SAND2021-15094C

Conservative and Entropy-Stable Nonconformal Interfaces With Lower Accuracy Quadrature: Circumventing the Inner-Product Preservation Property

Jared Crean* and Travis Fisher†
Sandia National Laboratories, PO Box 5800, Albuquerque, NM 87185

The use of p-, and hp-nonconformal interfaces enables greater geometric flexibility in performing computational science simulations, especially when relying on efficient tensor-product-based high-order Summation-by-Parts element schemes. For high-speed compressible computational fluid dynamics, the underlying numerical method must be conservative such that the discretization of the governing equations satisfies the Rankine Hugoniot relations. This paper extends the conservative nonconformal interface method of [1] to Summation-by-Parts elements with face quadratures of degree less than $2p$, specifically allowing which allows the use of tensor-product elements on the Legendre-Gauss-Lobatto nodes, which are accurate up to degree $2p-1$. This formulation does not satisfy the inner-product preservation property of [1], but nonetheless remains conservative, entropy stable, and free-stream preserving. Mathematical theory is developed to determine the required accuracy of the mortar grid quadrature rule, and numerical results verify the mathematical results.

I. Introduction

In [1], a method is developed for discretizing both $p-$ and $hp-$nonconformal interfaces between summation-by-parts (SBP) operators that preserves the conservation, entropy-stability, and free-stream preservation properties of the discretization on either side of the interface. In that work, a mortar grid and associated quadrature rule were defined, and the nonconformal coupling term was computed on the mortar grid. The coupling term was required to satisfy an inner-product preservation property, namely that any inner product computed on the mortar grid would give the same (discrete) result if computed using the face quadrature rule of the SBP operator. Note that this property is similar to the property originally introduced by [2], which required inner product preservation between the SBP operators on either side of the interface.

The proof of the property in [1] required the SBP face quadrature rule to be exact for polynomials up to degree $2p$. This precludes the use of tensor-product SBP operators based on the Legendre-Gauss-Lobatto (LGL) nodes, which have a quadrature that is exact for only $2p-1$ polynomials. In this work, we show how to remove this limitation. Specifically, we develop:

- A p-nonconformal scheme for generalized elements, including SBP operators on simplex elements
  - This scheme loses one order of accuracy for elements with $2p-1$ quadrature, but has full accuracy for $2p$ accuracy quadratures
  - For $2p-1$ quadrature elements, the scheme does not satisfy the inner product preservation property of [1], yet remains stable and conservative.
- An hp-nonconformal scheme for generalized elements.
  - This approach removes a limitation of previous work [3] that requires the elements on one side of the interface to be subdivisions of the elements on the other side.
  - The accuracy properties are the same as the p-nonconformal scheme
  - For curved interfaces, there can be a discrete geometric incompatibility between the two sides of the interface. This incompatibility is shown to be high order accurate, and an algorithm is presented that computes a consistent set of normal vectors on the interface.

In Section II, we review important notation, the continuous partial differential equation, and curvilinear mappings. Section III introduces the p-nonconformal discretization and proves its theoretical properties. Section IV extends

*Postdoctoral Appointee, Sandia National Laboratories, PO Box 5800, Mail Stop 0828, Albuquerque, NM 87185-0828
†Principle Member of the Technical Staff, Sandia National Laboratories, PO Box 5800, Mail Stop 0828, Albuquerque, NM 87185-0828, AIAA Senior Member

the p-nonconformal discretization to hp-nonconformal interfaces and introduces algorithms to compute consistent normal vectors when there is discrete geometric nonconformity. Section V summarizes discretization requirements development in the previous two sections to obtain stability and accuracy. Numerical results are presented in Section VI and conclusions are given in Section VII.

II. Notations and Definition

Vectors will be denotes with bold font, for example, $\boldsymbol{u}$. Matrices will be denoted with sans-serif font, for example A. Subscripts for i, j, k, or any variable that is explicitly summed over, will denote indexing, for example $\boldsymbol{u}_i$ denotes the ith entry of the vector $\boldsymbol{u}$.

A. SBP Operator

Let $\mathbf{1}$ be a vector with every entry equal to 1. The length of the vector can be inferred from context. We require a Summation-by-Parts (SBP) operator that satisfies the following definition.

Definition 1 (SBP Operator)**.** For an element κ with simply-connected domain Ω_κ with boundary Γ_κ:

1) Differentiation operators D_x, D_y that satisfy, at minimum, $\mathsf{D}_x\mathbf{1} = \mathbf{0}$ and $\mathsf{D}_y\mathbf{1} = \mathbf{0}$
2) Symmetric boundary operators B_x, B_y that satisfy, at minimum, $\mathbf{1}^T\mathsf{B}_x\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^T\mathsf{B}_y\mathbf{1} = \mathbf{0}$.
3) A diagonal mass matrix P that forms a quadrature rule of degree $2p-1$, at least
4) Weak form integration matrices $\mathsf{Q}_x = \mathsf{P}\mathsf{D}_x$ and $\mathsf{Q}_y = \mathsf{P}\mathsf{D}_y$.
5) Skew-symmetric splitting $\mathsf{Q}_x = \mathsf{S}_x + \frac{1}{2}\mathsf{B}_x$ and $\mathsf{Q}_y = \mathsf{S}_y + \frac{1}{2}\mathsf{B}_y$ where S_x and S_y are skew-symmetric matrices.

Often, SBP operators will satisfy stronger accuracy conditions:

Definition 2 (SBP Accuracy)**.**

1) $\mathsf{D}_x\boldsymbol{p}_\kappa = \frac{\partial \boldsymbol{p}}{\partial x}|_\kappa$ where $\boldsymbol{p}_\kappa$ is a vector of polynomials of degree p evaluated at the nodes of the element and $\frac{\partial \boldsymbol{p}}{\partial x}|_\kappa$ are their derivatives. A similar condition applies to D_y.
2) $\boldsymbol{v}_\kappa^T\mathsf{B}_x\boldsymbol{p}_\kappa = \int vpn_x \mathrm{d}\Gamma_\kappa$ where v and p are polynomials whose degrees sum to $r \geq p$. A similar condition applies to B_y.

For SBP operators that satisfy the accuracy conditions, the following integral relations can be shown:

$$\boldsymbol{v}_\kappa^T\mathsf{Q}_x\boldsymbol{p}_\kappa = \boldsymbol{v}_\kappa^T\mathsf{P}\mathsf{D}_x\boldsymbol{p}_\kappa = \int v\frac{\partial p}{\partial x}\mathrm{d}\Omega_\kappa + O(h^{p+1}), \tag{1}$$

and

$$\begin{aligned}
\mathsf{Q}_x + \mathsf{Q}_x^T &= \left(\mathsf{S}_x + \frac{1}{2}\mathsf{B}_x\right) + \left(\mathsf{S}_x + \frac{1}{2}\mathsf{B}_x\right)^T && (2)\\
&= \left(\mathsf{S}_x + \frac{1}{2}\mathsf{B}_x\right) + \left(\mathsf{S}_x^T + \frac{1}{2}\mathsf{B}_x^T\right) && (3)\\
&= \left(\mathsf{S}_x + \frac{1}{2}\mathsf{B}_x\right) - \mathsf{S}_x + \frac{1}{2}\mathsf{B}_x && (4)\\
&= \mathsf{B}_x && (5)
\end{aligned}$$

Therefore

$$\boldsymbol{v}^T\mathsf{Q}_x\boldsymbol{p} = -\boldsymbol{v}^T\mathsf{Q}_x^T\boldsymbol{p} + \boldsymbol{v}^T\mathsf{B}_x\boldsymbol{p}, \tag{6}$$

which, considering (1), is the discrete analogy of integration by parts.

The boundary operators can be decomposed into operators over individual faces, in exactly the same way that an integral over all faces of an element can be broken into the sum of integrals over individual faces:

$$\mathsf{B}_x = \sum_{\gamma\in\Gamma_L}\mathsf{B}_{x,L} = \sum_{\gamma\in\Gamma_L}\mathsf{R}_L^T\mathsf{B}_L\mathsf{N}_{x,L}\mathsf{R}_L, \tag{7}$$

where R_L interpolates from the volume nodes of an element to the face nodes on a given face γ, B_L is a diagonal matrix of face integration weights, and $\mathsf{N}_{x,L}$ is a diagonal matrix containing the x component of the normal vector, oriented outward from element L. Note that the subscript will be used to distinguish between the two elements that share the face γ, and that R_L has an implicit dependence on γ (ie. R_L is different on each face of the element).

We require $\mathsf{R}_L\mathbf{1} = \mathbf{1}$ to satisfy Definition 1, and exact interpolation of polynomials up to degree p to satisfy the accuracy conditions in Definition 2.

For elements where the face nodes and volume nodes are collocated, R is a diagonal matrix where the entries on the diagonal are either 1 or 0. This results in a diagonal B_x. Further, for two dimensional tensor-product elements, B_L is the mass matrix of the one-dimensional SBP operator.

An important linear algebra operation is the Hadamard Product.

Definition 3 (Hadamard Product)**.** The Hadamard, or elementwise product, denoted $\mathsf{C} = \mathsf{A} \circ \mathsf{B}$, is defined as

$$\mathsf{C}_{ij} = \mathsf{A}_{ij}\mathsf{B}_{ij}, \tag{8}$$

where the subscripts denote matrix indexing.

B. Problem Definition

We will solve the following hyperbolic partial differential equation

$$\frac{\partial u}{\partial t} + \frac{\partial f_x(u)}{\partial x} + \frac{\partial f_y(u)}{\partial y} = 0 \tag{9}$$

with conservative variables u, entropy variables $w = w(u)$, and nonlinear fluxes f_x and f_y. For the following discretization, a numerical flux function $f_x^\star(u_L, u_R)$ with the entropy-conservative property is required:

$$(w(u_L) - w(u_R))^T \, f_x^\star(u_L, u_R) = \psi_{x,L} - \psi_{x,R}, \tag{10}$$

where ψ is the potential flux. The flux function must also satisfy consistency:

$$f_x^\star(u, u) = f_x(u) \tag{11}$$

and symmetry

$$f_x^\star(u_L, u_R) = f_x^\star(u_R, u_L) \tag{12}$$

A similar flux function in the y direction is required as well.

C. Curvilinear Mapping

In order to accommodate curved elements, the standard approach is to define the discretization operators on a reference domain, with coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)$, and map that domain to the physical (curved) domain with coordinates $\boldsymbol{x} = (x, y, z)$. The following is a brief review that highlights the properties that will be important for the discrete analysis of free-stream preservation. For a more detailed treatment, see [4, 5].

The metric invariants can be derived by transforming the spatial derivative in the following equation into reference space:

$$\frac{\partial u}{\partial t} + \frac{\partial f_i}{\partial x_i} = 0. \tag{13}$$

This can be done by multiplying the above equation by $J = \det(\frac{\partial x_i}{\partial \xi_j})$, and applying $\frac{\partial (\cdot)}{\partial x_i} = \frac{\partial \xi_j}{\partial x_i}\frac{\partial (\cdot)}{\partial \xi_j}$:

$$J\frac{\partial u}{\partial t} + J\frac{\partial f_i}{\partial x_i} = 0 \tag{14}$$

$$J\frac{\partial u}{\partial t} + J\frac{\partial \xi_j}{\partial x_i}\frac{\partial f_i}{\partial \xi_j} = 0 \tag{15}$$

$$J\frac{\partial u}{\partial t} + \frac{\partial}{\partial \xi_j}\left(J\frac{\partial \xi_j}{\partial x_i} f_i\right) - f_i\frac{\partial}{\partial \xi_j}\left(J\frac{\partial \xi_j}{\partial x_i}\right) = 0, \tag{16}$$

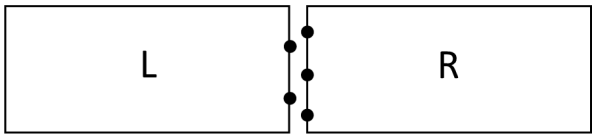


Fig. 1 A p-nonconformal interface between two elements

where the final line follows from the product rule of differentiation. If the original equation is equal to zero when u is constant, the mapping equation should have the same property. This argument gives the metric identities:

$$\frac{\partial}{\partial \xi_j}\left(J\frac{\partial \xi_j}{\partial x_i}\right) = 0 \tag{17}$$

