

CONVEX OPTIMIZATION BASED STRUCTURE-PRESERVING FILTER FOR MULTIDIMENSIONAL FINITE ELEMENT SIMULATIONS

VIDHI ZALA, ROBERT M. KIRBY, AND AKIL NARAYAN

ABSTRACT. In simulation sciences, it is desirable to capture the real-world problem features as accurately as possible. Methods popular for scientific simulations such as the finite element method (FEM) and finite volume method (FVM) use piecewise polynomials to approximate various characteristics of a problem, such as the concentration profile and the temperature distribution across the domain. Polynomials are prone to creating artifacts such as Gibbs oscillations while capturing a complex profile. An efficient and accurate approach must be applied to deal with such inconsistencies in order to obtain accurate simulations. This often entails dealing with negative values for the concentration of chemicals, exceeding a percentage value over 100, and other such problems. We consider these inconsistencies in the context of partial differential equations (PDEs). We propose an innovative filter based on convex optimization to deal with the inconsistencies observed in polynomial-based simulations. In two or three spatial dimensions, additional complexities are involved in solving the problems related to structure preservation. We present the construction and application of a structure-preserving filter with a focus on multidimensional PDEs. Methods used such as the Barycentric interpolation for polynomial evaluation at arbitrary points in the domain and an optimized root-finder to identify points of interest improve the filter efficiency, usability, and robustness. Lastly, we present numerical experiments in 2D and 3D using discontinuous Galerkin formulation and demonstrate the filter's efficacy to preserve the desired structure. As a real-world application, implementation of the mathematical biology model involving platelet aggregation and blood coagulation has been reviewed and the issues around FEM implementation of the model are resolved by applying the proposed structure-preserving filter.

1. INTRODUCTION

A widely used application of mathematical modeling is to create simulations used to predict and analyze different processes. Simulations obtained from numerical solutions provide physically meaningful results if the values of simulation variables at each timestep follow the *structure* of the exact solution. Structure in this context refers to properties such as non-negative values for a chemical concentration or simulating a quantity requiring an upper bound, e.g., $[0,1]$, and other such aspects of the numerical solution. If the solution fails to comply with these structural restrictions, the resulting solution is not meaningful. Therefore, designing an approach to regulate the solution is crucial for simulations in many applications. Examples include thermodynamics, numerical weather predictions, bio-chemical processes such as platelet aggregation and blood coagulation, and fluid flow problems.

Computing solutions to mathematical models described in multiple dimensions further adds to the complexity of solving a structure-preservation problem. To implement multidimensional models using numerical methods, we require approximations of the quantities of interest in the domain. Popular choices for numerical methods such as the finite element method (FEM) and the finite volume method (FVM) employ piecewise polynomial-based approximations. These approximations are prone to artifacts such as the Gibbs phenomenon, leading to a violation of structure. We focus on the FEM implementation for the advection-diffusion-reaction (ADR) problem and design an algorithm that computes a solution at each timestep that adheres to a particular structure (e.g., positivity and monotonicity). The

(V. Zala, R. M. Kirby) SCIENTIFIC COMPUTING AND IMAGING INSTITUTE AND SCHOOL OF COMPUTING, UNIVERSITY OF UTAH, SALT LAKE CITY, UT 84112

(A. Narayan) SCIENTIFIC COMPUTING AND IMAGING INSTITUTE AND DEPARTMENT OF MATHEMATICS, UNIVERSITY OF UTAH, SALT LAKE CITY, UT 84112

E-mail addresses: vidhi.zala@utah.edu, kirby@cs.utah.edu, akil@sci.utah.edu.

nonphysical values from such simulations may result in a propagation of nonphysical values, which often cause a blowup, resulting in the failure of the entire simulation. For example, when the concentration values are expected to be bound within $[0, 1]$ but the numerical solution produces a value greater than one, even a small change in the next timestep can cause a rapid compounding of the solution.

The problem of structure preservation has been well studied in different fields for applications to various partial differential equations (PDEs) and using different numerical approaches. The next section provides a brief overview of some existing methods proposed to solve the structure-preservation problem.

1.1. Review. Many sophisticated approaches have been considered in the literature to tackle different aspects of the structure-preservation problem. We make a broad classification into ‘intrusive’ and ‘nonintrusive’ classes of solutions. Many popular approaches generally involve changing either the solution or the PDE and require the imposition of stringent conditions on solver parameters, such as the step-size of the numerical method or the domain shape and granularity. Such methods can be classified as intrusive. On the other hand, some methods constrain the solution obtained from the solver with minimal, often superficial, change to the numerical scheme, i.e., in a nonintrusive way.

Prominent among the intrusive class of approaches is the transformation of structure preservation into a PDE-constrained optimization problem, methods that modify the spatial discretization [22], and limiters derived from Karush-Kahan-Tucker (KKT) optimality conditions [21]. Another example of the intrusive approach [15] proposes an operator-splitting positivity-preservation method based on the energy dissipation law.

The strategies for constraint satisfaction in [3] consider a version of the problem that is a specialization of the one solved by the solution proposed in this manuscript, which falls under the nonintrusive class. In [2, 1] and extensions [3, 17], the authors explore approaches for positivity preservation on the basis functions derived from Bernstein polynomials such that they are non-negative and possess a partition of unity property. Therefore, the interpolated solution respects the original bounds at any point in the domain, nonintrusively. Other nonintrusive methods prescribe solutions such as limiters or truncation and modification of the domain (e.g., curvilinear mesh adaptation) to adhere to the desired solution structures. In addition, many non-intrusive methods perform pre- or postprocessing of intermediate solutions without modifying the underlying problem to incorporate the constraints or making changes to the domain. Some non-intrusive approaches to structure preservation can be found in [14, 26, 19, 16, 20]. In [25], the authors present a positivity-preserving limiter by maintaining cell averages within a certain tolerance. The work in [11] imposes positivity on a discrete grid for kinetic equations via a nonlinear filtering procedure. In this paper we propose a non-intrusive method based on a general filtering approach in [23, 24].

Many of the existing techniques prescribe approaches to preserve the structure that work only for low-order problems. These approaches struggle to maintain high-order accuracy and convergence robustly, including the filter’s ability to preserve structure without restrictions on a per-case basis or problem-specific modifications. Furthermore, the increase in the dimensionality of the problem exacerbates the issue of accuracy and convergence. Therefore, an ideal approach would achieve balance between structure preservation and maintaining accuracy and convergence while providing usability by finitely terminating in an acceptable time.

