

We thank both Reviewers 1 and 2 for their feedback. We describe steps taken to address reviewer comments and suggestions, which are described in the following response. We have also edited a few additional sections of the manuscript to improve their readability, and have updated the numerical results of Figure 5, which were previously incorrectly reported.

Revisions in the manuscript are also colored for ease of identification. We hope these revisions improve the readability of this paper and its suitability for the audience of SISC.

## 1 Reviewer 1

- The paper is well-written for the most part (authors should adjust a few grammatical typos throughout their paper), and is interesting. It would be good if the authors would add a few more numerical tests with model problems in domains with complex geometry where curved meshes are actually needed (at this point, authors considered either rectangular or parallelepiped domain).

We thank the reviewer for their comments. We agree that testing these methods on curved unstructured meshes is necessary to evaluate their performance, but since this is currently being done in the context of a separate ongoing project, we wish to restrict the scope of the numerical experiments in this paper. We have added the following remarks in the conclusion noting that the proposed method can be extended to unstructured curved meshes:

We note that, while the numerical experiments presented here consider only mapped Cartesian domains, the method is also applicable to complex geometries, and future work will focus on studying the performance of such methods on curvilinear quadrilateral and hexahedral unstructured meshes.

## 2 Reviewer 2

- Page 7, eqn (3.8): why this equation holds? Can you add some comment here?

Certainly. We've added a continuous description of the equation, and have interpreted the matrix version as a quadrature approximation of a specific variational problem for the approximation of the derivative. This variational problem includes boundary correction terms, which correspond to the matrix "strong form" of (3.8). The added text is as follows:

More specifically, the expression (3.8) corresponds to a quadrature approximation of the following variational approximation of the derivative: find  $u \in P^N([-1, 1])$  such that,

$$\int_{-1}^1 u(x)v(x) = \int_{-1}^1 I_N f \frac{\partial I_N g}{\partial x} v + \left( (g - I_N g) \frac{(fv + I_N(fv))}{2} \right) \Big|_{-1}^1, \quad \forall v \in P^N([-1, 1]),$$

where  $I_N$  denotes the degree  $N$  polynomial approximation at the  $(N + 1)$  Gauss points.

- Page 10, Theorem 3.4: In the proof, this leads to a extra coefficient  $1/2$  in front of the second term of the equation below (3.14).

We thank the reviewer for catching this, and have fixed this in the revised version.

- In Theorem 3.4, it presents an semi-discrete entropy conservation for a two-element mesh. How to extend this to general case with multiple elements?

The block structure of the two-element mesh can be extended to multiple elements. We have added a discussion involving a three-element mesh on pages 11-12, which illustrates the procedure without introducing too much additional notation. We also describe how the extension to multiple elements approach requires a global SBP operator over the entire mesh, which is constructed in a similar block-wise fashion. We hope this revision makes the extension to multiple elements more clear.

- In the statement, boundary condition is not considered. How to utilize this theorem to solve the initial-boundary value problem numerically?

We have added a short discussion on pages 12-13 describing how to impose boundary conditions in a weak fashion through a boundary numerical flux. If the boundary numerical flux satisfies an entropy stability property, then the discretization with boundary conditions is also entropy stable. The added text is as follows:

If the boundary numerical flux satisfies the entropy stability conditions

$$\psi_L - \mathbf{v}_L^T \mathbf{f}_L^* \leq 0, \quad \psi_R - \mathbf{v}_R^T \mathbf{f}_R^* \leq 0$$

then the resulting scheme satisfies a global entropy inequality [1].

- A pseudo code showing how to solve 1D conservation law would be very helpful.

We have added a description of how to numerically solve the proposed system of ODEs using explicit time-stepping. With the exception of the evaluation of the nonlinear term, the procedure is relatively standard for time-explicit DG formulations. The added text is as follows:

Assuming that the ODE system (3.14) exactly integrated in time and that entropy dissipative numerical fluxes and boundary conditions are used, the solution will satisfy a discrete entropy inequality (2.3). In practice, the system (3.14) is solved using an ODE time-stepper. For an explicit time-stepper, then all that is necessary is to invert the diagonal matrix  $\mathbf{W}_h$  and evaluate the spatial terms in (3.15).

- Page 19, Section 5.3: this problem is solved on the triangular mesh. How to generalized the proposed scheme from quadrilateral and hexahedral meshes to triangular mesh?

We apologize for the confusion. Here, “triangulated” simply means “meshed”. The meshes considered are all quadrilateral and hexahedral, with triangular and simplicial meshes treated in a separate work.

- Page 4, line 152: Does the matrix size of  $V_f$  equal to  $(N + 1) \times 2$  or  $2 \times (N + 1)$ ?

We thank the reviewer for catching this - it is indeed  $2 \times (N + 1)$ . We have fixed this in the revised manuscript.

- Page 14. line 495:  $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_d)$ . Is  $\mathbf{u}_i$  a vector or a scalar?

We apologize for the confusion.  $\mathbf{u}_i$  should be a scalar, and we have edited the paper to clear this up. We no longer use bold to denote velocity  $u_i$  in the revised manuscript.

- Page 15, line 518-522:  $u$  and  $v$  here represent the velocity in  $x$  and  $y$ -direction, respectively. These conflict with the notations in previous sections, where,  $u$  is the conservative variable and  $v$  is the entropy variable. Definition of  $\{\{\cdot\}\}$  should be added here, too.

We have added definitions of the average and logarithmic mean. We have also removed  $u, v$  notation and replaced it with  $u_i$  notation, where  $u_i$  denotes the scalar velocity in the  $i$ th coordinate direction.

## References

- [1] 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.