

ON MORTAR-BASED ENTROPY STABLE MODAL DISCONTINUOUS GALERKIN METHODS

JESSE CHAN, MARIO BENCOMO, DAVID C. DEL REY FERNANDEZ

Abstract. Entropy stable schemes provide a semi-discrete statement of stability for nonlinear conservation laws. Entropy stable nodal discontinuous Galerkin (DG) methods have been constructed using entropy conservative finite volume fluxes [1] and a summation-by-parts (SBP) finite difference framework [2, 3, 4, 5, 6, 7, 8]. Entropy stable DG schemes require evaluating entropy conservative numerical fluxes between combinations of volume and surface quadrature nodes. On tensor product elements with surface nodes aligned with volume nodes, flux evaluations are required only between “lines” of nodes, reducing the number of flux evaluations to $O(N)$ per node. However, because surface nodes are no longer aligned on non-conforming meshes, each volume node interacts each non-aligned surface node, increasing the number of flux evaluations to $O(N^{d-1})$ per node in d dimensions.

1. Introduction. Boilerplate introduction on high order + stability

The work presented here builds upon the skew-symmetric formulation introduced in [9], but focuses instead on geometrically non-conforming meshes of tensor product elements. Such meshes may arise when applying domain decomposition techniques to a complex geometry (e.g. meshing sub-domains independently) [10, 11] or performing local mesh refinement [12]. Entropy stable GLL collocation schemes have been constructed on non-conforming meshes in [13] using special SBP projection operators. We study an alternative mortar-based approach to non-conforming interfaces, which differs from [13] in that it is applicable to a broader class of ESDG methods (such as Gauss collocation schemes [8]). The proposed approach will also reduce costs for both GLL and Gauss collocation on 3D non-conforming hexahedral meshes.

2. Entropy stable modal discontinuous Galerkin formulations. The primary complication of non-conforming meshes is the presence of hanging nodes or T-junctions, which arise where one element shares a “non-conforming” face with two or more neighboring elements. These are handled naturally by DG discretizations through the use of composite quadrature rules on non-conforming faces [14, 15], which the proposed approach will utilize.

The skew-symmetric formulation (??) can be stably applied to non-conforming meshes using composite quadratures, but results in high computational costs. These costs are due to the structure of matrices present in the formulation (??), which can be rewritten in block matrix form as

$$(1) \quad M \frac{d\mathbf{u}_h}{dt} + \sum_{i=1}^d \left(\begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}^i - (\mathbf{Q}^i)^T & \mathbf{E}^T \mathbf{B}^i \\ -\mathbf{B}^i \mathbf{E} & \mathbf{0} \end{bmatrix} \circ \mathbf{F}_S^i \right) \mathbf{1} + \mathbf{V}_f^T \mathbf{B}^i \mathbf{f}_i^* \right) = 0.$$

The matrix \mathbf{F}_S^i is formed by computing $(\mathbf{F}_S^i)_{jk} = \mathbf{f}_S^i(\tilde{\mathbf{u}}_j, \tilde{\mathbf{u}}_k)$ on the fly. Each entry requires evaluating the entropy conservative dyadic flux \mathbf{f}_S^i , resulting in a large number of floating point operations [16]. However, due to the Hadamard product in (1), dyadic flux evaluations are only necessary between pairs of nodes corresponding to non-zero entries of $(\mathbf{Q}^i - (\mathbf{Q}^i)^T)$ and \mathbf{E} .

For quad and hex elements under tensor product volume quadrature, \mathbf{Q}^i is the Kronecker product of 1D differentiation and identity matrices, which induces sparsity. As a result, dyadic flux evaluations are only required between “lines” of volume nodes [3, 8]. Dyadic flux evaluations also follow the sparsity pattern of the matrix \mathbf{E} , which maps from volume nodes to surface nodes. For GLL or Gauss collocation methods on conforming quad/hex meshes, \mathbf{E} is also sparse when surface quadrature nodes are