1.2. Contribution. We pose the problem of preserving the structure in polynomial-based numerical methods as a convex optimization problem. Our previous works discuss a filtering algorithm to solve this problem and demonstrate the solution for function approximations [23] and PDE solutions in 1D [24]. The robust design of the problem enables the preservation of different properties of the solution simultaneously (e.g., positivity, monotonicity, and boundedness). It transforms the problem into a composite constraint satisfaction problem that can be solved by convex minimization. To address this

problem, we use a filter that can be applied as a postprocessing step to deal with structural inconsistencies. The novel idea in the design of the filter is the efficiency achieved by applying the filter only to the violating regions in a subdomain and structure-preservation continuously *throughout* the domain, and not just a subset of points within the domain. This guarantee of structure-preservation depends on the granularity of the global minimization procedure in 2D and 3D, which is expensive. An efficient gradient-based method is required to find the global minimum that represents a structure violation. To streamline this stage and reduce the time and space complexity, we adopt the Barycentric interpolation approach from [5], expanded upon by [12]. Using these techniques, a considerable speed-up in time taken by minimization and overall filter computations is achieved.

This paper focuses on designing a structure-preserving filter using a convex optimization approach for 2D and 3D problems on different element types. We will present the formulation derived from [23] in Section 2 and extend the concepts to more complex problems tackled in this paper. The remainder of the paper is organized as follows: Section 2 discusses the setup for applications of the filter to time-dependent PDEs in 2D and 3D. Section 3 introduces the notations, details the design of different building blocks of the filter, and summarizes the 1D problem formulation from [23, 24]. A procedure for the filtering process developed for multidimensional application is presented in Section 4. Section 5 describes the numerical results to demonstrate the filter's efficacy in preserving the desired structures in different application scenarios. This section is divided into subsections describing the process of choosing the parameters to run the experiments, the 2D and 3D canonical experiments, and an advection problem on different homogeneous meshes using the discontinuous Galerkin (dG) formulation. We conclude with a demonstration of the use of the proposed filter on a real-world application: the model of platelet aggregation and blood coagulation problem.

2. SETUP

In this section, we discuss the setup for filter applications to function approximations and time-dependent PDEs to solve an advection problem using the FEM with method-of-lines discretizations and the dG formulation. The choice of advection problem is just for illustration purposes. The setup remains the same for any time-dependent PDE solution using a polynomial-based method. The filter behaves as a postprocessing step that preserves the desired structure at each timestep¹ and is agnostic to the numerical method used to obtain the solutions. Section 5 shows numerical examples using the same setup on element types such as triangle, quadrilateral, hexahedron, tetrahedron, and prism and meshes comprised of these element types.

2.1. Detecting and resolving structural inconsistency in 2D and 3D. If the polynomial projections lose structural conformity, they are likely to do so at or between the quadrature points. We hope to capture the structural inconsistencies by checking the values on a lattice of quadrature points and the centroids formed by adjacent quadrature points. A quadrilateral is used as a sample element to describe the setup; however, the same applies to any canonical element type in 2D and 3D. Let us call such a lattice of points (X_L). For a quadrilateral, let Q represent the number of quadrature points in one dimension. X_L is defined by a combination of the quadrature grid on the quadrilateral (total Q^2 points) and $(Q - 1)^2$ centroids formed by the midpoints of the quadrature grid as defined in (1).

$$(1) \quad X_L = \{Q^2 \text{ points in the quadrature grid}\} \cup \{(Q - 1)^2 \text{ points in the staggered quadrature grid}\},$$

For a triangle, we can follow the same idea of combining the quadrature points and the centroids to construct the lattice. An example lattice on a mesh with quadrilaterals and triangles is shown in Figure 1.

Consider a function in 2D defined by (2).

¹within a defined tolerance. For example, in all our positivity preservation examples, we consider the tolerance to be numerical zero 10^{-7} .

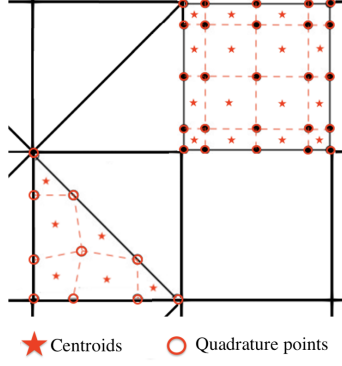


FIGURE 1. Example lattice used for detecting structural nonconformity in a mesh.

$$(2) \quad f(x, y) = \nu \sin(2\pi x) \sin(2\pi y - 0.85\pi),$$

where

$$\nu(x, y) = \begin{cases} 1, & \text{if } x \geq 0 \text{ and } x \leq 0.5 \text{ and } y \geq 0.4 \text{ and } y \leq 0.85 \\ 0, & \text{otherwise.} \end{cases}$$

Projecting (2) using polynomial order $N = 5$ on a quadrilateral in the domain $\Omega = [0, 1] \times [0, 1]$, we get the coefficients $\tilde{\mathbf{v}} = \{\tilde{v}_j\}_{j=1}^P$. The projection is as shown in the left subfigure of Figure 2. In this case, $P = N^d$, where d is dimension.. For this case $d = 1$; therefore, $P = 25$. Since the function is non-negative, the projection should preserve positivity. To this end, we employ the optimization outlined in Section 4 as a post-processing filter. The process behaves as a spectral filter based on the empirical evidence presented in [23, Section 5.2] and further theoretical explanation provided in [23, Proof 5.1]. We will hereafter refer to this optimization as a nonlinear filter (\mathcal{F}). Let \mathbf{v} be the filtered version of $\tilde{\mathbf{v}}$. The filtered projection is shown in the right subfigure of Figure 2.

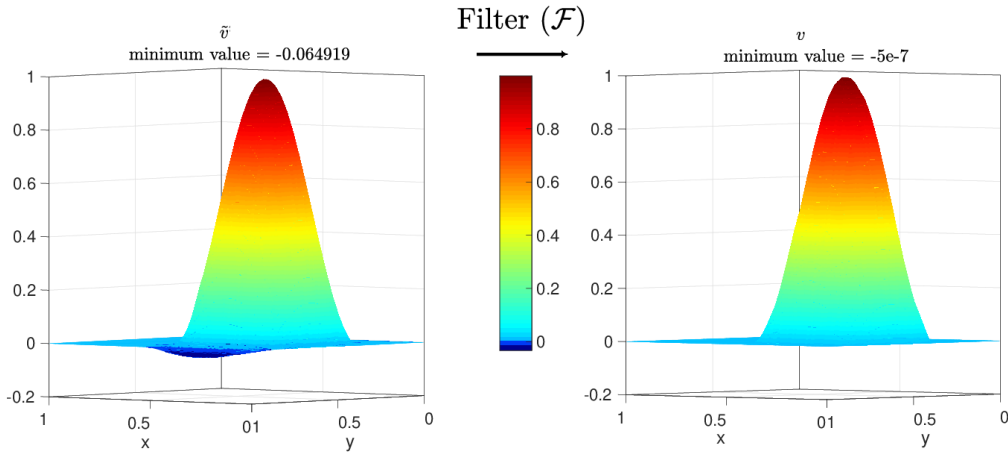


FIGURE 2. *Left:* Projection of (2) using polynomial order = 5. *Right:* Filtered version after application of the structure-preserving filter. The filter converges to a solution that preserves positivity within a tolerance (set to 10^{-6} here).