I can be shown [4] the vectors $J\frac{\partial \xi_i}{\partial \boldsymbol{x}}$ can be computed from the mapping $\boldsymbol{x}(\boldsymbol{\xi})$:

$$J\frac{\partial \xi_i}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \xi_j} \times \frac{\partial \boldsymbol{x}}{\partial \xi_k} \qquad i, j, k \text{ cyclic} \tag{18}$$

where $\times$ denotes the vector cross product. Expanding the cross product

$$\begin{aligned} \frac{\partial \boldsymbol{x}}{\partial \xi_j} \times \frac{\partial \boldsymbol{x}}{\partial \xi_k} &= \begin{vmatrix} \hat{\boldsymbol{i}} & \hat{\boldsymbol{j}} & \hat{\boldsymbol{k}} \\ \frac{\partial x_1}{\partial \xi_j} & \frac{\partial x_2}{\partial \xi_j} & \frac{\partial x_3}{\partial \xi_j} \\ \frac{\partial x_1}{\partial \xi_k} & \frac{\partial x_2}{\partial \xi_k} & \frac{\partial x_3}{\partial \xi_k} \end{vmatrix} \\ &= \left(\frac{\partial x_2}{\partial \xi_j}\frac{\partial x_3}{\partial \xi_k} - \frac{\partial x_3}{\partial \xi_j}\frac{\partial x_2}{\partial \xi_k}\right)\hat{\boldsymbol{i}} - \left(\frac{\partial x_1}{\partial \xi_j}\frac{\partial x_3}{\partial \xi_k} - \frac{\partial x_3}{\partial \xi_j}\frac{\partial x_1}{\partial \xi_k}\right)\hat{\boldsymbol{j}} + \left(\frac{\partial x_1}{\partial \xi_j}\frac{\partial x_2}{\partial \xi_k} - \frac{\partial x_2}{\partial \xi_j}\frac{\partial x_1}{\partial \xi_k}\right)\hat{\boldsymbol{k}} \end{aligned} \tag{19}$$

The components of the normal vectors used in (7) can be computed as

$$n_x = J\frac{\partial \boldsymbol{\xi}}{\partial x} \cdot \boldsymbol{n}_\xi, \tag{20}$$

where $\boldsymbol{n}_\xi$ is the ξ component of the face normal vector for the reference element. When the reference element is chosen to be straight-sided, this vector is constant along each face. An important conclusion that can be drawn from the above derivation is that if $\boldsymbol{x}(\boldsymbol{\xi})$ is polynomial of degree $p+1$, then $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$ is polynomial of degree p, and $J\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ is polynomial of degree $2p$ by (19) in three dimensions. In two dimensions, $x_3 = \xi_3$, and is orthogonal to the other dimensions, in which case (19) gives a polynomial of degree p.

III. p-Nonconformal Discretization

In this section we extend the discretization in [1] to SBP operators with $2p-1$ accurate quadratures. We first present the discretization, then show how the discretization can be interpreted as the construction of a new SBP operator using a different quadrature rule on the p-nonconformal face. We then show the properties of this discretization, and prove stability, conservation, and free-stream preservation.

A. Discretization

Figure 1 shows an example p-nonconformal interface between two elements. To discretize this interface, we define a face M with quadrature rule B_M and normal vectors $\mathsf{N}_{x,M}$ and $\mathsf{N}_{y,M}$, evaluated at the quadrature points. This quadrature rule may be different than the quadrature rules for the left and right faces, B_L and B_R. Interpolation operators P_{LM} and P_{RM} interpolate from the face quadrature nodes of the L and R elements to the quadrature nodes on M. The interpolation from the volume nodes of L to the quadrature points of M can be written as $\mathsf{P}_{LM}\mathsf{R}_L$, and an analogous expression can be written for element R.

This equation (9) can be discretized on element L as

(a) Element in reference space **(b) Element in physical space**

Fig. 2 Mapping of element from reference space to physical space

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \left(\mathsf{D}_x \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L)\right)\mathbf{1} + \left(\mathsf{D}_y \circ \mathsf{F}_y(\boldsymbol{u}_L, \boldsymbol{u}_L)\right)\mathbf{1} \\ = \frac{1}{2}\mathsf{P}^{-1}\sum_x \Bigg(\left(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L\right) \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L) \\ - \left(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R\right) \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R) \Bigg)\mathbf{1} \\ + \mathsf{P}^{-1}\text{O.F.T},\end{aligned} \tag{21}$$

The matrix of flux function values is

$$\left(\mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R)\right)_{ij} = 2f^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}), \tag{22}$$

where the i and j subscripts denotes the conservative variables at node i of element L and node j of element R, respectively.

The right-hand side shows the simultaneous approximation term (SAT) term for the p-nonconformal interface. The SATs for the other faces are included in O.F.T. Note that the operator $\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R$ has the property $\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R = -\left(\mathsf{R}_R^T \mathsf{P}_{RM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM}^T \mathsf{R}_L\right)^T$, which is the essential property for proving entropy-conservation of simplex elements in [6].

From equation (7), the R_L operator interpolates from the volume nodes to the face quadrature points, and P_{LM} interpolates from the face quadrature points on M. The combined operator, $\mathsf{P}_{LM}\mathsf{R}_L$, therefore, interpolates from the volume nodes of L to the face quadrature points of M. As a result, the operator $\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L$ has similar structure to the face operator in (7), using B_M to perform the face integral rather than B_L. Hence, the idea behind the discretization is to define an appropriate middle element M, which is common to L and R, and use it to compute the face term between the p-nonconforming elements. In Section III.C, we will show how to define D_x and $\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L$ such that they form an SBP operator, but first we review the mapping of elements from reference space into physical space.

B. Element Coordinate Mapping

Figure 2 illustrates the mapping of an element from reference space into physical space. The continuous properties of the mapping were reviewed in Section II.C, we now show how to construct a mapping for each element.

The essential requirement of the mapping is that it have C^0 inter-element continuity. This ensures that adjacent elements will generate a consistent representation of their shared face. If the governing equation is solved on a curved domain, it may be necessary to have curved elements on the boundary, to ensure the geometric approximation error converges to zero at least as fast at the solution approximation error (ie. the error associated with D_x and the SAT). The current work considers only interfaces, and it may be possible to avoid curving interfaces on the interior of the domain, however we treat the general case of curved interfaces for generality.

The following result is well-known in the continuous finite element literature.

Lemma 1 (C^0 Inter-element Continuity)**.** *Let the coordinate field of an element be defined on a Lagrange polynomial basis. Further, let the node set include the boundary nodes such that the nodes on the boundary support a Lagrange basis for the face. Then the coordinate field has C^0 inter-element continuity.*

Proof. We prove this for a tensor-product element, but the result is easily extended to general elements.

Let the reference coordinate system have $\xi, \eta \in [-1, 1]$, with n nodes in the ξ direction and m nodes in the η direction. With ℓ_i denoting the ith Lagrange polynomial, the x coordinate field can be written

$$x(\xi, \eta) = \sum_{i}^{n} \sum_{j}^{m} x_{ij} \ell_i(\xi) \ell_j(\eta) \tag{23}$$

where x_{ij} is the x coordinate evaluated at node (i, j).

On the right face of the element, we have $\xi = 1$. By the nodal interpolation property of Lagrange polynomials, and the requirement that the node set include the boundary, we have $\ell_n(1) = 1$ and all other $\ell_i(1) = 0$. Thus on the face we have

$$x(1, \eta) = \sum_{j}^{m} x_{nj} \ell_j(\eta). \tag{24}$$

This shows that the x coordinate of the face depends only on the x coordinates evaluated at the Lagrange nodes on that face, and not on the x coordinate elsewhere. Thus, if two adjacent elements have the same Lagrange nodes and have the same values for x_{nj} (x_{1j} for the adjacent element), they will produce exactly the same shape for their common face. □

An important consequence of this Lemma is that the normal vectors can be evaluated via Eq. (20) at **any** point on the face, and the result will be the same whether the mapping of the L or R element is used.

C. SBP Operator Construction

In this section, we show how to construct an SBP operator given:

- an SBP operator in reference space, with weak integration matrix Q_{ξ_i}
- a face node set and associated quadrature rule B_M.

The mechanics of this approach were originally developed for three-dimensional curvilinear elements in [6] for computing metrics that satisfy the metric invariants, and was subsequently used in [1] in the more general context of constructing SBP operators.

We first define the metrics and the determinant of the mapping

$$\Lambda_{\xi_i, x_j} \approx \text{diag}\left(\left[\left(J \frac{\partial \xi_i}{\partial x_j}\right)_1 \cdots \left(J \frac{\partial \xi_i}{\partial x_j}\right)_n\right]\right) \tag{25}$$

$$|J| = \text{diag}\left(\left[|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}|_1 \cdots |\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}|_n\right]\right) \tag{26}$$

The approximation sign in the Λ_{ξ_i, x_j} equation will be explained shortly.

The mass matrix can be mapped to physical space as $\mathsf{P}|J|$, and we can then use the metrics to map Q_{ξ_i} to the skew-symmetric matrix S_x in physical space:

$$\mathsf{S}_{x_i} = \frac{1}{2}\left(\Lambda_{\xi_1, x_i} \mathsf{Q}_{\xi_1} + \Lambda_{\xi_2, x_i} \mathsf{Q}_{\xi_2} - \mathsf{Q}_{\xi_1}^T \Lambda_{\xi_1, x_i} - \mathsf{Q}_{\xi_2, x_i}^T \Lambda_{x i_2, x_i}\right). \tag{27}$$

We now enforce the condition that the boundary operator use the M grid quadrature rule

$$\mathsf{B}_{x_i} = \sum_{\gamma} \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x_i, M} \mathsf{P}_{LM} \mathsf{R}_L. \tag{28}$$

The R_L and P_{LM} interpolation operators operate on functions in reference space, not physical space.

Using this boundary operator to construct the remaining operator matrices:

$$\mathsf{Q}_{x_i} = \mathsf{S}_{x_i} + \frac{1}{2}\sum_{\gamma} \mathsf{B}_{x_i} \tag{29}$$

$$= \frac{1}{2}\left(\Lambda_{\xi_1,x_i}\mathsf{Q}_{\xi_1} + \Lambda_{\xi_2,x_i}\mathsf{Q}_{\xi_2} - \mathsf{Q}_{\xi_1}^T\Lambda_{\xi_1,x_i} - \mathsf{Q}_{\xi_2,x_i}^T\Lambda_{\xi_2,x_i} + \sum_{\gamma}\mathsf{B}_{x_i}\right)$$

$$\mathsf{D}_{x_i} = (\mathsf{P}|J|)^{-1}\,\mathsf{Q}_{x_i} \tag{30}$$

It is clear that the mapped SBP operator will satisfy parts 4 and 5 of Definition 1 because they are used in the construction. Part 2 will be satisfied if B_M chosen to be sufficiently accurate to exactly integrate $\mathsf{N}_{x_i,M}$. From Sections II.C and III.B, $\mathsf{N}_{x_i,M}$ are polynomials of $2(q-1)$, where q is the degree of the mapping. What remains to be shown is Part 1 of Definition 1, as well as determining if any properties of Definition 2 hold.

1. Constant Exactness

In this section we show how to make the mapped SBP operator differentiate constants exactly. For reference-space SBP operators with sufficiently accurate quadratures and interpolation operators, this property will hold using the exact metrics Λ_{ξ_i,x_j}. For other operators, the metric projection trick from [6] will be used to find metrics such that the property does hold.