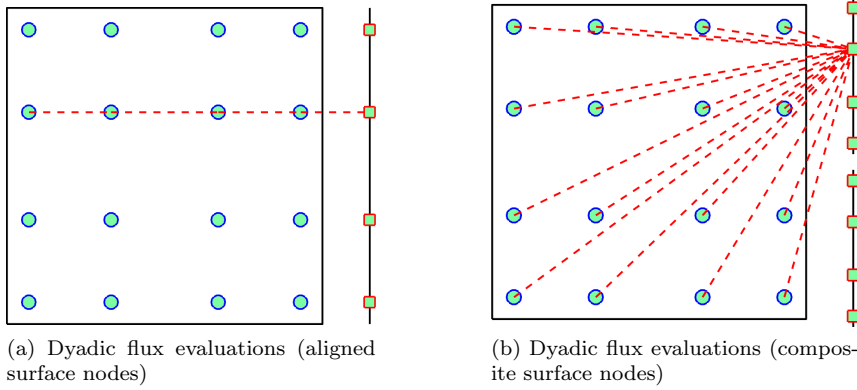


Fig. 1: Figure 1a and 1b illustrate when dyadic flux evaluations $f_S^i(\tilde{u}_j, \tilde{u}_k)$ between nodes are necessary (indicated by dashed red lines) for conforming interfaces (Figure 1a) and non-conforming interfaces (Figure 1b) under a Gauss collocation scheme. Conforming interfaces (aligned surface nodes) require evaluations between a surface node and a line of volume nodes, while non-conforming interfaces (nonaligned composite surface nodes) require flux evaluations between a surface node and *all* volume nodes. The proposed mortar approach (Figure ??) reduces the number of dyadic flux evaluations by introducing composite nodes as mortars which couple only to surface nodes.

aligned with volume quadrature nodes [8]. In such cases, dyadic flux evaluations are required only between lines of volume nodes and adjacent surface nodes (Figure 1a).

Composite quadrature rules on non-conforming meshes, however, are not aligned with volume nodes. It can be shown that the resulting matrix \mathbf{E} is fully dense, and evaluating the formulation (1) requires dyadic flux evaluations between each surface node and *all* volume nodes (as illustrated in Figure 1b). This greatly increases computational costs, especially at high orders of approximation. The proposed work will reduce the number of dyadic flux evaluations required for such composite quadratures. The increased cost of (1) under composite quadrature is due to the lack of sparsity in \mathbf{E} , which interpolates from volume to surface (composite) nodes. The proposed work avoids this issue by treating composite quadrature nodes as “mortar” nodes which are coupled directly to surface nodes but not directly to volume nodes, as illustrated in Figure ??.

3. Mortar methods for hybrid and non-conforming meshes. This work is motivated by complications which arise when designing entropy stable couplings between elements which do not share the same boundary nodes. This can arise, for example, for hybrid and non-conforming meshes.

Hybrid meshes. entropy stable DG methods on hybrid meshes were introduced in [9] using a skew-symmetric formulation. The resulting methods are stable for more arbitrary choices of surface quadrature, in particular when an SBP property may not hold.

Non-conforming meshes. On conforming meshes, it is most efficient to utilize both Gauss quadrature for volume integrals and Gauss quadrature for face or surface integrals. For solutions represented in terms of their values at tensor product volume

Gauss nodes, extrapolation to face Gauss nodes can be done in an efficient line-by-line manner using one-dimensional interpolation matrices.

For non-conforming meshes, it can be advantageous to use composite Gauss quadratures on non-conforming interfaces [15]. However, interpolating the solution at volume Gauss nodes to split-side Gauss nodes is no longer a one-dimensional operation.

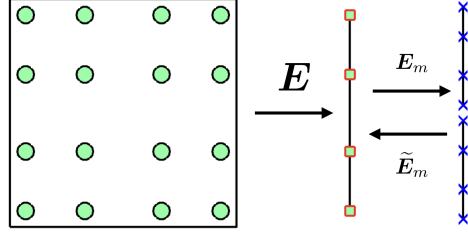


Fig. 2: Illustration of mortar operators for a Gauss collocation scheme. The matrix \mathbf{E} maps from volume quadrature points to surface quadrature points, \mathbf{E}_m maps from surface quadrature points to mortar surface points, and $\tilde{\mathbf{E}}_m$ maps from mortar surface points to surface points.