2.2. Solution to advection problem in dG. The advection equation for a quantity described by a scalar field u is expressed mathematically by a continuity equation (3) in the domain $\Omega \subset \mathbb{R}^d$.

$$(3) \quad u_t + \mathbf{a} \cdot \nabla u = 0,$$

where \mathbf{a} represents the velocity of advection. Assume proper initial and boundary conditions are specified.

Galerkin-type methods assume an ansatz for u as a time-varying element of a fixed P -dimensional linear subspace V that contains the polynomial functions for a fixed degree P , where frequently $V \subset L^2(\Omega)$. Consider a partition of Ω consisting of E subintervals defined by $\tau(\Omega) = \{e_0, e_1, \dots, e_E\}$. For a particular element $e \in \tau(\Omega)$,

$$(4) \quad u_e(x, t) \approx u_P(x, t) := \sum_{i=0}^{P-1} \hat{v}_i(t) \phi_i(x), \quad V = \text{span}\{\phi_1, \dots, \phi_P\},$$

where $x \in \Omega$, and $\{\phi_j\}_{j=0}^{P-1}$ represents the traditional FEM basis, which is not necessarily orthogonal. Let $\{\psi_j\}_{j=0}^{P-1}$ represent a collection of orthonormal basis functions. We can transform the vector of coefficients $\tilde{\mathbf{v}}$ into its orthonormal form $\tilde{\mathbf{v}}$.

In the discontinuous Galerkin formulation, discontinuities are allowed on element boundaries. The semidiscrete form for (3) is derived in the standard Galerkin way by using the ansatz (4) and forcing the residual to be L^2 -orthogonal to V . Usually, integration by parts is performed in the residual orthogonalization step, and often depending on the equation and spatial discretization, numerical flux and/or stabilization terms are included in the weak formulation.

The result is a system of ordinary differential equations prescribing time-evolution of the discrete degrees of freedom represented by vector $\tilde{\mathbf{v}} = \{\tilde{v}_0, \dots, \tilde{v}_{E*P-1}\}$.

$$(5) \quad \mathbf{M} \frac{\partial}{\partial t} \tilde{\mathbf{v}} + \mathbf{A} \tilde{\mathbf{v}} = \mathbf{F}(\tilde{\mathbf{v}})$$

where \mathbf{M} and \mathbf{A} are the $P \times P$ mass and advection matrices, respectively, defined as

$$(M_{i,j}) = \langle \phi_i, \phi_j \rangle, \quad (A_{i,j}) = \left\langle \phi_i, a \frac{\partial}{\partial x} \phi_j(x) \right\rangle,$$

and \mathbf{F} is a general term to compensate for any numerical fluxes or stabilization terms. Finally, (5) is transformed to a fully discrete system that can be solved using an appropriate numerical integration scheme.

Consider the case when the solution on a particular partition e at a timestep n is defined by $\hat{\mathbf{v}}^n$ does not satisfy the desired structural properties. In order to preserve the structure, we employ the optimization Section 4 as a postprocessing filter. Let \mathbf{v} represent the filtered solution which satisfies the structure. The process can be illustrated as Equation (6). The proposed filter works by simply augmenting this scheme non-intrusively.

$$(6) \quad \mathbf{v}^n \xrightarrow{\text{Timestepper for (3)}} \hat{\mathbf{v}} \xrightarrow{(\mathcal{F})} \mathbf{v},$$

For efficiency, the filter is applied only to the subintervals $\tau(\Omega)$ in which structural violations are found at lattice points. The problem is now converted from optimization in an N -dimensional space into a much more efficient and parallelizable set of E independent optimizations each in $P := N/E \ll N$ dimensions. In dG, we can filter the selected subintervals simultaneously, which further adds to the procedure's efficiency.

For a mesh with the number of elements $E > 1$, the process of enforcing positivity per element leads to changes in solution properties; in particular, elementwise boundary values and the mean of the

discrete solution are not preserved. Assume that numerical fluxes are computed as explicit functions of element boundary values. A shift in element boundary values by the filter causes changes in the corresponding fluxes, thus adding errors to the simulation. We can resolve this issue by imposing additional equality constraints (i.e., function values at element boundaries) in the filter. A similar approach for mass (integral) conservation can be obtained using a procedure that is essentially identical to flux conservation. The mathematical details for the flux and integral conservation by posing as a constraint satisfaction problem can be found in [24, Section 3.2 and 3.3]. The robust design of the filter allows for the incorporation of an arbitrary number of structural constraints into one optimization problem. However, increasing the number of constraints reduces the degrees of freedom for the filter accordingly. In particular, if the filter has P degrees of freedom and we have K linear constraints, then the dimension of the optimization problem can be reduced to an $(P - K)$ -dimensional problem using the procedure described in [24, Section 3.2].

3. FORMULATION OF STRUCTURE-PRESERVATION PROBLEM AND SOLUTION DESIGN

This section establishes the notations and summarizes formulation, filter design, and implementation ideas from [23, 24].

3.1. The problem. Given a function $u(x) \in V$, where V is a finite-dimensional Hilbert space of real-valued functions on $\Omega \in \mathbb{R}^d$, a comprehensive constraint-satisfaction problem can be constructed as follows:

Consider a family of linear constraints, each of the form defined by (7).

$$(7) \quad L_x(u) \leq \ell(x), \quad x \in \Omega,$$

where L is a linear operator bounded on V , and ℓ is a function on the domain Ω , which results in a collection of feasible sets, one for each constraint. Let us consider a full feasible set to be an intersection K of the collection of feasible sets. Therefore, $K \subset V$. The feasible set adhering to such a constraint family is convex. Some examples of constraints are positivity, monotonicity, and flux preservation.

