal of Computational Physics*, 256:630–655, Jan 1 2014.

- 
- [11] Y. Cheng, I. M. Gamba, F. Li, and P. J. Morrison. Discontinuous Galerkin Methods for the Vlasov-Maxwell Equations. *SIAM Journal on Numerical Analysis*, 52(2):1017–1049, 2014.
- [12] Y. Cheng, F. Li, J. Qiu, and L. Xu. Positivity-preserving DG and central DG methods for ideal MHD equations. *Journal of Computational Physics*, 238:255–280, Apr 1 2013.
- [13] J. Chun, M. Kim, H. Jung, and S. Hong. Shape optimization of electromagnetic devices using immune algorithm. *IEEE Transactions on Magnetics*, 33(2, 2):1876–1879, Mar 1997. 7th Biennial IEEE Conference on Electromagnetic Field Computation (IEEE CEFC 96), Okayama, Japan, Mar 18-20, 1996.
- [14] P. Ciarlet and P.-A. Raviart. Interpolation theory over curved elements, with applications to finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 1(2):217–249, 1972.
- [15] P. G. Ciarlet. *Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- [16] C. Dawson, E. Kubatko, J. Westerink, C. Trahan, C. Mirabito, C. Michoski, and N. Panda. Discontinuous Galerkin methods for modeling hurricane storm surge. *Advances in Water Resources*, 34(9):1165–1176, 2011. New Computational Methods and Software Tools.
- [17] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.
- [18] R. Duvigneau, M. Visonneau, and G. Deng. On the role played by turbulence closures in hull shape optimization at model and full scale. *Journal of Marine Science and Technology*, 8(1):11–25, 2003.
- [19] L. Engvall and J. Evans. TriGA: Generalization of Isogeometric Analysis to Unstructured Triangular and Tetrahedral Discretizations. *preprint*, 2015.
- [20] J. A. Evans and T. J. R. Hughes. Discrete spectrum analyses for various mixed discretizations of the Stokes eigenproblem. *Computational Mechanics*, 50(6, SI):667–674, Dec 2012.
- [21] H. Fahs. Improving accuracy of high-order discontinuous Galerkin method for time-domain electromagnetics on curvilinear domains. *International Journal of Computer Mathematics*, 88(10):2124–2153, 2011.
- [22] W. Fan and P. Qiao. Vibration-based Damage Identification Methods: A Review and Comparative Study. *Structural Health Monitoring-an International Journal*, 10(1):83–111, Jan 2011.
- [23] R. Farouki and T. Sakkalis. Pythagorean Hodographs. *IBM Journal of Research and Development*, 34(5):736–752, Sep 1990.
- [24] S. Ganapathysubramanian and N. Zabaras. Computational design of deformation processes for materials with ductile damage. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):147–183, 2003.

- [25] G. J. Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to sbp-sat finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013.
- [26] G. J. Gassner. A kinetic energy preserving nodal discontinuous Galerkin spectral element method. *International Journal for Numerical Methods in Fluids*, 76(1):28–50, Sep 10 2014.
- [27] C. Geuzaine, A. Johnen, J. Lambrechts, J.-F. Remacle, and T. Toulorge. The generation of valid curvilinear meshes. In N. Kroll, C. Hirsch, F. Bassi, C. Johnston, and K. Hillewaert, editors, *IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach*, volume 128 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 15–39. Springer International Publishing, 2015.
- [28] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, Sep 10 2009.
- [29] W. Guo, F. Li, and J. Qiu. Local-Structure-Preserving Discontinuous Galerkin Methods with Lax-Wendroff Type Time Discretizations for Hamilton-Jacobi Equations. *Journal of Scientific Computing*, 47(2):239–257, May 2011.
- [30] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. Algorithms, analysis, and applications.
- [31] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.
- [32] T. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method {NURBS}. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4104–4124, 2008.
- [33] T. J. Hughes, J. A. Evans, and A. Reali. Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *Computer Methods in Applied Mechanics and Engineering*, 272(0):290–320, 2014.
- [34] H. T. Huynh, Z. J. Wang, and P. E. Vincent. High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids. *Computers & Fluids*, 98(SI):209–220, Jul 2 2014.
- [35] A. Johnen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233(0):359–372, 2013.
- [36] A. Kloeckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, Nov 2009.