4. Mortar formulation. Let \mathbf{V}_q and \mathbf{V}_f denote interpolation matrices which evaluate at volume and surface quadrature points, respectively, and let \mathbf{W}, \mathbf{W}_f denote diagonal matrices whose entries consist of volume and surface quadrature weights

add defs of matrices

Let $\mathbf{E} = \mathbf{V}_f \mathbf{P}_q$ denote the polynomial mapping which extrapolates values from volume quadrature points to values at surface quadrature points. We also define \mathbf{B}_i as the diagonal matrix containing products of quadrature weights and the i th component of the outward normal vector $\hat{\mathbf{n}}_i$

$$\mathbf{B}_i = \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i).$$

An entropy stable skew-symmetric formulation can be given on the reference element \hat{D} as follows:

$$\mathbf{M} \frac{d\mathbf{u}_N}{dt} + \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}_i - \mathbf{Q}_i^T & \mathbf{E}^T \mathbf{B}_i \\ -\mathbf{B}_i \mathbf{E} & \mathbf{0} \end{bmatrix} \circ \mathbf{F}_S \right) \mathbf{1} + \mathbf{V}_f^T \mathbf{B}_i \mathbf{f}_i^* = 0.$$

We incorporate mortars by modifying this formulation. Let \mathbf{V}_m denote the matrix which maps volume quadrature points to values at surface *mortar* points. We also need to introduce interpolation operators $\mathbf{T}_f, \mathbf{T}_m$ for surface trace spaces. Here, \mathbf{T}_f maps from polynomials on the surface $\partial \hat{D}$ to surface quadrature points, while \mathbf{T}_m maps from polynomials on $\partial \hat{D}$ to mortar quadrature points. We can then define surface and mortar mass and projection matrices

$$\begin{aligned} \mathbf{M}_m &= \mathbf{T}_m^T \mathbf{W}_m \mathbf{T}_m, & \mathbf{P}_m &= \mathbf{M}_m^{-1} \mathbf{T}_m^T \mathbf{W}_m \\ \mathbf{M}_f &= \mathbf{T}_f^T \mathbf{W}_f \mathbf{T}_f, & \mathbf{P}_f &= \mathbf{M}_f^{-1} \mathbf{T}_f^T \mathbf{W}_f. \end{aligned}$$

Note that we have used the mortar mass matrix \mathbf{M}_m in the above definition of the face projection. This is only necessary if the surface and mortar quadratures do not exactly integrate degree $2N$ polynomials. If both the surface and mortar quadratures are exact for polynomials of degree $2N$, then $\mathbf{M}_m = \mathbf{M}_f$, and no distinction is necessary between the two mass matrices.

REMARK. *In order to show stability, we require that both \mathbf{P}_f and \mathbf{P}_m are defined using the same mass matrix. We have used the mortar mass matrix \mathbf{M}_m here; however, we could also use the surface mass matrix \mathbf{M}_f in both \mathbf{P}_f and \mathbf{P}_m . We use the mortar mass matrix in this work, as the mortar quadrature is generally more accurate than the surface quadrature for our use cases (e.g. hybrid and non-conforming meshes).*

We can now define operators which map between surface and mortar quadrature points. Let \mathbf{E}_m denote the map from surface to mortar points, and let $\tilde{\mathbf{E}}_m$ denote the map from mortar to surface points. Both operators are defined through an L^2 projection to the trace space and interpolation to appropriate points

$$\mathbf{E}_m = \mathbf{T}_m \mathbf{P}_f, \quad \tilde{\mathbf{E}}_m = \mathbf{T}_f \mathbf{P}_m.$$

We also define a mortar boundary matrix

$$\tilde{\mathbf{B}}_i = \mathbf{W}_m \text{diag}(\hat{\mathbf{n}}_i).$$

Then, a mortar-based formulation can be given as follows:

$$(2) \quad \mathbf{M} \frac{d\mathbf{u}_N}{dt} + \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \\ \mathbf{V}_m \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}_i - \mathbf{Q}_i^T & \mathbf{E}^T \mathbf{B}_i \\ -\mathbf{B}_i \mathbf{E} & \mathbf{B}_i \tilde{\mathbf{E}}_m \\ -\tilde{\mathbf{B}}_i \mathbf{E}_m & \end{bmatrix} \circ \mathbf{F}_S \right) \mathbf{1} + \mathbf{V}_m^T \tilde{\mathbf{B}}_i \mathbf{f}_i^* = 0.$$

Here, we have appended an extra row and column to the decoupled SBP matrix, and the inter-element numerical flux \mathbf{f}^* is now computed in terms of the *mortar* nodes.