The problem is essentially that of developing a map \mathcal{F} from a function $u \in V$, to a unique function $\mathcal{F}(u) \in K$. Initially, u may not satisfy the structural constraints but its mapped version $\mathcal{F}(u)$ does. We get the equivalent well-posed problem by treating (7) as a convex feasibility problem (8).

$$(8) \quad \mathcal{F}(u) := \operatorname{argmin}_{k \in K} \|u - k\|_V,$$

Under certain conditions, such as if $0 \in K$, \mathcal{F} is contractive; therefore, the operation can be called a filter. For a brief discussion of the norm-contractive nature of (8) see [23, Proposition 5.1].

3.2. Toward construction of a convex optimization-based solution to (8). We first transform the continuous problem (8) to a discrete version to construct a feasible solution. Let $\{\psi_j\}_{j=0}^{P-1}$ be a collection of orthonormal basis functions in V , different from the standard FEM basis (ϕ) , which are not orthonormal.

Therefore, we have

$$V = \operatorname{span} \{\psi_0, \dots, \psi_{P-1}\}, \quad \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad i, j = 0, \dots, P-1,$$

where $\langle \cdot, \cdot \rangle$ is the inner product on V , and δ_{ij} , the Kronecker delta function.

Given the function $u \in V$, we can represent it using piecewise polynomial expansion by a combination of ψ and coefficients $\{\tilde{v}_j\}_{j=0}^{P-1}$, such that

$$u = \sum_{j=0}^{P-1} \tilde{v}_j \psi_j.$$

For 1D, using polynomial order N , we have $P = N$. Let C be the affine conic region in \mathbb{R}^P corresponding to $K \subset V$, $\tilde{\mathbf{v}}$ be a vector of expansion coefficients of u that does not satisfy that desired constraints, and \mathbf{v} be a vector of filtered coefficients of u that satisfies the desired constraints. For a constraint c that is not satisfied by expansion $\tilde{\mathbf{v}}$, let \mathbf{v} be the *filtered* expansion coefficients that satisfy the constraints, $\|\cdot\|_2$ is the Euclidean 2-norm on vectors, and C is the affine conic region in \mathbb{R}^P corresponding to $C \subset V$. The procedure can be illustrated as in (9).

$$(9) \quad \underset{\mathbf{v} \in C}{\operatorname{argmin}} \|\tilde{\mathbf{v}} - \mathbf{v}\|_2,$$

If $K = 1$, we have a single constraint of the form (7). In this case, the feasible region C is defined by

$$C = \bigcap_{x \in \Omega} H_x,$$

where H_x is an $(P - 1)$ -dimensional planar surface in \mathbb{R}^P defined by the single linear constraint (7). The discussion in [24, Theorem 3.1] establishes the convexity of C .

The algorithms in [23] perform filtering based on the global minimum search on the signed distance function s ,

$$(10) \quad s(x) := \operatorname{sdist}(\tilde{\mathbf{v}}, C_x) = \begin{cases} -\operatorname{dist}(\tilde{\mathbf{v}}, H_x), & x \notin C_x, \\ +\operatorname{dist}(\tilde{\mathbf{v}}, H_x), & x \in C_x. \end{cases}$$

For each family in the feasible sets $\{C_k\}_{k=1}^K$, we have K signed distance functions $\{s_k\}_{k=1}^K$, each of the form (12). The full feasible set C is the intersection of the C_k . The filter aims to project of $\tilde{\mathbf{v}}$ onto the feasible set C_k , iteratively. At each iteration, the constraint index k and the spatial point x are found to minimize the signed distance function given by (11).

$$(11) \quad (x^*, k^*) := \underset{x \in \Omega, k \in \{1, \dots, K\}}{\operatorname{argmin}} s_k(x)$$

This greedy procedure is an infinite-constraint generalization of Motzkin's relaxation method [18]. The major computational work in the filtering procedure is to minimize the objective defined by (12) at each iteration.

$$(12) \quad s(x) = \lambda(x) (L_x(u) - \ell(x)), \quad \lambda^2(x) := \frac{1}{\sum_{j=0}^{P-1} (\mathcal{L}_x(\psi_j))^2},$$

where ℓ is a function on Ω .

Next, update $\tilde{\mathbf{v}}$ by projecting it onto the hyperplane corresponding to (x^*, k^*) :

$$(13) \quad \tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \mathbf{h}(x^*, k^*) \min\{0, s_{k^*}(x^*)\}$$

where $\mathbf{h}(x, k)$ is the normal vector corresponding to hyperplane H_x of constraint family k that points toward C_k . This vector is readily computable from the orthonormal basis, see [23] for details. This procedure is repeated until $s_{k^*}(x^*)$ vanishes to within a numerical tolerance.

Geometrically, H_x divides the hyperspace into two parts, as shown in Figure 3. Let the current state vector of the projection defined by $\tilde{\mathbf{v}}$ be a point shown by the blue dot in Figure 3. The filter works by applying correction (13) to $\tilde{\mathbf{v}}$ by computationally inspecting the signed distance function (10). Here, "inspection" means, for the positivity-preservation example, the ability to compute the global minimum of $s(x)$ to determine regions where s is negative. Based on this inspection, the algorithms project the state vector of the current iterate $\tilde{\mathbf{v}}$ onto H_y for some $y \in \Omega$. This projection can cause a particular H_{y1} with a positive $\operatorname{dist}(\tilde{\mathbf{v}}, H_{y1})$, to go negative. Therefore, we need to repeat the projection step until

the solution $\tilde{\mathbf{v}}$ lies completely in (or on the boundary of) C . [23, Algorithm 1] summarizes all the steps of the filter.