- 
- [37] T. Koornwinder. Two-variable analogues of the classical orthogonal polynomials. *Theory and Application of Special Functions*, pages 435–495, 1975.
- [38] L. Krivodonova and M. Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics*, 211(2):492–512, Jan 20 2006.
- [39] S. N. Labs. The CUBIT geometry and mesh generation toolkit, 2015.
- [40] U. Langer, A. Mantzaflaris, S. E. Moore, and I. Touloupoulos. Multipatch Discontinuous Galerkin Isogeometric Analysis. *preprint*, 2015.
- [41] U. Langer and S. E. Moore. Discontinuous Galerkin Isogeometric Analysis of Elliptic PDEs on Surfaces. *preprint*, 2015.
- [42] H. Liu and H. Yu. The Entropy Satisfying Discontinuous Galerkin Method for Fokker-Planck equations. *Journal of Scientific Computing*, 62(3):803–830, Mar 2015.
- [43] J. Loverich, A. Hakim, and U. Shumlak. A Discontinuous Galerkin Method for Ideal Two-Fluid Plasma Equations. *Communications in Computational Physics*, 9(2):240–268, Feb 2011.
- [44] E. Lund, H. Moller, and L. Jakobsen. Shape design optimization of stationary fluid-structure interaction problems with large displacements and turbulence. *Structural and Multidisciplinary Optimization*, 25(5-6):383–392, DEC 2003.
- [45] X. Luo, M. Shephard, L.-Q. Lee, C. Ng, and L. Ge. Curved mesh correction and adaptation tool to improve compass electromagnetic analyses. *Journal of Physics: Conference Series*, 125(1):012082, 2008.
- [46] C. Michoski, D. Meyerson, T. Isaac, and F. Waelbroeck. Discontinuous Galerkin methods for plasma physics in the scrape-off layer of tokamaks. *Journal of Computational Physics*, 274(0):898–919, 2014.
- [47] M. Moumnassi, S. Belouettar, E. Bechet, S. P. A. Bordas, D. Quoirin, and M. Potier-Ferry. Finite element analysis on implicitly defined domains: An accurate representation based on arbitrary parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):774–796, 2011.
- [48] E. Olbrant, C. D. Hauck, and M. Frank. A realizability-preserving discontinuous Galerkin method for the M1 model of radiative transfer. *Journal of Computational Physics*, 231(17):5612–5639, Jul 1 2012.
- [49] R. G. Owens. Spectral approximations on the triangle. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 454(1971):857–872, 1998.
- [50] P.-O. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL*, 2009.

- [51] J. Proriot. Sur une famille de polynomes a deux variables orthogonaux dans un triangle. *C. R. Acad. Sci. Paris*, pages 2459–2461, 1975.
- [52] R. F. Riesenfeld, R. Haimes, and E. Cohen. Initiating a CAD renaissance: Multidisciplinary analysis driven design: Framework for a new generation of advanced computational design, engineering and manufacturing environments. *Computer Methods in Applied Mechanics and Engineering*, 284(0):1054–1072, 2015. Isogeometric Analysis Special Issue.
- [53] U. Schramm and W. D. Pilkey. Structural shape optimization for the torsion problem using direct integration and b-splines. *Computer Methods in Applied Mechanics and Engineering*, 107(1–2):251 – 268, 1993.
- [54] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22(3):477–484, July 2003.
- [55] R. Sevilla, S. Fernández-Méndez, and A. Huerta. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76(1):56–83, 2008.
- [56] R. Sevilla, S. Fernández-Méndez, and A. Huerta. 3d nurbs-enhanced finite element method (nefem). *International Journal for Numerical Methods in Engineering*, 88(2):103–125, 2011.
- [57] A. S. Silveira, R. C. Moura, A. F. C. Silva, and M. A. Ortega. Higher-order surface treatment for discontinuous Galerkin methods with applications to aerodynamics. *International Journal for Numerical Methods in Fluids*, pages n/a–n/a, 2015.
- [58] G. A. Slipher and J. E. Hubbard, Jr. Exploitation of Higher-Order Membrane Modes for Improved Synthetic Jet Performance. *AIAA JOURNAL*, 47(6):1388–1407, Jun 2009. AIAA 46th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan 07-10, 2008.
- [59] B. Srinivasan, A. Hakim, and U. Shumlak. Numerical Methods for Two-Fluid Dispersive Fast MHD Phenomena. *Communications in Computational Physics*, 10(1):183–215, Jul 2011.
- [60] G. Strang and G. J. Fix. *An analysis of the finite element method*. Prentice-Hall series in automatic computation. Englewood Cliffs, N.J. Prentice-Hall, [cop.1973], 1973.
- [61] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics : a practical introduction*. Springer, Berlin, New York, 1997.
- [62] T. Toulorge and W. Desmet. Curved Boundary Treatments for the Discontinuous Galerkin Method Applied to Aeroacoustic Propagation. *AIAA Journal*, 48(2):479–489, Feb 2010. AIAA/CEAS 15th Aeroacoustics Conference, Miami, FL, May 11-13, 2009.
- [63] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *Journal of Computational Physics*, 254:8 – 26, 2013.
- [64] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2976–2988, 2008.

- 
- [65] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, Jul 20 2013.
- [66] T. Warburton. An explicit construction of interpolation nodes on the simplex. *Journal of Engineering Mathematics*, 56(3):247–262, 2006.
- [67] T. Warburton. A low-storage curvilinear discontinuous Galerkin method for wave problems. *SIAM Journal on Scientific Computing*, 35(4):A1987–A2012, 2013.
- [68] D. Wirasaet, S. Brus, C. Michoski, E. Kubatko, J. Westerink, and C. Dawson. Artificial boundary layers in the discontinuous Galerkin shallow water solution in channels. *preprint, preprint*, 2013.
- [69] Y. Xing. Exactly well-balanced discontinuous Galerkin methods for the shallow water equations with moving water equilibrium. *Journal of Computational Physics*, 257(A):536–553, Jan 15 2014.
- [70] Y. Xing, X. Zhang, and C.-W. Shu. Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. *Advances in Water Resources*, 33(12):1476–1493, 2010.
- [71] J. Xu and A. N. Chernikov. Automatic curvilinear quality mesh generation driven by smooth boundary and guaranteed fidelity. *Procedia Engineering*, 82(0):200–212, 2014. 23rd International Meshing Roundtable (IMR23).
- [72] F. Zhang, Y. Xu, and F. Chen. Discontinuous Galerkin Methods for Isogeometric Analysis for Elliptic Equations on Surfaces. *Communications in Mathematics and Statistics*, 2(3-4):431–461, 2014.