We first note that

$$\begin{aligned} \mathbf{B}_i \tilde{\mathbf{E}}_m &= \text{diag}(\mathbf{W}_f) \mathbf{T}_f \mathbf{P}_m = \text{diag}(\hat{\mathbf{n}}) \mathbf{W}_f \mathbf{T}_f \mathbf{M}_m^{-1} \mathbf{T}_m \mathbf{W}_m \\ &= \mathbf{W}_f \mathbf{T}_f \mathbf{M}_m^{-1} \mathbf{T}_m^T \mathbf{W}_m \text{diag}(\hat{\mathbf{n}}_i) = \mathbf{P}_f^T \mathbf{T}_m^T \tilde{\mathbf{B}}_i = (\tilde{\mathbf{B}}_i \mathbf{E}_m)^T \end{aligned}$$

where we have used that \mathbf{W}_m is diagonal and that the outward normal $\hat{\mathbf{n}}_i$ on the reference element is constant over each face. This implies skew-symmetry of the decoupled SBP operator and entropy stability of the overall system.

[Finish](#)

5. A mortar-based implementation. While the formulation presented above is convenient for analysis, it is computationally expensive to implement. However, using properties of the mortar matrices $\mathbf{E}_m, \tilde{\mathbf{E}}_m$, we can rewrite the above formulation in a way which reflects a traditional mortar-based finite element implementation. In other words, we wish to implement (2) such that the only modification from the implementations in [7] is an intermediate step on mortar faces.

We first note that, since \mathbf{E}_m maps from surface nodes to mortar nodes, $\mathbf{V}_m = \mathbf{E}_m \mathbf{V}_f$. Thus, we can rewrite the numerical flux contribution as

$$\begin{aligned} \mathbf{V}_m^T \tilde{\mathbf{B}}_i \mathbf{f}_i^* &= \mathbf{V}_f^T \mathbf{E}_m^T \mathbf{W}_m \text{diag}(\hat{\mathbf{n}}_i) \mathbf{f}_i^* = \mathbf{V}_f^T \mathbf{P}_f^T \mathbf{T}_m^T \mathbf{W}_m \text{diag}(\hat{\mathbf{n}}_i) \mathbf{f}_i^* \\ &= \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{T}_m \mathbf{M}_m^{-1} \mathbf{T}_m^T \mathbf{W}_m \mathbf{f}_i^* = \mathbf{V}_f^T \mathbf{B}_i \tilde{\mathbf{E}}_m \mathbf{f}_i^*. \end{aligned}$$

We can similarly reformulate the remaining mortar contributions within (2). We decompose \mathbf{F}_S into interactions between volume nodes, surface nodes, and mortar nodes

$$\mathbf{F}_S = \begin{bmatrix} \mathbf{F}_S^{vv} & \mathbf{F}_S^{vs} & \mathbf{F}_S^{vm} \\ \mathbf{F}_S^{sv} & \mathbf{F}_S^{ss} & \mathbf{F}_S^{sm} \\ \mathbf{F}_S^{mv} & \mathbf{F}_S^{ms} & \mathbf{F}_S^{mm} \end{bmatrix}$$