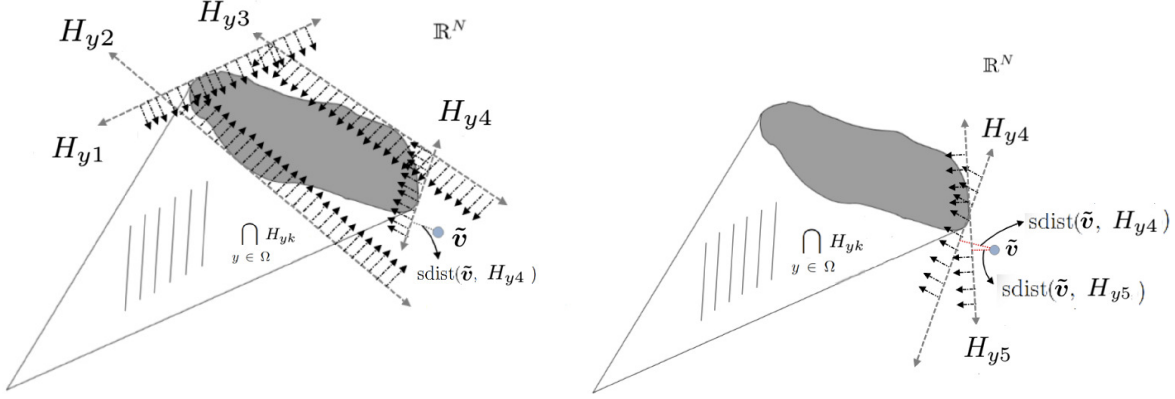


FIGURE 3. Division of the space into half-spaces by the hyperplanes representing constraint boundaries defined by the set of points (say y_1, y_2, \dots, y_n) for the initial projection with coefficients $\tilde{\mathbf{v}}$. The algorithm greedily calculates the correction $s(x)$. Left: A geometrical visual of the distance calculation from $\tilde{\mathbf{v}}$ to the hyperplanes defining the boundaries of the constraints. Right: Projection of $\tilde{\mathbf{v}}$ on to H_{y4} . H_{y4} is selected over H_{y5} since it defines the violating hyperplane that is farthest away.

Finding the violating hyperplane that is farthest away corresponds to finding the global minimum of $s(x)$ on Ω . Unlike the 1D procedure described in [23], the multidimensional global minimization requires a more expensive approach such as the gradient descent (GD). In this paper, we implement 2D and 3D global minimization using line-search-based GD with backtracking. We further provide a strategy to investigate the times taken for convergence and the accuracy for this GD approach by varying the values of the parameters. Then, we choose the best values for the parameters for the accuracy of the method. Section 4.1 provides details of the GD variant chosen for filter implementation and corresponding parameter selection to efficiently solve the minimization problem.

4. ALGORITHM

The structure of the solution is preserved by applying the filter as a post-processing step that corrects $\hat{\mathbf{v}}$ obtained from timestepping algorithm using iterative optimization. Each iteration of the filter involves the following crucial computational processes.

- (1) A GD-based minimization to solve (11), which is a part of global search for structural violations.
- (2) A calculation of the correction to $\hat{\mathbf{v}}$ using (13)

We briefly discuss the algorithmic details and the choices made to streamline the processes and its applications to the PDE timestepping.

4.1. Finding the global minimum. An important and the most expensive part of the optimization process is the global minimization (11). For efficiency and usability of the filter, we need to fine-tune the most expensive step, i.e., minimizing the scaled distance function (12), which quantifies the difference between the current state of the coefficients and the coefficients that satisfy the constraints at a critical point of current iteration. The success of the proposed filter depends on accurately finding the points in the domain that violate the structure. Since the problem formulation uses dG, it can be solved element-by-element on a domain with E elements. Applying the filter at each timestep is essentially solving E optimization problems at every timestep. Each optimization problem is over a single element. We can reduce the number of optimization problems by testing the solution for structural inconsistencies at a

lattice of points defined by (1). For example, in the case of positivity preservation, if the minimum value of an element is nonpositive, the filter is applied on that element.

In the 1D case, the filter finds the minimum value by evaluating the solution at the critical points on the element. If the subspace V restricted to an element is a space of polynomials, then the critical points of (12) are defined by the roots of the derivative of (12). For the root-finding problem, we employ the confederate matrix-generation approach. Relative to 1D, minimization in higher dimensions is more complex and expensive. Many variants of the gradient descent-based (GD-based) methods have been proposed to perform this operation efficiently and accurately. These methods operate only up to a certain tolerance and, like all the GD-based methods, have the potential for stagnation.

In the 2D and 3D cases, the problem of finding the points on the domain that violate the structural constraints is nonconvex. We cannot ensure that all such points will be flagged for correction by any particular minimization algorithm. From our analysis, one suitable GD approach for this job is to use adaptive descent size and the ability to retrace the steps. For this reason, we choose the method of steepest descent with a backtracking line search strategy. The backtracking line search starts with a relatively large step size and repeatedly shrinks it by a factor γ until the Armijo–Goldstein condition (14) is fulfilled. This widely used algorithm makes an informed choice about step size and direction at each iteration to efficiently arrive at the optimum value. It avoids stagnation by tracing back its steps if the difference between descent values falls below a fixed tolerance. An essential part of this algorithm is choosing the parameters denoted by $c \in (0, 1)$ and $\gamma \in (0, 1)$ that determine the step size and backtracking performed by the algorithm. For an objective function f , with a starting position x , given parameters c and γ the Armijo–Goldstein condition is defined as (14).

$$(14) \quad f(\mathbf{x} + \gamma_j \mathbf{p}) \leq f(\mathbf{x}) + \gamma_j c \nabla f(x)^T \mathbf{p} \quad \forall j \text{ until convergence,}$$

where $\nabla f(x)^T$ is the slope in the direction of descent p and γ is calculated as

$$\gamma_j = \begin{cases} \gamma, & \text{if } j = 0 \\ c\gamma_{j-1} & \text{otherwise.} \end{cases}$$

c and γ play a vital role in testing the condition [4], which determines whether a step-wise movement from a current position to a modified position achieves an adequately corresponding decrease in the objective function. Although many recommendations exist [4, 9, 10, 6, 7] for an optimal way to choose c and γ , these parameters must be tuned for maximum efficiency.

4.2. Applying the structure-preserving filter to $\tilde{\mathbf{v}}$ obtained from the timestepper. From the discussion in Section 2, the optimization that enforces a constrained solution is applied as a postprocessing part inside the timestepper, independent of the choice of the timestepping routine. As predicted by the ‘curse of dimensionality’, structure preservation and optimization for higher dimensional problems are more complex than their 1D counterparts. Since the central idea of the filter is agnostic to the dimensions of the problem, the procedure Algorithm 1 remains the same as described in [23]. Similarly, the applications to PDE solutions follow the procedures established in [24] for 1D problems.

Conversion from an input coefficient vector $\hat{\mathbf{v}}$ to corresponding coefficients $\tilde{\mathbf{v}}$ in an orthonormal basis is the first stage of the filtering procedure. Algorithm 1 presents a summary of steps taken by the PDE solver for the filter application.

A significant chunk of work in the iterative correction stage is done by evaluating the solution at the lattice points using basis interpolation, an expensive operation that involves processing and storage of large interpolation matrices. To make the processing computationally efficient, we use the Barycentric interpolation method from [5], expanded upon by [12]. This method reduces the number of calculations and storage required to evaluate the basis functions at arbitrary points in the domain,

thereby reducing the cost of the iterative correction stage. Although the implementation details of Barycentric interpolation method are beyond the scope of this paper, for the minimization part of the numerical experiments in Section 5, we observe a significant speed-up by using the Barycentric approach.