To determine if the property holds, note that $\mathsf{D}_{x_i}\mathbf{1} = (\mathsf{P}|J|)^{-1}\,\mathsf{Q}_{x_i}\mathbf{1}$ will be zero if and only if $\mathsf{Q}_{x_i}\mathbf{1} = \mathbf{0}$. Then

$$\mathsf{Q}_{x_i}\mathbf{1} = \mathbf{0} \tag{31}$$

$$\frac{1}{2}\left(\sum_j\left(\Lambda_{\xi_j,x_i}\mathsf{Q}_{\xi_j} - \mathsf{Q}_{\xi_j}^T\Lambda_{\xi_j,x_i}\right) + \sum_{\gamma}\mathsf{B}_{x_i}\right)\mathbf{1} = \mathbf{0} \tag{32}$$

$$\frac{1}{2}\left(\sum_j\left(\Lambda_{\xi_j,x_i}\mathsf{Q}_{\xi_j}\mathbf{1} - \mathsf{Q}_{\xi_j}^T\Lambda_{\xi_j,x_i}\mathbf{1}\right) + \sum_{\gamma}\mathsf{B}_{x_i}\mathbf{1}\right) = \mathbf{0} \tag{33}$$

$$\frac{1}{2}\left(\sum_j\left(-\mathsf{Q}_{\xi_j}^T\Lambda_{\xi_j,x_i}\mathbf{1}\right) + \sum_{\gamma}\mathsf{B}_{x_i}\mathbf{1}\right) = \mathbf{0} \tag{34}$$

$$\frac{1}{2}\left(\sum_j\left(\mathsf{Q}_{\xi_j}\Lambda_{\xi_j,x_i}\mathbf{1} - \sum_{\gamma}\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L\Lambda_{\xi_j,x_i}n_{\xi_j}\mathbf{1}\right) + \sum_{\gamma}\mathsf{B}_{x_i}\mathbf{1}\right) = \mathbf{0} \tag{35}$$

$$\frac{1}{2}\left(\sum_j\left(\mathsf{Q}_{\xi_j}\Lambda_{\xi_j,x_i}\mathbf{1} - \sum_{\gamma}\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L\Lambda_{\xi_j,x_i}n_{\xi_j}\mathbf{1}\right) + \sum_{\gamma}\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x_i,M}\mathsf{P}_{LM}\mathsf{R}_L\mathbf{1}\right) = \mathbf{0} \tag{36}$$

The $\mathsf{Q}_{\xi_j}\Lambda_{\xi_j,x_i} = \mathsf{P}\mathsf{D}_{\xi_j}\Lambda_{\xi_j,x_i}$ term is a discretization of the metric invariants (17), and is therefore high-order accurate. For two-dimensional elements where $q \le p+1$, D_{ξ_j} can exactly differentiate the metrics, and this term will be zero.

What remains is the difference of two face terms. Note that the reference element is straight-sided, and thus n_{ξ_j} is a constant for each face. To determine if these terms sum to zero, we rewrite $\mathsf{N}_{x_i,M}$ in terms of the n_{ξ_j} and the metrics evaluated at the M quadrature points using (20). If the interpolation operators are sufficiently accurate, the metrics at the quadrature nodes can be written as interpolations of $\Lambda_{\xi_j x_i}$. The final two terms can then be rewritten

$$\sum_j\sum_{\gamma}\left(-\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L\Lambda_{\xi_j,x_i}n_{\xi_j} + \mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L\Lambda_{\xi_j,x_i}n_{\xi_j}\right)\mathbf{1} \tag{37}$$

It has been shown in [1] that $\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L = \mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L$ under certain conditions.

Lemma 2 (Inner-product Preservation Property)**.** *Let R_L and P_{LM} be constructed such that are exact for polynomials of degree $r \ge p$ (where p is the degree of the SBP operator), and let B_L and B_M be exact for polynomials of degree $2r$. Then $\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L = \mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L$*

Proof. The proof is given for the two dimensional case. The proof for the three dimensional case is similar.

Given an SBP operator with m volume nodes, n face nodes, and an M grid element with s nodes, let

$$\mathsf{V}_r^\Omega = \phi_j(\boldsymbol{x}_{\gamma,i}) \qquad \text{for } j = 1 : \frac{(r+1)(r+2)}{2} \tag{38}$$

denote the Vandermonde matrix containing a polynomial basis evaluated at the volume nodes of the element. This equation has at least one solution if $m \geq \frac{(r+1)(r+2)}{2}$.

Similarly, let V_r^γ denote the two-dimensional Vandermonde matrix on the nodes of B_L, and V_r^M the two-dimensional Vandermonde matrices on the nodes of B_M. The operators must satisfy the conditions

$$\mathsf{R}_L \mathsf{V}_r^\Omega = \mathsf{V}_r^\gamma \tag{39}$$

$$\mathsf{P}_{LM} \mathsf{V}_r^\gamma = \mathsf{V}_r^M \tag{40}$$

Note that the equation for P_{LM} is written in terms of two-dimensional Vandermonde matrices, even though the interpolation itself is one dimensional. Because the one-dimensional polynomials are a subset of the two-dimensional polynomials, V_r^γ and V_r^M have several linearly dependent columns and the equation for P_{LM} has at least one solution if $n \geq r+1$. R_L and P_{LM} can be solved for

$$\mathsf{R}_L = \mathsf{V}_r^\gamma \left(\mathsf{V}_r^\Omega\right)^\ddagger \tag{41}$$

$$\mathsf{P}_{LM} = \mathsf{V}_r^M \left(\mathsf{V}_r^\gamma\right)^\ddagger. \tag{42}$$

Here $\mathsf{A}^\ddagger$ denotes a right generalized inverse of the matrix A, that is, a matrix that satisfies $\mathsf{A}\mathsf{A}^\ddagger = \mathsf{I}$. We do not require $\mathsf{A}^\ddagger$ be the Moore-Penrose pseudo-inverse, and avoiding this requirement allows a broader range of R_L and P_{LM} to be used in the remainder of the proof. In particular, it allows choosing a R_L to have a tensor-product structure if the SBP operator is a tensor-product element.

We now prove the main result. If B_L and B_M are exact for $2r$ polynomials

$$\left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L \mathsf{V}_r^\gamma = \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M \mathsf{V}_r^M. \tag{43}$$

We now right multiply by $\left(\mathsf{V}_r^\gamma\right)^\ddagger$ and left multiply by $\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}$

$$\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L \mathsf{V}_r^\gamma \left(\mathsf{V}_r^\gamma\right)^\ddagger = \left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M \mathsf{V}_r^M \left(\mathsf{V}_r^\gamma\right)^\ddagger \tag{44}$$

$$\mathsf{B}_L = \left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M \mathsf{V}_r^M \left(\mathsf{V}_r^\gamma\right)^\ddagger, \tag{45}$$

where the second line follows from the generalized inverse property. Now right multiply by $\mathsf{V}_r^\gamma \left(\mathsf{V}_r^\Omega\right)^\ddagger$ and left multiply by its transpose to get

$$\underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T} \mathsf{B}_L \underbrace{\mathsf{V}_r^\gamma \left(\mathsf{V}_r^\Omega\right)^\ddagger}_{\mathsf{R}_L} = \underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T} \underbrace{\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^M\right)^T}_{\mathsf{P}_{LM}^T} \mathsf{B}_M \underbrace{\mathsf{V}_r^M \left(\mathsf{V}_r^\gamma\right)^\ddagger}_{\mathsf{P}_{LM}} \underbrace{\mathsf{V}_r^\gamma \left(\mathsf{V}_r^\Omega\right)^\ddagger}_{\mathsf{R}_L}, \tag{46}$$

which is the desired result. □

Remark 1. Lemma 2 requires B_M to be exact for **at least** $2r$ polynomials. The quadrature rule accuracy (and number of points) can be greater. The interface between two elements should have B_M exact for two times the **maximum** degree polynomial on either the left or right element.

Remark 2. Lemma 2 is referred to as inner-product preserving because it ensures $\boldsymbol{v}\mathsf{R}_L^T \mathsf{B}_L \mathsf{R}_L \boldsymbol{u} = \boldsymbol{v}^T \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{P}_{LM} \mathsf{R}_L \boldsymbol{u}$ for all $\boldsymbol{u}$, $\boldsymbol{v}$. As shown below, it is sufficient but not necessary for an entropy-stable, conservative, and free-stream preserving discretization.

There is a special case of this lemma useful for straight-sided elements.

Lemma 3 (Constant Preservation Property)**.** *Let* R_L *and* P_{LM} *be constructed such that are exact for polynomials of degree* $r \geq p$ *(where* p *is the degree of the SBP operator), and let* B_L *and* B_M *be exact for polynomials of degree* r*. Then* $\mathsf{R}_L^T \mathsf{B}_L \mathsf{R}_L \mathbf{1} = \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{P}_{LM} \mathsf{R}_L \mathbf{1}$

Proof. The proof is very similar to Lemma 2. Starting with

$$\left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L \mathbf{1} = \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M \mathbf{1} \tag{47}$$

and left multiply with $\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T \left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}$:

$$\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T \left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L \mathbf{1} = \left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T \left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M \mathbf{1} \tag{48}$$

$$\underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T} \mathsf{B}_L \mathbf{1} = \underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T} \left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T} \underbrace{\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T} \left(\mathsf{V}_r^M\right)^T}_{\mathsf{P}_{LM}^T} \mathsf{B}_M \mathbf{1} \tag{49}$$

Noting that $\mathsf{R}_L^T \mathsf{B}_L \mathsf{R}_L \mathbf{1} = \mathsf{R}_L^T \mathsf{B}_L \mathbf{1}$ and $\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{P}_{LM} \mathsf{R}_L \mathbf{1} = \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathbf{1}$, we have the desired result.

□

Remark 3. As with Lemma 2, B_M should exactly integrate the **maximum** degree polynomial present on either the left or right element.

If the conditions of Lemma 2 are satisfied, then (37) sums to zero, establishing that $\mathsf{D}_{x_i} \mathbf{1} = \mathbf{0}$. Alternatively, for straight-sided elements, Lemma 3 applies for any SBP operator, and Λ_{ξ_j, x_i} are constant, and thus can be pulled out of equation (37). The difference is then zero by Lemma 3.

If the conditions of Lemma 2 for curved elements, or Lemma 3 for straight-sided elements, are not satisfied, the metric projection trick from [6] is used to compute metrics Λ_{ξ_j, x_i} such that $\mathsf{D}_{x_i} \mathbf{1} = \mathbf{0}$ holds. The only requirements of the metric projection are that R_L and P_{LM} exactly interpolate constants, and that B_M can exactly integrate $\mathsf{N}_{x,M}$. Unlike Lemma 2, no requirements are placed on B_L. For completeness, the metric projection procedure is reproduced in Appendix A. Note that for three-dimensional elements, the metric invariants are not exactly satisfied discretely, and so the metric projection is required even if Lemma 2 is satisfied.

2. Accuracy

In this section we analyze the accuracy of an SBP operator constructed from Eqs. (28) and (29). This analysis will assume straight-sided elements, and will attempt to prove polynomial exactness. For conformal discretizations, the curvilinear operators are not exact for polynomials, but retain design-order accuracy [6].

For straight-sided elements, the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is linear therefore $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}$ and $|J|$ are constant. Rewriting

$$\Lambda_{\xi_j, x_i} = \frac{\partial \xi_j}{\partial x_i} \mathsf{I} \tag{50}$$

$$|J| = A\mathsf{I} \tag{51}$$

$$\mathsf{N}_{x,M} = n_x A \mathsf{I} \tag{52}$$

where I is the identity matrix.

The Q_{x_i} operator can then be written

$$\mathsf{Q}_{x_i} = \frac{1}{2}\sum_j \left(A\frac{\partial \xi_j}{\partial x_i}\mathsf{Q}_{\xi_j} - \mathsf{Q}_{\xi_j}^T A\frac{\partial \xi_j}{\partial x_i}\right) + \frac{1}{2}\mathsf{B}_{x_i} \tag{53}$$

$$= \frac{1}{2}\sum_j A\frac{\partial \xi_j}{\partial x_i}\left(\mathsf{Q}_{\xi_j} - \mathsf{Q}_{\xi_j}^T\right) + \frac{1}{2}\mathsf{B}_{x_i} \tag{54}$$

$$= \frac{1}{2}\sum_j A\frac{\partial \xi_j}{\partial x_i}\left(\mathsf{Q}_{\xi_j} + \mathsf{Q}_{\xi_j} - \sum_\gamma \mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L^T n_{\xi_j}\right) + \frac{1}{2}\sum_\gamma \left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L n_{x_i}\right) \tag{55}$$

$$= \sum_j \left(A\frac{\partial \xi_j}{\partial x_i}\mathsf{Q}_{\xi_j}\right) - \frac{1}{2}\sum_j\sum_\gamma \left(\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L^T A\frac{\partial \xi_j}{\partial x_i}n_{\xi_j}\right) + \frac{1}{2}\sum_\gamma \left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L n_{x_i}\right) \tag{56}$$

$$= \sum_j \left(A\frac{\partial \xi_j}{\partial x_i}\mathsf{Q}_{\xi_j}\right) - \frac{1}{2}\sum_\gamma \left(\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L^T n_{x_i}\right) + \frac{1}{2}\sum_\gamma \left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L n_{x_i}\right) \tag{57}$$

$$\tag{58}$$

If Lemma 2 holds, the face terms sum to zero and Q_{x_i} will inherit the accuracy properties of Q_{ξ_j}. Otherwise, a one-sided version of Lemma 2 can be developed.