Then, the on-element contributions to (2) can be expanded as follows

$$\begin{aligned} & \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \\ \mathbf{V}_m \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}_i - \mathbf{Q}_i^T & \mathbf{E}^T \mathbf{B}_i \\ -\mathbf{B}_i \mathbf{E} & \mathbf{B}_i \tilde{\mathbf{E}}_m \end{bmatrix} \circ \mathbf{F}_S \right) \mathbf{1} \\ &= \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}_i - \mathbf{Q}_i^T & \mathbf{E}^T \mathbf{B}_i \\ -\mathbf{B}_i \mathbf{E} \end{bmatrix} \circ \mathbf{F}_S \right) \mathbf{1} \\ &+ \mathbf{V}_f^T \left(\mathbf{B}_i \tilde{\mathbf{E}}_m \circ \mathbf{F}_S^{sm} \right) \mathbf{1} - \mathbf{V}_m^T \left(\tilde{\mathbf{B}}_i \mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1} \end{aligned}$$

Since multiplication by diagonal matrices $\mathbf{B}_i, \tilde{\mathbf{B}}_i$ is associative under the Hadamard product, the latter terms can be rewritten as

$$\begin{aligned} \mathbf{V}_f^T \left(\mathbf{B}_i \tilde{\mathbf{E}}_m \circ \mathbf{F}_S^{sm} \right) \mathbf{1} &= \mathbf{V}_f^T \mathbf{B}_i \left(\tilde{\mathbf{E}}_m \circ \mathbf{F}_S^{sm} \right) \mathbf{1} \\ \mathbf{V}_m^T \left(\tilde{\mathbf{B}}_i \mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1} &= \mathbf{V}_m^T \tilde{\mathbf{B}}_i \left(\mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1} = \mathbf{V}_f^T \mathbf{B}_i \tilde{\mathbf{E}}_m \left(\mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1} \end{aligned}$$

Then, (2) can be rewritten as

$$\begin{aligned} (3) \quad \mathbf{M} \frac{d\mathbf{u}_N}{dt} + \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{Q}_i - \mathbf{Q}_i^T & \mathbf{E}^T \mathbf{B}_i \\ -\mathbf{B}_i \mathbf{E} \end{bmatrix} \circ \mathbf{F}_S \right) \mathbf{1} + \mathbf{V}_f^T \mathbf{B}_i \tilde{\mathbf{f}}_i^* &= 0 \\ \tilde{\mathbf{f}}_i^* &= \tilde{\mathbf{E}}_m \mathbf{f}_i^* + \left(\tilde{\mathbf{E}}_m \circ \mathbf{F}_S^{sm} \right) \mathbf{1} - \tilde{\mathbf{E}}_m \left(\mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1} \end{aligned}$$

If the surface and mortar nodes are identical, then

$$\mathbf{E}_m = \tilde{\mathbf{E}}_m = \mathbf{I}, \quad \left(\tilde{\mathbf{E}}_m \circ \mathbf{F}_S^{sm} \right) \mathbf{1} = \left(\mathbf{E}_m \circ \mathbf{F}_S^{ms} \right) \mathbf{1}$$

and the correction terms within $\tilde{\mathbf{f}}^*$ cancel out, and we recover the original scheme from [7].

Note that in (3), the only interactions between surface and mortar nodes occurs within the computation of $\tilde{\mathbf{f}}_i^*$, which can be done solely on mortar interfaces. The computational structure of

1. Compute volume contributions to the RHS
2. Communicate surface contributions to mortar interfaces and compute $\tilde{\mathbf{f}}_i^*$.
3. Communicate $\tilde{\mathbf{f}}_i^*$ back to each element.

6. Error estimates. Weak derivative vs strong derivative error estimates. SBP yields that the skew-symmetric form is equivalent to weak form.

Line DG approach can work, but it's flipped: we increase quadrature strength in the direction *orthogonal* to derivative instead. May improve accuracy over GLL by giving equivalence w/skew form.