Algorithm 1 Constrained PDE timestepping

```

1: Input: Terminal time  $T$ , timestep size  $\Delta t$ , PDE solver spatial basis  $\phi$ 
2: Orthonormalize the coefficients  $\hat{v}$  from non-orthogonal basis  $\phi$  to  $\tilde{v}$  in orthogonal basis  $\psi \in \mathbb{R}^P$ 
3: Define  $nsteps = \frac{T}{\Delta t}$ 
4: for  $i = 0, \dots, nsteps$  do
5:   Solve PDE to obtain the coefficients  $\tilde{v}_e^i \in \mathbb{R}^P$  for all elements in  $e$  in  $\Omega$ 
6:   Input: constraints  $(L_k, \ell_k)_{k=1}^K$ 
7:   for Each element  $e \in \Omega$  do
8:     while True do
9:       if  $\exists sdist(X_L) < 0 \forall X_L$  in (1) then
10:        Compute  $(x^*, k^*)$  via GD_linesearch (see Section 4.1).
11:       else
12:         continue
13:       end if
14:       Update  $\tilde{v}_e^{i+1}(t)$  via (13).
15:     end while
16:   Append to global  $v^{i+1} = [v^{i+1}, \tilde{v}_e^{i+1}]$ 
17: end for
18: end for

```

5. MAIN RESULTS

This section presents four-part numerical results of the filter's application to different types of problems in 2D and 3D. Section 5.1 details the process to choose the parameter values for the GD line search algorithm described in Section 4.1. Section 5.2 shows the results of applying the filter to a single-element function projection in multiple dimensions. Section 5.3 presents the results and analysis of application of the filter using the dG formulation of the advection problems defined on composite and homogeneous domains, respectively. Finally, in Section 5.4 we present the results of the FEM solution to the mathematical-biology problem of platelet aggregation and blood coagulation (PAC) with and without the application of the filter. The PAC model is posed as an advection-diffusion-reaction (ADR) system of PDEs.

5.1. Parameter selection for backtracking line search used in GD-based minimization. To fix the values for c and γ (14) for the numerical results in Section 5, we first choose a set of functions. The choice of the functions depends on the factors such as the constraint types and the domain geometry. For example, for an application where positivity is the primary structural concern, a list of functions with values close to zero is a good choice. For applications involving discontinuities in the function or the domain, discontinuous functions would be a good choice. The feasible space for c and γ is $(0, 1)$. A noncomprehensive sampling on $(0, 1)$ is performed for both parameters, and the GD is called with the parameters set to permutations of the sampled values. GD_linesearch denotes the call to the GD with line search. As its output, the GD_linesearch routine provides the number of iterations (denoted by *niter*) taken by the algorithm. The error refers to the difference between the minimum value found and the known minimum value.

As the large number of GD iterations per filter iteration makes the filtering process more expensive, it is desirable to choose the parameter values for which GD converges in the least number of iterations. Another important quantity to consider here is the error. It is desirable to have the GD returning with the minimum value as close to the known (or golden) minimum value as possible. With these

quantities (*niter* and *err*) as the metrics, we prescribe a procedure to select the ideal value of c and γ in Algorithm 2. The selection criteria narrow the samples to a range of c and γ with the minimal number of iterations and then pick from that subset the ones with the smallest error. In the case of a tie, all the c and γ values are included in the return variables. The procedure can be summarized as taking a list of functions that fairly represent the complexity of space and returning two ranges of ideal values: one for each of the parameters c and γ , respectively. Since the quadrilateral and hexahedron represent richer space than their other canonical 2D and 3D counterparts, Algorithm 2 is applied and analyzed only on a quadrilateral for 2D and a hexahedron for 3D.

Algorithm 2 Experiment to determine ideal values of c and γ given a set of functions f_i , for $i = 0, 1, \dots, n$

```

1:  $G \leftarrow P$  samples in  $(0, 1)$ 
2: for each  $f_i \forall i = 0, 1, \dots, n$  do
3:    $H \leftarrow \{\}$ 
4:   for each  $\gamma_j$  in  $G$  do
5:     for each  $c_k$  in  $G$  do
6:        $(niter, err) \leftarrow \text{GD.linesearch}(f_i, c_k, \gamma_j)$ 
7:        $H \leftarrow \{H; \{c_k, \gamma_j, niter, err\}\}$ 
8:     end for
9:   end for
10: end for
11:  $H_1 \leftarrow$  select ranges of  $c$  and  $\gamma$  with least niter from  $H$ 
12:  $H_2 \leftarrow$  select ranges of  $c$  and  $\gamma$  with least err from  $H_1$ 
13: return Ranges for  $c$  and  $\gamma$  from  $H_2$ 

```

Future developments in the approaches to find optima in multiple dimensions may improve the efficacy and efficiency of the filter. To this end and to provide robustness, we employ the object-oriented principle by modularizing the minimization part of the filter.

We consider different example functions (15) on 2D domain $[-1, 1] \times [-1, 1]$ and (16) on 3D domain $[-1, 1] \times [-1, 1] \times [-1, 1]$, which have varying complexities and projection orders to set the GD parameters in a comprehensive and fair manner.

$$\begin{aligned}
 f_0(x, y) &= (x + 0.6)^2 + (y - 0.2)^2. \\
 f_1(x, y) &= -\sin((x - 0.1) + 0.5\pi) \cos(y - 0.2). \\
 f_2(x, y) &= \begin{cases} 1, & \text{if } x \leq 0 \text{ and } y \leq 0, \\ 0, & \text{otherwise.} \end{cases}
 \end{aligned}
 \tag{15}$$

$$\begin{aligned}
 f_3(x, y, z) &= (x + 0.6)^2 + (y - 0.2)^2 + (z + 0.1)^2. \\
 f_4(x, y, z) &= -\sin((x - 0.1) + 0.5\pi) \cos(y - 0.2) \cos(z - 0.2). \\
 f_5(x, y, z) &= \begin{cases} 1, & \text{if } x \leq 0 \text{ and } y \leq 0 \text{ and } z \leq 0, \\ 0, & \text{otherwise.} \end{cases}
 \end{aligned}
 \tag{16}$$

Using the orthogonal basis for (13), $\tilde{\mathbf{v}}$ is obtained, which is then used by Algorithm 2 to find the values of parameters c and γ . For functions that have an exact analytical minimum, calculating the error between the exact minimum and the optimum returned by the GD is straightforward. For the discontinuous functions such as f_2 and f_5 , an approximate *golden* minimum is calculated by projecting the function using polynomial order $N = 8$ on a dense grid of points in Ω and finding the least value of the projected polynomial. In all our test cases, GD converges to the exact (or golden) minimum within

acceptable tolerance (numerical 0). Therefore, the selection of the parameters is made on the basis of the fewest iterations taken by GD (*niter*) to converge.