Lemma 4. *Let $\boldsymbol{q}$ be a degree q polynomial and R_L and P_{LM} be defined as in Lemma 2. If $q + r$ is less than or equal to the degree of polynomial for which the B_L and B_M are exact, then $\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L^T\boldsymbol{q} = \mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L\boldsymbol{q}$*

Proof. If a polynomial of degree $q + r$ can be exactly integrated by B_L and B_M, then we have

$$\left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L\boldsymbol{q} = \left(\mathsf{V}_r^M\right)^T \mathsf{B}_M\boldsymbol{q} \tag{59}$$

and left multiply with $\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}$:

$$\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T \mathsf{B}_L\boldsymbol{q} = \left(\mathsf{V}_r^\Omega\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}\left(\mathsf{V}_r^M\right)^T \mathsf{B}_M\boldsymbol{q} \tag{60}$$

$$\underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T} \mathsf{B}_L\boldsymbol{q} = \underbrace{\left(\mathsf{V}_r^\Omega\right)^{\ddagger,T}\left(\mathsf{V}_r^\gamma\right)^T}_{\mathsf{R}_L^T}\underbrace{\left(\mathsf{V}_r^\gamma\right)^{\ddagger,T}\left(\mathsf{V}_r^M\right)^T}_{\mathsf{P}_{LM}^T} \mathsf{B}_M\boldsymbol{q} \tag{61}$$

Noting that $\mathsf{R}_L^T\mathsf{B}_L\mathsf{R}_L\boldsymbol{q} = \mathsf{R}_L^T\mathsf{B}_L\boldsymbol{q}$ and $\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{P}_{LM}\mathsf{R}_L\boldsymbol{q} = \mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\boldsymbol{q}$, we have the desired result.

□

Remark 4. For this property to hold for the elements on both sides of the interface, B_M should be at least as accurate as the **maximum** quadrature on either side of the interface.

Numerical results show that generalized elements are exact for polynomials of degree p, and conventional elements, which have $2p - 1$ exact quadratures, are exact for degree $p - 1$ polynomials, consistent with this theory.

D. Discretization Properties

Now that the properties $\mathsf{D}_{x_i}\mathbf{1} = \mathbf{0}$ and $\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{LR}\mathsf{R}_R = -\left(\mathsf{R}_R^T\mathsf{P}_{RM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{LM}\mathsf{R}_L\right)^T$ have been established, we can show stability, conservation, and free-stream preservation of the discretization 21. The discretization can be rewritten in weak form by contracting with $\boldsymbol{v}^T\mathsf{P}$:

$$
\begin{aligned}
\boldsymbol{v}^T \mathsf{P} \frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \boldsymbol{v}^T \left(\mathsf{Q}_x \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} + \boldsymbol{v}^T \left(\mathsf{Q}_y \circ \mathsf{F}_y(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} \\
= \frac{1}{2} \boldsymbol{v}^T \sum_x \Bigg(\left(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L\right) \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L) \\
- \left(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R\right) \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R) \Bigg) \mathbf{1} \\
+ \text{O.F.T},
\end{aligned}
\tag{62}
$$

Applying the splitting $\mathsf{Q}_{x_i} = \mathsf{S}_{x_i} + \frac{1}{2}\mathsf{B}_{x_i}$, and recalling that B_{x_i} is defined according to Eq. (28) gives

$$
\begin{aligned}
\boldsymbol{v}^T \mathsf{P} \frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \boldsymbol{v}^T \left(\mathsf{S}_x \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} + \boldsymbol{v}^T \left(\mathsf{S}_y \circ \mathsf{F}_y(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} \\
= -\frac{1}{2} \boldsymbol{v}^T \sum_x \Bigg(\left(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R\right) \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R) \Bigg) \mathbf{1} \\
+ \text{W.F.T},
\end{aligned}
\tag{63}
$$

Here, W.F.T denotes the face terms for the other faces after cancellation with B_{x_i}.

1. Entropy Stability

To show entropy stability, we replace the the test function $\boldsymbol{v}$ with the entropy variable $\boldsymbol{w}$. It has been shown previously that the entropy contributed by the volume terms is [6, 7]

$$
\boldsymbol{w}_L^T \left(\mathsf{S}_x \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} = -\mathbf{1}^T \mathsf{B}_x \boldsymbol{\psi}_{x,L} = -\sum_{\gamma \in \Gamma_L} \mathbf{1}^T \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_L \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L \boldsymbol{\psi}_{x,L}. \tag{64}
$$

To determine the entropy of the face term, sum the face terms from the two elements that share a face. To simplify notation, denote $\mathsf{B}_x^{LR} = \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM} \mathsf{R}_R$ and $\mathsf{B}_x^{RL} = -\mathsf{R}_R^T \mathsf{P}_{RM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L$, where the negative sign comes from the convention that $\mathsf{N}_{x,M}$ is oriented outward from element L. The sum of the face terms in the x direction is

$$
\boldsymbol{w}_L^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R)\right) \mathbf{1} + \boldsymbol{w}_R^T \left(\mathsf{B}_x^{RL} \circ \mathsf{F}_x(\boldsymbol{u}_R, \boldsymbol{u}_L)\right) \mathbf{1}. \tag{65}
$$

Denoting n_L as the number of volume nodes on the left element, and similarly n_R on the right element, this expression can be rewritten as a summation

$$\sum_i^{n_L} \boldsymbol{w}_{L,i}^T \sum_j^{n_R} 2\left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) + \sum_i^{n_R} \boldsymbol{w}_{R,i}^T \sum_j^{n_L} 2\left(\mathsf{B}_x^{RL}\right)_{ij} f_x^\star(\boldsymbol{u}_{R,i}, \boldsymbol{u}_{L,j}) \tag{66}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \boldsymbol{w}_{L,i}^T \left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) + 2\sum_i^{n_R}\sum_j^{n_L} \boldsymbol{w}_{R,i}^T \left(\mathsf{B}_x^{RL}\right)_{ij} f_x^\star(\boldsymbol{u}_{R,i}, \boldsymbol{u}_{L,j}) \tag{67}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \boldsymbol{w}_{L,i}^T \left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) - 2\sum_i^{n_R}\sum_j^{n_L} \boldsymbol{w}_{R,i}^T \left(\mathsf{B}_x^{LR}\right)_{ij}^T f_x^\star(\boldsymbol{u}_{R,i}, \boldsymbol{u}_{L,j}) \tag{68}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \boldsymbol{w}_{L,i}^T \left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) - 2\sum_i^{n_R}\sum_j^{n_L} \boldsymbol{w}_{R,i}^T \left(\mathsf{B}_x^{LR}\right)_{ji} f_x^\star(\boldsymbol{u}_{R,i}, \boldsymbol{u}_{L,j}) \tag{69}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \boldsymbol{w}_{L,i}^T \left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) - 2\sum_i^{n_L}\sum_j^{n_R} \boldsymbol{w}_{R,j}^T \left(\mathsf{B}_x^{LR}\right)_{ij} f_x^\star(\boldsymbol{u}_{R,j}, \boldsymbol{u}_{L,i}) \tag{70}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \left(\mathsf{B}_x^{LR}\right)_{ij} \left(\boldsymbol{w}_{L,i} - \boldsymbol{w}_{R_j}\right)^T f_x^\star(\boldsymbol{u}_{L,i}, \boldsymbol{u}_{R,j}) \tag{71}$$

$$= 2\sum_i^{n_L}\sum_j^{n_R} \left(\mathsf{B}_x^{LR}\right)_{ij} \left(\boldsymbol{\psi}_{x,L,i} - \boldsymbol{\psi}_{x,R,j}\right). \tag{72}$$

On the third line, the property $\mathsf{B}_x^{LR} = -\left(\mathsf{B}_x^{RL}\right)^T$ is used, on the fourth line the meaning of the i and j indices in the second term are reversed, and the final line uses the entropy conservative property of the flux function (10).

Transforming back into matrix notation and substituting the definitions of B_x^{LR} and B_x^{RL} gives

$$2\sum_i^{n_L}\sum_j^{n_R} \left(\mathsf{B}_x^{LR}\right)_{ij} \left(\boldsymbol{\psi}_{x,L,i} - \boldsymbol{\psi}_{x,R,j}\right) = 2\boldsymbol{\psi}_{x,L}^T \mathsf{B}_x^{LR}\mathbf{1} - 2\mathbf{1}^T \mathsf{B}_x^{LR}\boldsymbol{\psi}_{x,R} \tag{73}$$

$$= 2\boldsymbol{\psi}_{x,L}^T \mathsf{B}_x^{LR}\mathbf{1} + 2\mathbf{1}^T \left(\mathsf{B}_x^{RL}\right)^T \boldsymbol{\psi}_{x,R} \tag{74}$$

$$= 2\boldsymbol{\psi}_{x,L}^T \big(\mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \underbrace{\mathsf{P}_{RM}\mathsf{R}_R)\mathbf{1}}_{=\mathsf{R}_L\mathsf{P}_{LM}\mathbf{1}} + 2\underbrace{\mathbf{1}^T \left(\mathsf{R}_L^T \mathsf{P}_{LM}^T\right.}_{=\mathbf{1}^T\mathsf{R}_R^T\mathsf{P}_{LR}^T} \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{RM}\mathsf{R}_R\big)\boldsymbol{\psi}_{x,R} \tag{75}$$

$$= 2\boldsymbol{\psi}_{x,L}^T \mathsf{B}_{x,L}\mathbf{1} + 2\mathbf{1}^T \mathsf{B}_{x,R}\boldsymbol{\psi}_{x,R} \tag{76}$$

$$= 2\mathbf{1}^T \mathsf{B}_{x,L}\boldsymbol{\psi}_{x,L} + 2\mathbf{1}^T \mathsf{B}_{x,R}\boldsymbol{\psi}_{x,R} \tag{77}$$

The third line uses the property that interpolation operators are exact for constants, and the final line uses the symmetry of B_x. The additional subscript on $\mathsf{B}_{x,L}$ and $\mathsf{B}_{x,R}$ denote which elements the boundary operator lives on.

From (65) to (77), we have shown

$$\boldsymbol{w}_L^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_R)\right)\mathbf{1} + \boldsymbol{w}_R^T \left(\mathsf{B}_x^{RL} \circ \mathsf{F}_x(\boldsymbol{u}_R, \boldsymbol{u}_L)\right)\mathbf{1} = 2\mathbf{1}^T \mathsf{B}_{x,L}\boldsymbol{\psi}_{x,L} + 2\mathbf{1}^T \mathsf{B}_{x,R}\boldsymbol{\psi}_{x,R} \tag{78}$$

We can now write the time derivative of entropy for the combination of the left and right elements. Starting with (63), choosing the test function as the entropy variables $\boldsymbol{w}$, moving the volume term to the right-hand side, and noting the factor of $-\frac{1}{2}$ in front of the face terms gives:

$$\boldsymbol{w}_L^T \mathsf{P}^L \frac{\partial \boldsymbol{u}_L}{\partial t} + \boldsymbol{w}_R^T \mathsf{P}^R \frac{\partial \boldsymbol{u}_R}{\partial t} = \underbrace{\mathbf{1}^T \mathsf{B}_{x,L}\boldsymbol{\psi}_{x,L} + \mathbf{1}^T \mathsf{B}_{x,R}\boldsymbol{\psi}_{x,R}}_{\text{from (64)}} \underbrace{-\mathbf{1}^T \mathsf{B}_{x,L}\boldsymbol{\psi}_{x,L} - \mathbf{1}^T \mathsf{B}_{x,R}\boldsymbol{\psi}_{x,R}}_{\text{from (77)}} = 0. \tag{79}$$

Thus, for a periodic mesh, the rate of change of entropy over the entire mesh is zero.

2. Conservation

For the conservation analysis, we start with (63) and chose the test function to be $\mathbf{1}$:

$$\begin{aligned}\mathbf{1}^T \mathsf{P}\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t}+&\mathbf{1}^T \left(\mathsf{S}_x \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}+\mathbf{1}^T \left(\mathsf{S}_y \circ \mathsf{F}_y(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}\\ =\quad &-\frac{1}{2}\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}-\frac{1}{2}\mathbf{1}^T \left(\mathsf{B}_y^{LR} \circ \mathsf{F}_y(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}+\mathbf{1}^T \text{W.F.T},\end{aligned} \tag{80}$$

For the volume terms

$$\mathbf{1}^T \left(\mathsf{S}_x \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}=0 \tag{81}$$

due to the skew-symmetry of S_x and the symmetry of $\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)$. The same result holds for the y direction term.