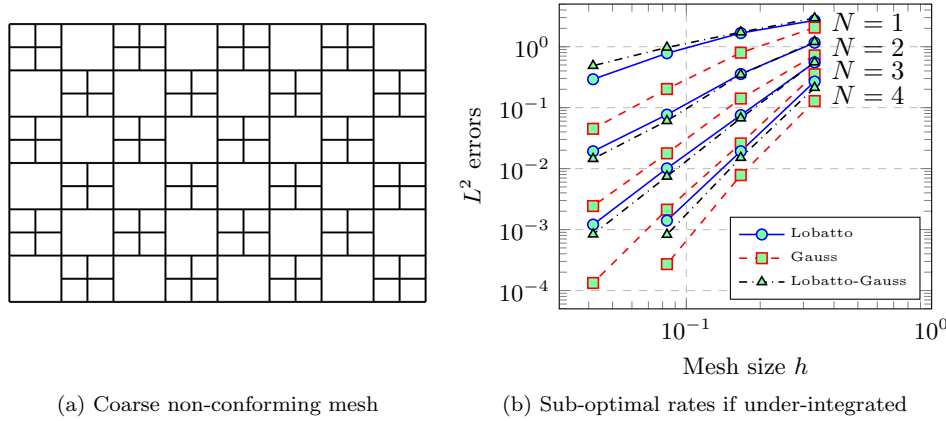


Fig. 3: Blah

7. Numerical experiments. Figure ?? shows preliminary results for the 2D compressible Euler equations by the PI on a manually constructed non-conforming quadrilateral mesh, with volume, surface, and mortar quadratures constructed from one-dimensional GLL or Gauss quadrature rules. The PI has verified that this preliminary implementation of the mortar-based formulation is discretely entropy stable. A preliminary accuracy analysis for the skew-symmetric formulation (??) also suggests that GLL quadrature should achieve an $O(h^N)$ rate of convergence and that Gauss quadrature should achieve an $O(h^{N+1})$ rate of convergence. Both predicted rates are confirmed in Figure ??.

7.1. Gauss Lobatto quadrature.

7.2. Gauss quadrature.

REFERENCES

- [1] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.
- [2] Gregor 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.
- [3] Mark H Carpenter, Travis C Fisher, Eric J Nielsen, and Steven H Frankel. Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces. *SIAM Journal on Scientific Computing*, 36(5):B835–B867, 2014.
- [4] Gregor J Gassner, Andrew R Winters, and David A Kopriva. Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *Journal of Computational Physics*, 327:39–66, 2016.
- [5] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.
- [6] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, 2018.
- [7] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346 – 374, 2018.
- [8] Jesse Chan, David C Fernandez, and Mark H Carpenter. Efficient entropy stable Gauss collocation methods. *arXiv preprint arXiv:1809.01178*, 2018.

- [9] Jesse Chan. Skew-symmetric entropy stable discontinuous Galerkin formulations. 2019. In preparation.
- [10] Yvon Maday, Cathy Mavriplis, and Anthony Patera. Non-conforming mortar element methods: application to spectral discretizations. In Tony Chan, Roland Glowinski, Jacques Periaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 392–418. SIAM, 1989.
- [11] Christine Bernardi, Yvon Maday, and Anthony T Patera. Domain decomposition by the mortar element method. In *Asymptotic and numerical methods for partial differential equations with critical parameters*, pages 269–286. Springer, 1993.
- [12] Matthew L Staten, Jason F Shepherd, Franck Ledoux, and Kenji Shimada. Hexahedral mesh matching: Converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces. *International journal for numerical methods in engineering*, 82(12):1475–1509, 2010.
- [13] Lucas Friedrich, Andrew R Winters, David C Fernández, Gregor J Gassner, Matteo Parsani, and Mark H Carpenter. An Entropy Stable h/p Non-Conforming Discontinuous Galerkin Method with the Summation-by-Parts Property. *arXiv preprint arXiv:1712.10234*, 2017.
- [14] Tan Bui-Thanh and Omar Ghattas. Analysis of an hp-nonconforming discontinuous Galerkin spectral element method for wave propagation. *SIAM Journal on Numerical Analysis*, 50(3):1801–1826, 2012.
- [15] Jeremy E. Kozdon and Lucas C. Wilcox. An Energy Stable Approach for Discretizing Hyperbolic Equations with Nonconforming Discontinuous Galerkin Methods. *Journal of Scientific Computing*, 2018.
- [16] Niklas Wintermeyer, Andrew R Winters, Gregor J Gassner, and Timothy Warburton. An entropy stable discontinuous Galerkin method for the shallow water equations on curvilinear meshes with wet/dry fronts accelerated by GPUs. *arXiv preprint arXiv:1804.02221*, 2018.