Consider k discrete equispaced samples of $\in (0, 1)$. For $k = 9$, by applying Algorithm 2, we get the results shown in Table 1.

M	$f_0(x, y)$			$f_1(x, y)$			$f_2(x, y)$		
	Range c	Range γ	niter	Range c	Range γ	niter	Range c	Range γ	niter
3	[0.4,1)	(0,1)	6	[0.7,1)	(0,1)	3	(0,1)	(0,1)	3
4	[0.5,1)	(0,1)	7	(0,1)	[0.3,1)	4	[0.2,1)	(0,1)	7
5	[0.5,1)	[0.3,1)	7	[0.6,1)	[0.2,1)	4	(0,1)	(0,1)	7
6	(0,1)	[0.5,1)	7	[0.3,1)	[0.5,0.8]	4	[0.4,0.9]	[0.7,1)	6
7	[0.2,1)	(0,1)	7	[0.3,1)	(0,1)	5	[0.5,1)	(0,1)	10
8	[0.7,1)	[0.5,1)	6	(0,1)	[0.7,1)	4	[0.3,1)	(0,1)	10
9	[0.7,1)	[0.7,1)	6	[0.5,1)	(0,0.2]	6	[0.2,0.6]	(0,1)	7
10	[0.5,0.7]	[0.5,1)	7	[0.5,1)	[0.3,1)	6	[0.5,1)	[0.8,1)	9

Table 1: Results for the 2D experiment on a quadrilateral domain to find the optimal ranges of the gradient descent line search parameters: c and γ . M denotes the polynomial order for projected functions (15). The number of iterations taken by GD is denoted by *niter*. The number of quadrature points for projection is constant ($Q = 11$)

Based on the results of these tests, we infer that for the numerical experiments, the appropriate choices for c and γ in 2D are 0.7 and 0.7, respectively. Following a similar procedure, Algorithm 2 on a hexahedron for functions defined in (16), the choices of c and γ for 3D are 0.2 and 0.7, respectively. These values are used for all the numerical experiments in the numerical results.

The C++ implementation of the filter in Nektar++ [8] supports change in c and γ as parameters in the configuration file.

5.2. Projection examples on 2D and 3D elements and application of the structure-preserving filter. Consider a 2D function (17) and a 3D function (18) that are both discontinuous clamped versions of a smooth sinusoidal function. The initial projections of $f(x, y)$ on a quadrilateral and $f(x, y, z)$ on a hexahedron are shown in Figure 4. The discontinuity produces oscillations similar to Gibb's phenomenon upon projection, which is interesting for the application and analysis of the filter.

$$(17) \quad f(x, y) = \nu(x, y) \sin\left(\pi(0.2 - x)\right) \sin\left(\pi(y + 0.2)\right),$$

where

$$\nu(x, y) = \begin{cases} 1, & \text{if } x \in [-0.8, 0.2] \text{ and } y \in [-0.2, 0.8], \\ 0, & \text{otherwise.} \end{cases}$$

$$(18) \quad f(x, y, z) = \nu(x, y, z) \sin\left(\pi(0.2 - x)\right) \sin\left(\pi(y + 0.2)\right) \sin\left(\pi(z + 0.2)\right),$$

where

$$\nu(x, y, z) = \begin{cases} 1, & \text{if } x \in [-0.8, 0.2] \text{ and } y \in [-0.2, 0.8] \text{ and } z \in [-0.2, 0.8] \\ 0, & \text{otherwise.} \end{cases}$$

It is evident from the left subfigures in Figure 4 that the projection of discontinuous non-negative functions leads to the negative values as shown in the highlighted regions. As seen in the right subfigures of Figure 4, the application of the structure-preserving filter restores the non-negative structures of (17) and (18), respectively. To analyze the p-convergence, the experiment is repeated for different polynomial

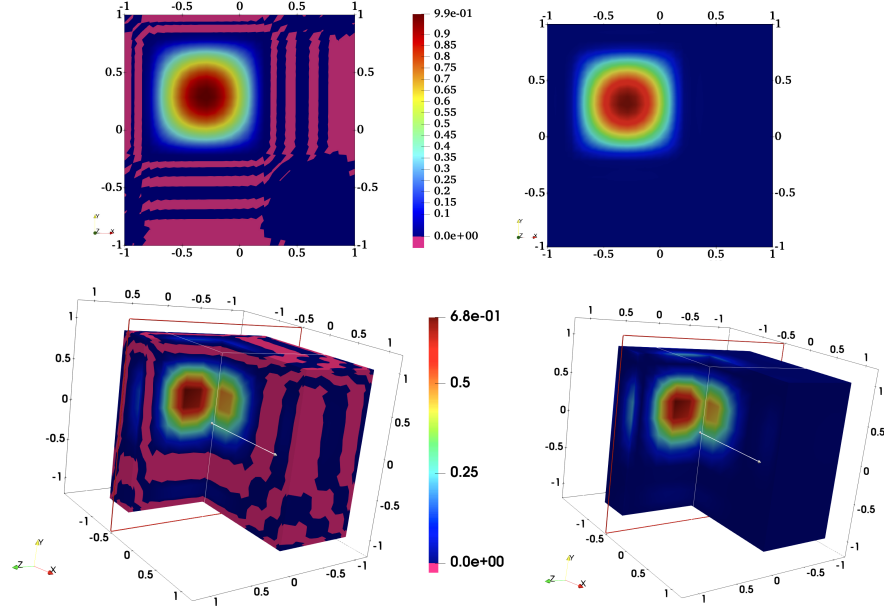


FIGURE 4. Galerkin projection of $f(x,y)$ (17) and $f(x,y,z)$ (18) on a quadrilateral and hexahedron element, respectively. The 2D projection uses polynomial order = 7, and the 3D projection uses polynomial order = 5. *Left:* Unfiltered version showing areas where the desired structure (positivity) is lost. *Right:* After the filter is applied, the solution is non-negative up to tolerance.

orders, as shown in Figure 5.