For the face terms, we evaluate the amount of material that leaves one element and enters the adjacent element. Summing the two face terms

$$\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}+\mathbf{1}^T \left(\mathsf{B}_x^{RL} \circ \mathsf{F}_x(\boldsymbol{u}_R,\boldsymbol{u}_L)\right)\mathbf{1} \tag{82}$$

$$=\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}-\mathbf{1}^T \left(\left(\mathsf{B}_x^{LR}\right)^T \circ \mathsf{F}_x(\boldsymbol{u}_R,\boldsymbol{u}_L)\right)\mathbf{1} \tag{83}$$

$$=\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}-\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x^T(\boldsymbol{u}_R,\boldsymbol{u}_L)\right)\mathbf{1} \tag{84}$$

$$=\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}-\mathbf{1}^T \left(\mathsf{B}_x^{LR} \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1} \tag{85}$$

$$=0, \tag{86}$$

where the second line uses the property transpose property of B_x^{LR}, and the fourth line uses property that $\mathsf{F}_x^T(\boldsymbol{u}_R,\boldsymbol{u}_L) = \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)$, which follows from the definition of F_x in (22).

Thus, for a periodic mesh, the material that flows out of one element flows into another, and the total rate of change of material will sum to zero:

$$\sum_L \mathbf{1}^T \mathsf{P}\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t}=0. \tag{87}$$

3. Free-Stream Preservation

We now show that the scheme is free-stream preserving showing $\frac{\partial \boldsymbol{u}}{\partial t}=\mathbf{0}$ when $\boldsymbol{u}$ is constant. Starting from the strong form (21), the volume term can be rewritten

$$\left(\left(\mathsf{D}_x \circ \mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}\right)_i = 2\sum_j \mathsf{D}_{x,ij} f_x^\star(\boldsymbol{u}_{L,i},\boldsymbol{u}_{L,j}) = 2f(u)\sum_j \mathsf{D}_{x,ij}=0 \tag{88}$$

where $f(u)$ is the flux of the (constant) solution u. The final equality follows from the arguments in Section III.C.1 that D_{x_i} is exact for constant function.

All that remains is to show the SAT in (21) is zero for constant functions.

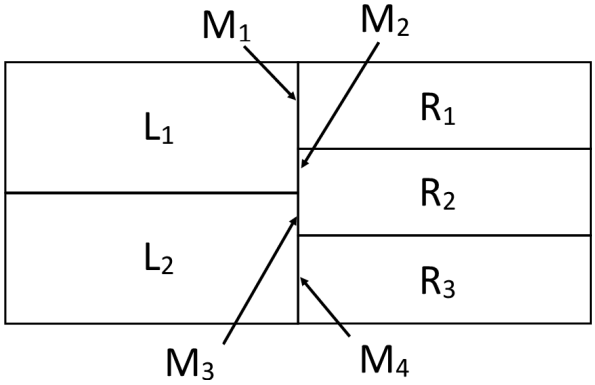


Fig. 3 An example hp-nonconformal mesh

$$\frac{1}{2}\mathsf{P}^{-1}\left(\left(\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{LM}\mathsf{R}_L\right)\circ\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)-\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{RM}\mathsf{R}_R\right)\circ\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}\right)_i \tag{89}$$

$$=\frac{1}{2}f(u)\mathsf{P}^{-1}\sum_j\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{LM}\mathsf{R}_L\right)_{ij}-\frac{1}{2}f(u)\mathsf{P}^{-1}\sum_j\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{RM}\mathsf{R}_R\right)_{ij} \tag{90}$$

$$=\frac{1}{2}f(u)\mathsf{P}^{-1}\sum_j\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\right)_{ik}\left(\mathsf{P}_{LM}\mathsf{R}_L\right)_{kj}-\frac{1}{2}f(u)\mathsf{P}^{-1}\sum_j\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\right)_{ik}\left(\mathsf{P}_{RM}\mathsf{R}_R\right)_{kj} \tag{91}$$

$$=\frac{1}{2}f(u)\mathsf{P}^{-1}\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\right)_{ik}\left(\sum_j\left(\mathsf{P}_{LM}\mathsf{R}_L\right)_{kj}-\sum_j\left(\mathsf{P}_{RM}\mathsf{R}_R\right)_{kj}\right) \tag{92}$$

$$=0 \tag{93}$$

The final line follows from the exact interpolation of constants by the interpolation operators, therefore the difference of the term in parenthesis is zero. Thus we have shown that $\frac{\partial \boldsymbol{u}}{\partial t}=\mathbf{0}$ when the solution is constant.

Importantly, none of the above proofs require the inner-production preservation property to hold. The metric projection trick can always be used to ensure $\mathsf{D}_{x_i}\mathbf{1}=\mathbf{0}$ holds, even if the special cases of Lemmas 2 or 3 do not hold.

IV. hp-Nonconformal Discretization

This section describes the hp-nonconformal discretization. Figure 3 shows an example hp-nonconformal mesh. The main idea behind this discretization is to extend the M integration rule from the p-nonconformal discretization to be an M grid of face elements. The M grid is formed by superimposing the vertices of all the L and R elements and defining an integration rule B_M on each element. We will use $M(L_i)$ and $M(R_i)$ to denote the set of M grid elements that overlap with L_i and R_i, respectively.

The discretization is then

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t}&+\left(\mathsf{D}_x\circ\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}+\left(\mathsf{D}_y\circ\mathsf{F}_y(\boldsymbol{u}_L,\boldsymbol{u}_L)\right)\mathbf{1}\\&=\frac{1}{2}\mathsf{P}^{-1}\sum_x\sum_{M\in M(L)}\left(\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{LM}\mathsf{R}_L\right)\circ\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_L)\right.\\&\qquad\left.-\left(\mathsf{R}_L^T\mathsf{P}_{LM}^T\mathsf{B}_M\mathsf{N}_{x,M}\mathsf{P}_{RM}\mathsf{R}_R\right)\circ\mathsf{F}_x(\boldsymbol{u}_L,\boldsymbol{u}_R)\right)\mathbf{1}\\&\qquad+\mathsf{P}^{-1}\text{O.F.T.}\end{aligned} \tag{94}$$

The discretization is very similar to (21), except that the SAT has a sum over the M grid elements on the boundary of element L.

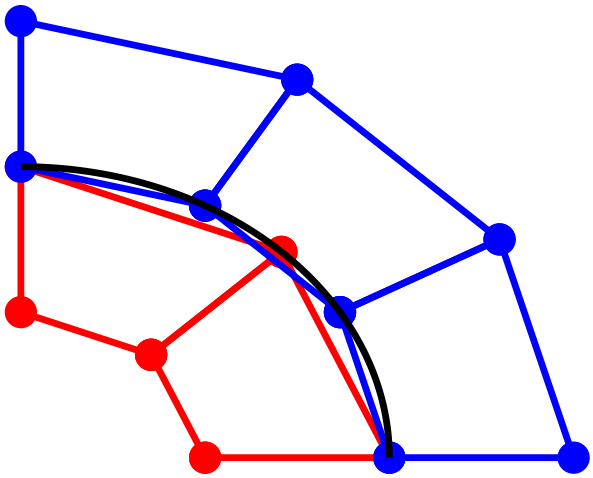


Fig. 4 A hp-nonconformal mesh illustrating the dependence of discrete geometry on the mesh resolution. The analytical shape of the interface is shown in black.

The SBP operator is defined similarly to Section III.C, except that B_{x_i} is defined to sum over all $M(L)$ elements. The properties established in that section hold for the hp-nonconformal operator, noting that the normal vectors $\mathsf{N}_{x,M}$ are scaled to the size of the M grid element.

We now briefly review the stability, conservation, and free-stream preservation properties of the discretization, which follow from the p-nonconformal discretization with minor modifications. Afterwards, several practical details are discussed, including issues that arise from hp-nonconformal curved interfaces interface.

A. Discretization Properties

1. Entropy Stability

Similar to (64), the volume entropy contribution is

$$\boldsymbol{w}_L^T \left(\mathsf{S}_x \circ \mathsf{F}_x(\boldsymbol{u}_L, \boldsymbol{u}_L)\right) \mathbf{1} = -\mathbf{1}^T \mathsf{B}_x \boldsymbol{\psi}_{x,L} = -\sum_{\gamma \in \Gamma_L} \sum_{M \in M(L)} \mathbf{1}^T \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_L \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L \boldsymbol{\psi}_{x,L}. \tag{95}$$

For the face term, (78) holds for each M grid element.

For the example mesh in Figure 3, the entropy balance for element L_1 will include (95), where M sums over M_1, and M_2, and the L parts of (78) for M_1 and M_2. Similarly, the entropy balance for L_2 will include the L parts of the M_3 and M_4 terms. Thus, the face term from every M grid element cancels with a portion of the volume term entropy, giving net entropy conservation.

2. Conservation

The proof of conservation is nearly identical to the p-nonconformal case in Section III.D.2. The volume term (81) remains the same. The face term derivation from (82) to (86) holds for each M grid element. Therefore, the material that exits a given L element via an M element is the exact same amount of material as enters the adjacent R element via the same M element, and we have conservation.

3. Free-Stream Preservation

The proof of free-stream preservation also follows from the p-nonconformal case in Section III.D.3. The volume term (88) holds. The face term derivation from (89) to (93) hold for each M grid element. Therefore, the result holds for all M grid elements, and we have free-stream preservation.

B. M Grid and $\mathsf{N}_{x,M}$ Calculation

This section describes how to construct the M grid and how to compute the normal vectors $\mathsf{N}_{x,M}$. For axis-aligned meshes such as Figure 3, this is straight-forward, however for curved mesh such as Figure 4 there is an additional complication. All mesh vertices are on the true interface curve (the unit circle in this example), however the discrete geometry, defined by the union of the element faces, is different for the two sides of the h-nonconformal interface. This leads to additional complexity in the algorithms to determine the M mesh topology and computing the $\mathsf{N}_{x,M}$. We impose the following requirement:

Requirement 1. *The M grid topology and $\mathsf{N}_{x,M}$ shall be independent of the the choice of L and R (up to a sign)*

This requirement is motivated by the notion that the number of blocks in a multi-block mesh is arbitrary, and therefore the $\mathsf{N}_{x,M}$ should not depend on it. This is a necessary condition for the discretization to be parallel consistent, that is, to give the same result when run in different numbers of parallel ranks.

The main consequence of Requirement 1 is that the $\mathsf{N}_{x,M}$ cannot be compute on either the L mesh or the R mesh, but must instead depend on both L and R in a symmetric manner. We now describe an algorithm for determining the topology of the M grid on curved interfaces, and then an algorithm for computing $\mathsf{N}_{x,M}$.

1. M Grid Topology

Given the set of vertices on the L and R vertices, the problem of determining the M grid topology reduces to ordering the points on the interface. For straight-sided meshes like Figure 3, the problem reduces to comparing the y coordinates of the vertices. For curved meshes, we use a 3 point comparison algorithm:

Definition 4 (Point Ordering)**.** The points p_1, p_2, and p_3 are considered ordered if $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 < 0$, where $\boldsymbol{v}_1 = p_1 - p_2$ and $\boldsymbol{v}_2 = p_3 - p_2$.

The definition is motivated by the idea that p_1, p_2 and p_3 lie on a parametric curve, and the ordering can be determined by considering the direction of the vectors from p_2 to p_1 and p_2 to p_3. If the vectors are in opposite directions, the points are considered ordered. Otherwise, p_3 comes before p_2 on the curve, and the points are not ordered.

Note that this ordering algorithm can fail for high curvature. Recalling that $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = \|\boldsymbol{v}_1\| \, \|\boldsymbol{v}_2\| \cos\theta$, if $\angle p_1 p_2 p_3$ is acute, the points will be considered ordered, even if the parametric curve traverses an angle $\frac{3\pi}{2} < \theta < 2\pi$.

Algorithm 1 uses an ordering predicate satisfying Definition 4 to compute the M grid topology. The algorithm uses the coordinates of the vertices of the L and R grid to determine the vertices of the M grid, as well as the connectivity between the L, M, and R grids. We note in passing that this algorithm satisfies Requirement 1, and therefore the R to M connectivity can be obtained by running the algorithm a second time with L and R reversed.

2. $\mathsf{N}_{x,M}$ and Parametric coordinate Calculation

The algorithm presented in this section computes two quantities, the normal vectors on the M grid, $\mathsf{N}_{x,M}$, and the parametric coordinates of the M grid vertices in the coordinate system of their associated L or R grid element (ie. the elements given by $elemM2L$ and $elemM2R$ from Algorithm 1). The normal vectors are clearly needed to evaluate discretization (94). The parametric coordinates are required to compute P_{LM} and P_{RM}, which are defined in reference space, not physical space. Section III.B shows how to construct the mapping $\boldsymbol{x}(\boldsymbol{\xi})$, however the inverse mapping $\boldsymbol{\xi}(\boldsymbol{x})$ is required. Additionally, for meshes such as Figure 4, the M grid vertex may not lie on the curve $\boldsymbol{x}(\boldsymbol{\xi})$, for example when searching for an R grid vertex on the L grid. Thus, the problem is to find the $\boldsymbol{\xi}$ value that gives the $\boldsymbol{x}(\boldsymbol{\xi})$ nearest to the given point $\boldsymbol{x}_\star$. The problem can be formally stated as

$$\min_{\boldsymbol{\xi}} \quad \|\boldsymbol{x}(\boldsymbol{\xi}) - \boldsymbol{x}_\star\|^2 . \tag{96}$$

This can be recognized as the well-known nonlinear least squares problem, and can be solved with the Gauss-Newton algorithm.

An initial guess can be computed by approximating $\boldsymbol{x}$ as a linear function of $\boldsymbol{\xi}$ and solving for the $\boldsymbol{\xi}$ value based on either the x or y coordinate, whichever has the largest magnitude $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$. This provides a reasonable initial guess while guarding against axis-aligned elements where either x or y is independent of $\boldsymbol{\xi}$.

Algorithm 1: Algorithm for computing M grid topology

Input : $vtxL$: coordinates of the vertices of the L faces
Input : $vtxR$: coordinates of the vertices of the R faces
Input : `isOrdered`: an ordering predicate that satisfies Definition 4
Input : ϵ: tolerance for merging vertices
Output $vtxM$: coordinates of the vertices of the M grid elements
:
Output $elemL2M$: array of arrays giving the M grid elements that overlap with each L grid element
:
Output $elemL2R$: array of arrays giving the R grid elements that overlap with each L grid element
:
Output $elemM2L$: array giving the L grid element that each M grid element overlaps with
:

```
1  Midx = 1; eR = 1;
2  for eL=1:length(vtxL) - 1 do
3  |  Get the lower and upper vertices, p_L1, and p_L2, of the current element from vtxL;
4  |
5  |  Append p_L1 to vtxM;
6  |  eRstart = eR; isequal = false; nM = 0;
   |  /* Process R grid faces that overlap with eL                              */
7  |  while true do
8  |  |  Get the lower and upper vertices, p_R1, and p_R2, of element eR from vtxR;
9  |  |  isequal = ||p_R2 - p_L2|| < ε;
10 |  |  isgreater = isOrdered (p_R1, p_L2, p_R2);
11 |  |  if isgreater or isequal then
12 |  |  |  if isequal then
   |  |  |  |  /* If the L and R elements end at the same point, the next element
   |  |  |  |     should start searching at the next R grid element              */
13 |  |  |  |  eR = eR + 1
14 |  |  |  end
15 |  |  |  break;
16 |  |  end
17 |  |  Append p_R2 to vtxM;
18 |  |  nM = nM + 1; eR = eR + 1;
19 |  end
20 |  eRend = eR;
21 |  if isequal then
22 |  |  eRend = eRend - 1;
23 |  end
24 |  Append range eRstart : eRend to elemL2R;
25 |  Append range Midx : (Midx + nM − 1) to elemL2M;
26 |  Append eL to elemM2L nM times;
27 |  Midx = Midx + nM;
28 end
```

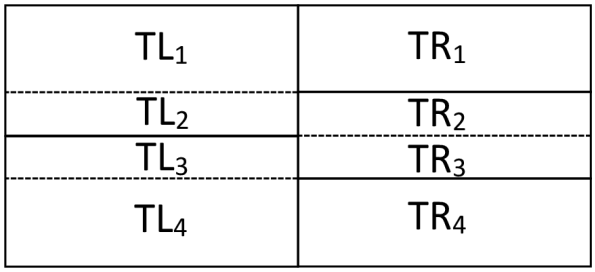


Fig. 5 The hp-nonconformal mesh from Figure 3 with the truncated elements labeled

Algorithm 2: Algorithm for computing $\mathsf{N}_{x,M}$, P_{LM}, P_{RM}

Input : ϵ: tolerance for merging vertices
Input : $vtxM$: coordinates of the vertices of the M grid elements
Input : $elemM2L$: array giving the L grid element that each M grid element overlaps with
Input : $elemM2R$: array giving the R grid element that each M grid element overlaps with
Output $\mathsf{N}_{x,M}$: normal vectors for each M grid element
:
Output P_{LM}, P_{RM}: interpolation operators for each M grid element
:

1 **for** *eM=1:length(elemM2L)* **do**
2 Get eL and eR from $elemM2L$ and $elemM2R$;
3 Compute $\boldsymbol{x}(\boldsymbol{\xi})$ for eL and eR (see Section III.B) ;
4 Get p_{M1} and p_{M2}, the coordinates of the vertices of eM, from $vtxM$;
5 Use search algorithm to compute $\boldsymbol{\xi}_{LM1}$, $\boldsymbol{\xi}_{LM2}$, the parametric coordinates of p_{LM1} and p_{LM2} in the coordinate system of eL, and $\boldsymbol{\xi}_{RM1}$, $\boldsymbol{\xi}_{RM2}$, in the coordinate system of eR;
6 Define coordinate nodes in the ranges $[\boldsymbol{\xi}_{LM1}, \boldsymbol{\xi}_{LM2}]$ and $[\boldsymbol{\xi}_{RM1}, \boldsymbol{\xi}_{RM2}]$ and evaluate $\boldsymbol{x}(\boldsymbol{\xi})$ at those nodes;
7 Average the physical coordinates of the face coordinate nodes;
8 Compute $\mathsf{N}_{x,M}$ on the truncated element, using the method described in Section III.B;
9 Compute P_{LM} and P_{RM}, using the coordinate nodes of eL and eR and the M grid integration rule shifted to $[\boldsymbol{\xi}_{LM1}, \boldsymbol{\xi}_{LM2}]$ and $[\boldsymbol{\xi}_{RM1}, \boldsymbol{\xi}_{RM2}]$;
10 **end**

An alternative approach for generating an initial guess is to construct a polynomial approximation to $\boldsymbol{\xi}(\boldsymbol{x})$. While this is not the precise inverse of $\boldsymbol{x}(\boldsymbol{\xi})$, it is a high-order approximation. One difficulty in constructing the approximation is that the nodes $\boldsymbol{x}$ do not have a tensor-product structure. One solution it to use algorithms that compute a polynomial basis on non-tensor product point sets [8]. Alternatively, a bounding box around the element can be established, and the coefficients of the tensor-product basis on the bounding box can be computed from the $\boldsymbol{x}$. For the meshes tested so far, the Gauss-Newton algorithm converges in 3 to 4 iterations when using the linear initial condition, and so the high-order inverse approximation is not used.

We now use this approach as a component of a larger algorithm to compute $\mathsf{N}_{x,M}$, P_{LM} and P_{RM}. The main idea of this algorithm is to define a so-called "truncated" element on the L and R grids, as shown in Figure 5. This truncated element has its own coordinate mapping of the same type described in Section III.B, however the parametric coordinates can also be expressed in the coordinate system of the "full" element. This allows computing the $\boldsymbol{x}$ coordinates of the truncated element nodes from the "full" element mapping $\boldsymbol{x}(\boldsymbol{\xi})$.

Once the $\boldsymbol{x}$ coordinates of each truncated element are known, the coordinates on the shared face are averaged. This ensures Requirement 1 is satisfied. Once the coordinates have been averaged, the normal vectors can be computed on either of the truncated elements. The algorithm is given in Algorithm 2.

The essential component of the algorithm is the computation of $\boldsymbol{\xi}_{LM1}$, $\boldsymbol{\xi}_{LM2}$, $\boldsymbol{\xi}_{RM1}$, and $\boldsymbol{\xi}_{RM2}$ This coordinates are necessary to compute P_{LM} and P_{RM}, which are defined in reference space, not physical space. Additionally, they are needed to compute the coordinates of the truncated element nodes from the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ of the L or R.

After the coordinates are averaged on line 7, the $\boldsymbol{x}$ coordinates of the truncated element may be different than the coordinates computed from the mapping $\boldsymbol{x}(\boldsymbol{\xi})$. Because of this, P_{LM} and P_{RM} will be interpolating to slightly different

points. As a result, the discretization may not be exact for polynomials when the geometry is not exactly represented by the mapping, even if the elements are straight-sided. The error associated with the averaging is degree q, so as long as $q \geq p$, this geometric error is not asymptotically worse than the discretization error.

As a final note, the computation of $\mathsf{N}_{x,M}$ on line 8 occurs on the truncated element, which results in normal vectors that are scaled to the size of the M grid elements.

V. Summary

This work has extended the method of [1] to operators with $2p-1$ quadratures. This allows constructing high-order accurate p- and hp-nonconformal discretizations that are conservative, entropy stable, and free-stream preserving on curvilinear grids. If Lemma 2 holds, then the discretization presented here is equivalent to the one in [1]. This requires B_M to exactly integrate twice the maximum degre polynomial on the interface. The metric projection trick is required for three dimensional elements, but not in two dimensions.

For elements with quadratures that are less than $2p$ exact, the stability requirements on the quadrature rules are:

- For straight-sided elements, Lemma 3 applied, which requires B_M to integrate the maximum degree polynomial on the interface
- For curvilinear elements, the metric projection trick (described in Appendix A) is required, and B_M must exactly integrate polynomials of degree $2(q-1)$, where q is the degree of the coordinate mapping.

The accuracy requirements are:

- Lemma 4 must hold, which requires B_M to be at least as accurate as the maximum quadrature on the L or R elements.

We now give guidance on how to couple the commonly-used Conventional and Generalized SBP operators. For curved elements:

- When coupling Conventional to either Conventional or Generalized, B_M should exactly integrate $2(q-1)$, for stability. For accuracy B_M should be at least as accurate as B_L or B_R, whichever is higher.
- When coupling Generalized to Generalized, B_M should exactly integrate twice the **maximum** degree operator, or $2(q-1)$, whichever is higher, for stability. For accuracy B_M should be at least as accurate as B_L or B_R, whichever is higher.

For straight-sided elements:

- When coupling Conventional to either Conventional or Generalized, B_M should exactly integrate the **maximum** degree operator for stability. For accuracy B_M should be at least as accurate as B_L or B_R, whichever is higher.
- When coupling Generalized to Generalized, B_M should exactly integrate twice the **maximum** degree operator for stability. For accuracy B_M should be at least as accurate as B_L or B_R, whichever is higher.

Numerical results indicate that couplings involving Conventional operators lose half an order of convergence for straight-sided elements and a full order of convergence for curvilinear elements. Coupling between Generalized retains full convergence rate (of the lowest degree operator) when Lemma 2 is satisfied.

VI. Numerical Results

In this section we present results for the linear advection equation in two dimensions, as well as the Euler equations. The results will demonstrate the accuracy and entropy stability of the discretization.

A. Linear Advection

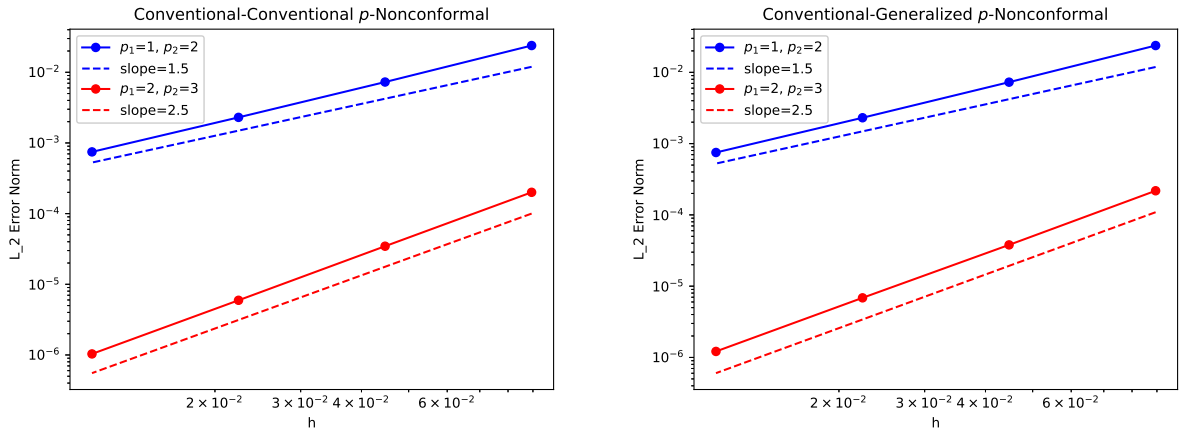
We solve the linear advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} + S = 0 \tag{97}$$

where a, b are the advection velocities and S is a source term used with manufactured solutions. The entropy stable fluxes are

$$f_x^\star(u_L, u_R) = \frac{a}{2}(u_L + u_R) \tag{98}$$

$$f_y^\star(u_L, u_R) = \frac{b}{2}(u_L + u_R) \tag{99}$$



(a) Conventional elements for all blocks

(b) Generalized elements in middle block, Conventional elsewhere

Fig. 6 Manufactured solution convergence results: p-nonconformal scheme with straight-sided elements

To test straight-sided elements, we use the manufactured solution

$$u(x,t) = \exp(x+y+t) \quad \text{for} \quad x, y \in \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{100}$$

To test curved elements, the same solution is used, and the unit square is mapped according to

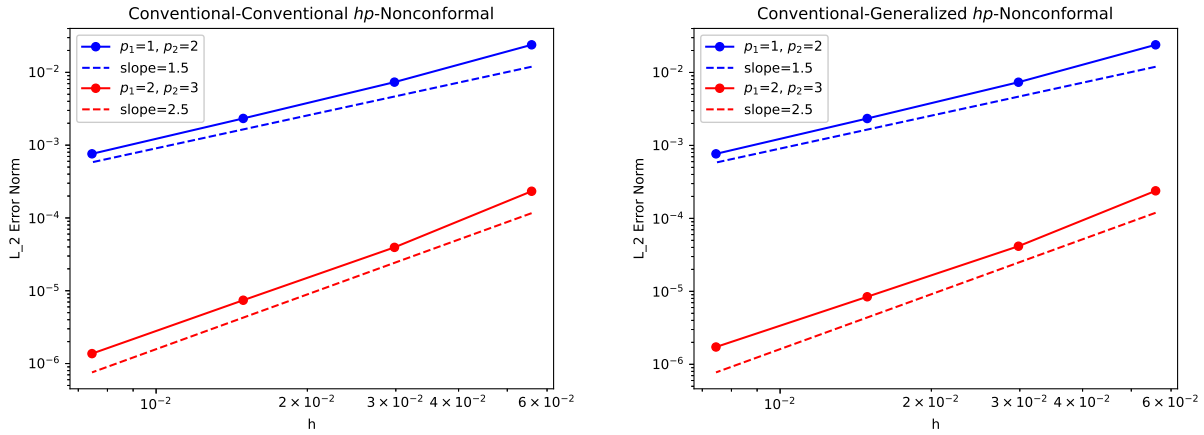
$$x_2 = x_1 + \alpha \sin(\frac{x_1 - c_x}{\beta_x}) \tag{101}$$

$$y_2 = y_1 + \alpha \sin(\frac{y_1 - c_y}{\beta_y}), \tag{102}$$

where x_1, y_1 are the coordinates in the unit square, x_2, y_2 are the mapped coordinates, $\alpha = 0.05$, $c_x = c_y = 0$, and$\beta_x = \beta_y = \frac{1}{\pi}$. Note that this only moves nodes in the interior of the domain, and not on the boundaries.

The classical forth-order explicit Runge-Kutta method is used to discretize the time derivative.

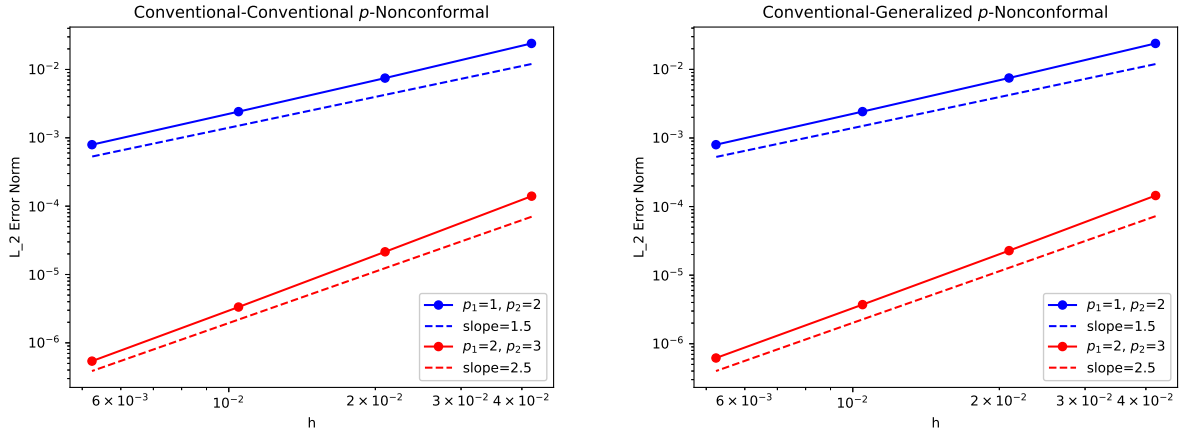
The results are presented in Figures 6 and 7 for straight-sided elements and Figures 8 and 9 for curved elements. Multi-block meshes are used for the convergence studies, with the blocks arranged in a $3x3$ grid. The first mesh has 5 elements per block in each direction, doubling the number of elements per direction in each subsequent mesh. The figures are labeled according to the element used in the blocks. The first label identifies the element used in all blocks except the middle one, and the second label identifies the element used in the middle block. Conventional refers to elements using the Legendre-Gauss-Lobatto (LGL) nodes, and Generalized refers to the Gauss-Legendre elements. Similarly, the figure legend identifies the element degree for all blocks except the middle as p_1, and p_2 for the middle block. In all cases, the element mapping is approximated by polynomials of degree $\min(p_1, p_2) + 1$. For hp-nonconformal meshes, the middle block has 1.5 times as many elements in each direction as the other blocks.



(a) Conventional elements for all blocks

(b) Generalized elements in middle block, Conventional elsewhere

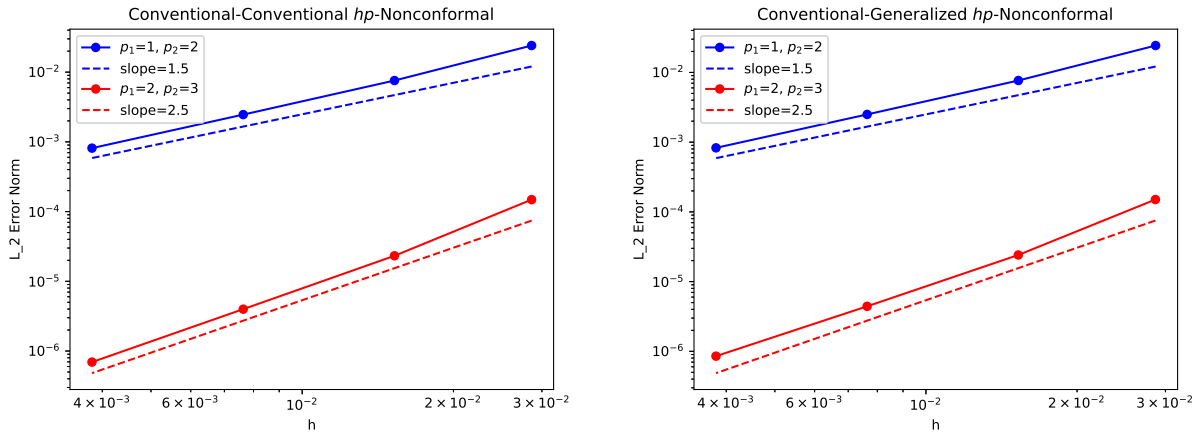
Fig. 7 Manufactured solution convergence results: hp-nonconformal with straight-sided elements



(a) Conventional elements for all blocks

(b) Generalized elements in middle block, Conventional elsewhere

Fig. 8 Manufactured solution convergence results: p-nonconformal with curved elements



(a) Conventional elements for all blocks

(b) Generalized elements in middle block, Conventional elsewhere

Fig. 9 Manufactured solution convergence results: hp-nonconformal with curved elements

B. Euler Equations

In this section, we solve the Euler equations

$$\frac{\partial u}{\partial t} + \frac{\partial f_i(u)}{\partial x_i} = 0, \tag{103}$$

where $\boldsymbol{u} = [\rho, \rho u, \rho v, E]$ are the conserved variables and $\boldsymbol{f}_i(\boldsymbol{u})$ are the fluxes. The calorically perfect ideal gas law is used to close the system.

When discretizing, the numerical flux function $f^\star(u_L, u_R)$ is the entropy-conservative flux of Chandrashekar [9] combined with an entropy-stable dissipation based on Merriam [10]. Tensor-product SBP operators based on the Legendre-Gauss-Lobatto (LGL) nodes are used for all cases. These SBP operators have nodes on the endpoint of the element, which makes the R_L and R_R operators trivial, however the P_{LM} and P_{RM} operators remain non-trivial because the M-grid quadrature points are distinct from the LGL points. Additionally, the quadrature rules for the LGL elements are exact for degree $2p-1$ polynomials, and so we expect convergence at a rate of p rather than $p+1$.

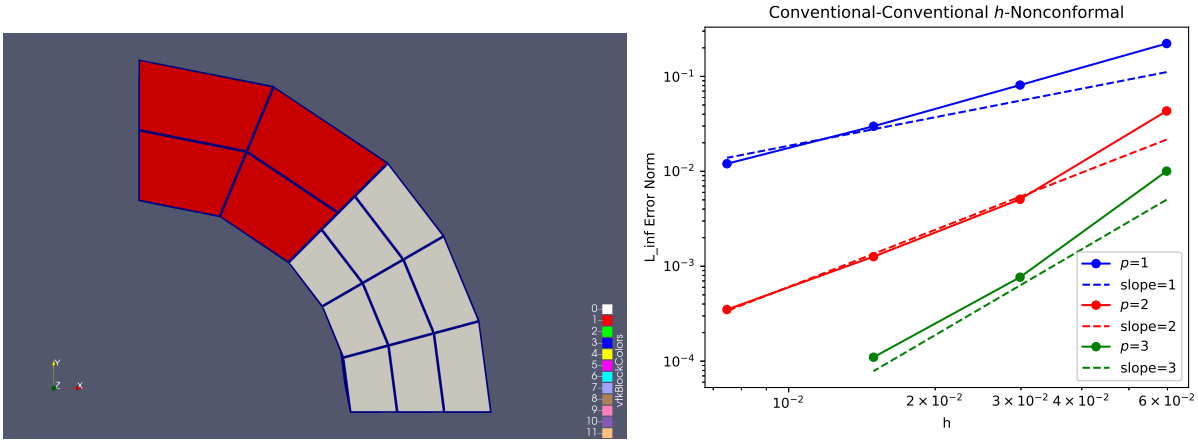
To test accuracy, we solve the steady vortex problem. The domain is a quarter annulus with $r \in [1.5, 2.5]$, and the exact solution is

$$\begin{aligned} f(r) &= 1 + \frac{1}{2}(\gamma-1)M_0^2\left(1 - \left(\frac{r_0}{r}\right)^2\right) \\ \rho(r) &= \rho_0 f(r)^{\frac{1}{\gamma-1}} \\ u(r,\theta) &= u_0 \frac{r_0}{r}\sin(\theta) \\ v(r,\theta) &= -u_0 \frac{r_0}{r}\cos(\theta) \\ T(r) &= T_0 f(r) \end{aligned} \tag{104}$$

The case is solved with $r_0 = 1$, $\rho_0 = 1$, $M_0 = 2.25$, and $T_0 = \frac{1}{\gamma R}$, where R is the gas constant. The final parameter u_0 can be computed from M_0 and the speed of sound.

The initial mesh is shown in Figure 10a Each subsequent mesh is created by uniform refinement. The exact solution is imposed on the boundary using a farfield condition. The convergence results are shown in Figure 10b for $p = 1$ to $p = 3$ operators, and show convergence rates of p for the $p = 2$ and $p = 3$ operators, and slightly greater than p for the $p = 1$.

To verify entropy stability, the unsteady vortex problem [11] is run in periodic domain. The exact solution is



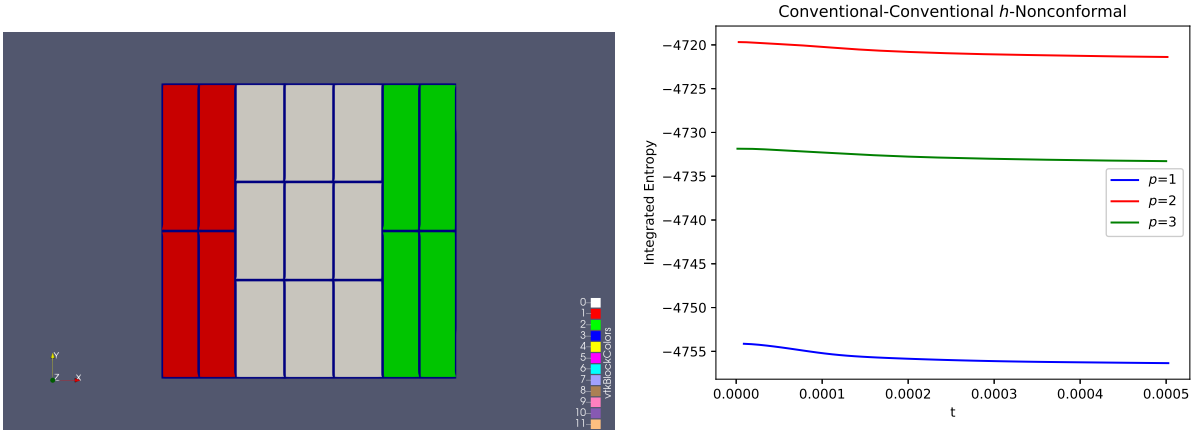
(a) First mesh **(b) Convergence results**

Fig. 10 Steady vortex

$$
\begin{aligned}
f(x,y,z,t) &= 1 - \left(\frac{x - x_0 - u_0 t}{r_0}\right)^2 \\
\rho(x,y,z,t) &= \rho_0 \left(1 - \left(\frac{\epsilon M_0}{\pi}\right)^2 \frac{\gamma - 1}{8} \exp(f(x,y,z,t))\right)^{\frac{1}{\gamma - 1}} \\
u(x,y,z,t) &= u_0 \left(1 - \epsilon \frac{y - y_0 - u_0 t}{2 r_0 \pi} \exp(\frac{f(x,y,z,t)}{2})\right) \\
v(x,y,z,t) &= u_0 \left(\epsilon \frac{x - x_0}{2 r_0 \pi} \exp(\frac{f(x,y,z,t)}{2})\right) \\
T(x,y,z,t) &= T_0 \left(1 - \left(\frac{\epsilon M_0}{\pi}\right)^2 \frac{\gamma - 1}{8} \exp(f(x,y,z,t))\right).
\end{aligned}
\tag{105}
$$

The domain is $x, y \in [0, 0.1]$, and the mesh is shown in Figure 11a. The left and right blocks form a 2×2 element grid, while the middle block is 3×3. This enables standard periodic boundary conditions to be applied between the left and right blocks (periodic nonconformal interfaces are possible, but not yet implemented in software).

The time derivative is discretized using the classical 4th order Runge-Kutta scheme, modified to be entropy stable [12]. Simulation is run to a maximum time of 0.005 at a CFL of 0.75, and the time history of the entropy integrated over the domain is shown in Figure 11b. As expected, the entropy is monotonically decreasing.



(a) Mesh **(b) Entropy history**

Fig. 11 Unsteady vortex

VII. Conclusions

We have presented a new method of discretizing nonconformal interfaces in a conservative, entropy-stable, and free-stream preserving manner. We have also shown that the inner-product preservation property is a sufficient condition for stability, but is not necessary if the face terms are evaluated with the M-grid integration rule. The inner-production preservation property does appear in the accuracy analysis, and leads to a loss of one order of convergence when it is not satisfied.

This method can be extended to curvilinear elements, by constructing an SBP operator in physical space by mapping the Q_{ξ_i} operator of a reference-space SBP operator into physical space, and combining it with the M-grid integration rule. Except for a few special cases where the exact metrics can be used, the metric projection trick must be employed to find metrics that satisfy the discrete metric invarients, and therefore allow the SBP operator to exactly differentiate constants.

Using the above ingredients, we have developed a method that allows discretizing nonconformal interfaces for SBP operators with $2p-1$ exact quadratures, even when the interface is curved, while retaining the desirable properties of the conformal scheme, namely conservation, entropy-stability, and free-stream preservation.

A. Metric Projection

The metric projection trick from [6] is written here for the case of hp-nonconformal interfaces. p-Nonconformal interfaces can be treated as a special case where there is only one M grid element per face.

Define the target metrics to be

$$\boldsymbol{m}_{\text{targ}} = \begin{bmatrix} \Lambda_{\xi_1,x_i}\mathbf{1} \\ \Lambda_{\xi_2,x_i}\mathbf{1} \end{bmatrix} \tag{106}$$

which contain the **exact** metrics. Now compute metrics $\boldsymbol{m}$ that are closest to $\boldsymbol{m}_{\text{targ}}$ while also satisfying (34):

$$\min_{\boldsymbol{m}} \quad (\boldsymbol{m}-\boldsymbol{m}_{\text{targ}})^T (\boldsymbol{m}-\boldsymbol{m}_{\text{targ}}) \tag{107}$$

$$\text{subject to} \quad \sum_j \left(-\mathsf{Q}_{\xi_j}^T \Lambda_{\xi_j,x_i}\mathbf{1}\right) + \sum_\gamma \mathsf{B}_{x_i}\mathbf{1} = \mathbf{0} \tag{108}$$

The solution is

$$\boldsymbol{m} = \boldsymbol{m}_{\text{targ}} - \begin{bmatrix} \mathsf{Q}_{\xi_1}^T & \mathsf{Q}_{\xi_2}^T \end{bmatrix}^\dagger \left(\begin{bmatrix} \mathsf{Q}_{\xi_1}^T & \mathsf{Q}_{\xi_2}^T \end{bmatrix} \begin{bmatrix} \Lambda_{\xi_1,x_i}\mathbf{1} \\ \Lambda_{\xi_2,x_i}\mathbf{1} \end{bmatrix} - \sum_\gamma \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L \mathbf{1} \right) \tag{109}$$

Because $\mathbf{1}$ is in the nullspace of Q_{ξ_j}, the compatibility condition must be satisfied:

$$\mathbf{1}^T \left(-\mathsf{Q}_{\xi_j}^T \Lambda_{\xi_j, x_i} \mathbf{1} \right) + \sum_\gamma \mathbf{1}^T \mathsf{B}_{x_i} \mathbf{1} = \mathbf{1}^T \mathbf{0} \tag{110}$$

$$\sum_\gamma \mathbf{1}^T \mathsf{R}_L^T \mathsf{P}_{LM}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathsf{P}_{LM} \mathsf{R}_L \mathbf{1} = 0 \tag{111}$$

$$\sum_\gamma \mathbf{1}^T \mathsf{B}_M \mathsf{N}_{x,M} \mathbf{1} = 0, \tag{112}$$

where we have used the property that R_L and P_{LM} exactly interpolation constants. The final line will hold if B_M can exactly integrate $\mathsf{N}_{x,M}$. In two dimensions, $\mathsf{N}_{x,M}$ is a degree $q-1$ polynomial for a degree for a degree q mapping. In three dimensions, $\mathsf{N}_{x,M}$ is a degree $2(q-1)$ polynomial.

Note that the equation for $\boldsymbol{m}$ has the form of a correct to $\boldsymbol{m}_{\text{targ}}$ by a quantity scaled by the constraint violation. Thus, if the conditions of Lemma 2 are satisfied, $\boldsymbol{m} = \boldsymbol{m}_{\text{targ}}$.

B. Gauss-Newton Algorithm

The Gauss-Newton Algorithm is summarized in this section. For a more detailed treatment, see [13].

Let $\boldsymbol{x}$ be a vector of length n denoting the variables, and let the objective function be

$$f(\boldsymbol{x}) = \frac{1}{2} \sum_j^m r_j^2(\boldsymbol{x}). \tag{113}$$

Note that minimizing $f(\boldsymbol{x})$ implies minimizing the magnitude of the $r_j(\boldsymbol{x})$, which are referred to as residuals. The residual Jacobian $\mathsf{J}(\boldsymbol{x})$ is defined as

$$(\mathsf{J}(\boldsymbol{x}))_{ij} = \frac{\partial r_i(\boldsymbol{x})}{\partial \boldsymbol{x}_j} \tag{114}$$

The Gauss-Newton iteration is then

$$\min_{\boldsymbol{p}} \frac{1}{2} \left\| \mathsf{J}(\boldsymbol{x}) \boldsymbol{p} + \boldsymbol{r}(\boldsymbol{x}) \right\| \tag{115}$$

$$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{p} \tag{116}$$

where $\boldsymbol{r}$ is the vector of the $r_j(\boldsymbol{x})$ The first equation is a linear least-squares problem, which can be solved with any of the standard methods. When J is large, it is important to use a numerically well conditioned method, such as the singular value decomposition, to solve the problem. For the problems considered in this work, J is small (1×2 for two dimensions or 2×3 in three dimensions). it may be adequate (and very computationally efficient) to solve the normal equations directly:

$$\mathsf{J}^T \mathsf{J} \boldsymbol{p} = -\mathsf{J}^T \boldsymbol{r} \tag{117}$$

Globalization strategies such as line searches and trust regions can be implemented to improve the robustness of the algorithm, although they have not proven necessary for the meshes considered thus far.

The convergence theory for Gauss-Newton is that $\mathsf{J}^T \boldsymbol{r}$ goes to zero provided J does not have any zero singular values and $\boldsymbol{p}$ satisfies the Wolfe conditions. For the meshes tested thus far, the Gauss-Newton problem can be solved to a tolerance of $\left\| \mathsf{J}^T \boldsymbol{r} \right\| < 10^{-14}$.

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] Shadpey, S., and Zingg, D., "Entropy-Stable Multidimensional Summation-by-Parts Discretizations on hp-Adaptive Curvilinear Grids for Hyperbolic Conservation Laws," *Journal of Scientific Computing*, 2020. https://doi.org/10.1007/s10915-020-01169-1.

[2] Kozdon, J., and Wilcox, L., "Stable Coupling of Nonconforming, High-Order Finite Difference Methods," *SIAM Journal on Scientific Computing*, Vol. 38, 2016, pp. A923–A952. https://doi.org/10.1137/15M1022823.

[3] Friedrich, L., Winters, A., Del Rey Fernández, D., Gassner, G., Parsani, M., and Carpenter, M., "An Entropy Stable h/p Non-Conforming Discontinuous Galerkin Method with the Summation-by-Parts Property," *Journal of Scientific Computing*, Vol. 77, 2018. https://doi.org/10.1007/s10915-018-0733-7.

[4] Thompson, J. F., Soni, B. K., and Weatherill, N. P., *Handbook of Grid Generation*, 1st ed., CRC Press, 1999.

[5] Kopriva, D., "Metric Identities and the Discontinuous Spectral Element Method on Curvilinear Meshes," *J. Sci. Comput.*, Vol. 26, 2006, pp. 301–327. https://doi.org/10.1007/s10915-005-9070-8.

[6] Crean, J., Hicken, J. E., Del Rey Fernández, D. C., Zingg, D. W., and Carpenter, M. H., "Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements," *Journal of Computational Physics*, Vol. 356, 2018, pp. 410 – 438. https://doi.org/https://doi.org/10.1016/j.jcp.2017.12.015, URL http://www.sciencedirect.com/science/article/pii/S0021999117308999.

[7] Carpenter, M., Fisher, T., Nielsen, E., and Frankel, S., "Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces," *SIAM Journal on Scientific Computing*, Vol. 36, 2014, pp. B835–B867. https://doi.org/10.1137/130932193.

[8] Hughes, T. J., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, 2000.

[9] Chandrashekar, P., "Kinetic Energy Preserving and Entropy Stable Finite Volume Schemes for Compressible Euler and Navier-Stokes Equations," *Communications in Computational Physics*, Vol. 14, No. 5, 2013, p. 12521286. https://doi.org/10.4208/cicp.170712.010313a.

[10] Merriam, M. L., "An Enttropy-Based Approach to Nonlinear Stability," Tech. rep., NASA, Ames Research Center, Moffett Field, California, 94035, March 1989.

[11] Erlebacher, G., Hussaini, M. Y., and Shu, C.-W., "Interaction of a shock with a longitudinal vortex," *Journal of Fluid Mechanics*, Vol. 337, 1997, pp. 129–153. https://doi.org/10.1017/S0022112096004880, URL http://journals.cambridge.org/article_S0022112096004880.

[12] Ranocha, H., Sayyari, M., Dalcin, L., Parsani, M., and Ketcheson, D. I., "Relaxation Runge–Kutta Methods: Fully Discrete Explicit Entropy-Stable Schemes for the Compressible Euler and Navier–Stokes Equations," *SIAM Journal on Scientific Computing*, Vol. 42, No. 2, 2020, pp. A612–A638. https://doi.org/10.1137/19M1263480, URL https://doi.org/10.1137/19M1263480.

[13] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2nd ed., Springer, 2006.