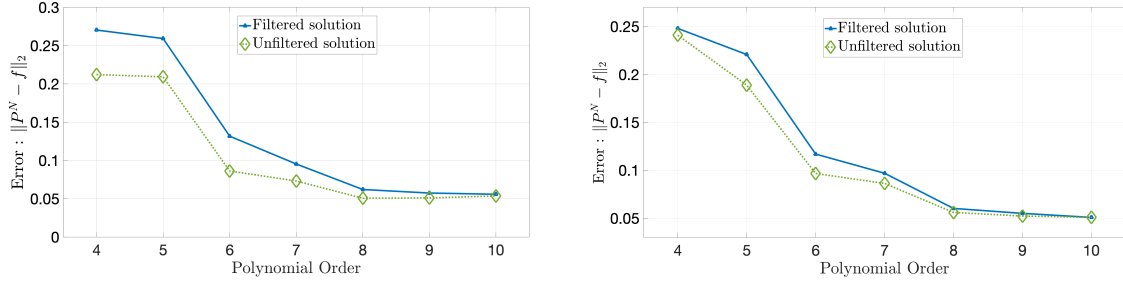


FIGURE 5. A p-convergence study of the filtered and unfiltered Galerkin projections from Figure 4. The right y-axis represents the time spent inside the filtering routine. *Left:* $f(x,y)$ defined in (17) and *Right:* $f(x,y,z)$ defined in (18)

As shown in Figure 5, the time taken by the filter scales with the size of the solution space. Another observation suggests that the 2D filtering experiments take a fraction of time compared to their 3D counterparts which results from the complexity of finding the global minimum in the domain, which is much higher in 3D than in 2D.

After analyzing the behavior and time taken by the filter in a worst-case scenario: projection of discontinuous (nonsmooth) functions, for subsequent experiments, we consider *smooth* continuous functions.

5.3. Structure preservation in 2D and 3D advection problems. Consider a dG solution to the advection problem (19) on a 2D domain as shown in the first row of Figure 6.

$$(19) \quad u_t + \mathbf{a} \cdot \nabla u = 0,$$

Consider a function f defined on $[-1, 1] \times [-1, 1]$ as (20).

$$(20) \quad f(x, y) = 1 - \cos\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi y}{2}\right)$$

For a given initial condition to be $f(x, y)$, $\mathbf{a} = [1, 1]$, and the periodic boundary conditions, we formulate the problem in the discontinuous Galerkin framework. The first step is projecting f on a set of E elements $\{e_0, e_1 \dots e_{E-1}\} \in \Omega$ using the typical nonorthogonal (hats and bubbles) basis ϕ . Locally, for an element e , we have

$$f_e(x, y) = \sum_{i=0}^{P-1} \hat{u}_i \phi_i(x, y),$$

where Ω is the mesh shown in Figure 6 consisting of a mix of quadrilaterals and triangles. The filter steps do not change for different element types as emphasized by the choice of composite mesh.

The solution for the advection problem using the timestep $\Delta t = 10^{-3}$, polynomial order 4, RK-4 integration scheme, and upwind flux calculations is shown in Figure 6. Row 2 of the figure shows the simulation state at a particular timestep and highlights the negative values in the domain. In row 3 after the application of the filter, the non-negative structure is restored. The filter is applied at each timestep to preserve the structure (positivity) of the solution on a lattice of points of interest. The lattice is a set of points defined in (1). If a structure violation is found at any point in the lattice, the parent element is flagged for filtering. Since the boundaries are periodic, the final state of the simulation looks similar to the initial state.

For a similar analysis of the 3D advection problem, consider the initial state as a smooth continuous function (21) on a cube mesh of 64 hexahedron elements defined in the domain $[-1, -1, -1] \times [1, 1, 1]$. The advection velocity is defined by $\mathbf{a} = [1, 1, 1]$, and all the boundary conditions are periodic. The integration method used is RK-4 with a timestep of 10^{-3} . The timestepping is performed for a total of 2000 timesteps and the flux calculation is upwind. Following a similar selection procedure as in 2D, we discover the elements that violate the structure at each timestep and apply the structure-preserving filter to those elements.

$$(21) \quad f(x, y, z) = 1 - \cos\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi y}{2}\right) \cos\left(\frac{\pi z}{2}\right)$$

Figure 7 shows an instance during the advection process where a loss of structure is encountered, i.e., negative intermediate values are found, as shown by the highlighted region in the left and middle subfigures of Figure 7. The process of discovering the structurally nonconformal elements and application of the filter to those elements adds to the total time cost of the advection solution, which is an important aspect to consider when making a decision about the criteria for applying the filter.

The percentage increase in total time taken by the experiment by the application of structure-preserving filter is shown in Figure 8. Note that for 2D, at low orders ($N=2,3,4$), the cost of filtering is between 50-70% of the total time. This is due, in part, to the efficiency at these orders of the unfiltered solver such as caching effects, as well as up-front costs associated with the optimization. We find that for 2D the ratio of filtered to unfiltered is lowest at $N=5$, and then proceeds to climb linearly as shown in the left subfigure of Figure 8. The time taken by the filter in 3D is notably higher than the time taken by the filter in 2D because of the higher complexity of finding the global minimum in 3D. The cost of overall filtering per timestep depends on the number of elements that the filter operates upon. Therefore, the procedure of selective application of the filter becomes increasingly important.

The procedure to filter the elements that exhibit loss of structure remains agnostic to the type of element. To demonstrate this, consider different tessellations of the domain $[-1, -1, -1] \times [1, 1, 1]$ and solve an advection problem with nontrivial initial state (22). The setup remains the same as the previous 3D example, i.e., $\mathbf{a} = [1, 1, 1]$ in all directions, periodic boundaries, RK-4 integration scheme using timestep 10^{-3} , and for a total of 2000 timesteps.

$$(22) \quad f(x, y, z, t = 0) = 0.2 \left([1 - \sqrt{(x^2 + y^2)}]^2 + z^2 \right)$$

The initial state looks like a torus inside a cube. The advection experiment is repeated on different homogeneous meshes with element types hexahedra, tetrahedra, and prisms individually. Figure 9 shows the results before and after the filter application at a particular timestep for all these meshes.

The analysis of the filter's effects on the accuracy and convergence of the advection process on the meshes with hexahedron and prism elements reveals that the difference between the L_2 errors is too small to warrant a convergence comparison. However, for the mesh with tetrahedral elements, the L_2 errors perceptibly vary at each timestep, as shown by a longer run of the same experiment (total 4000 timesteps) in Figure 10.

5.4. dG FEM implementation of the PAC model with filtering. A prominent scenario in which the problem of structure-preservation is encountered is in the mathematical-biology domain. A detailed mathematical model for platelet aggregation and blood coagulation (PAC) by [13] is considered for the purpose of this experiment. The model describes the process of evolution, interaction, and decay of the

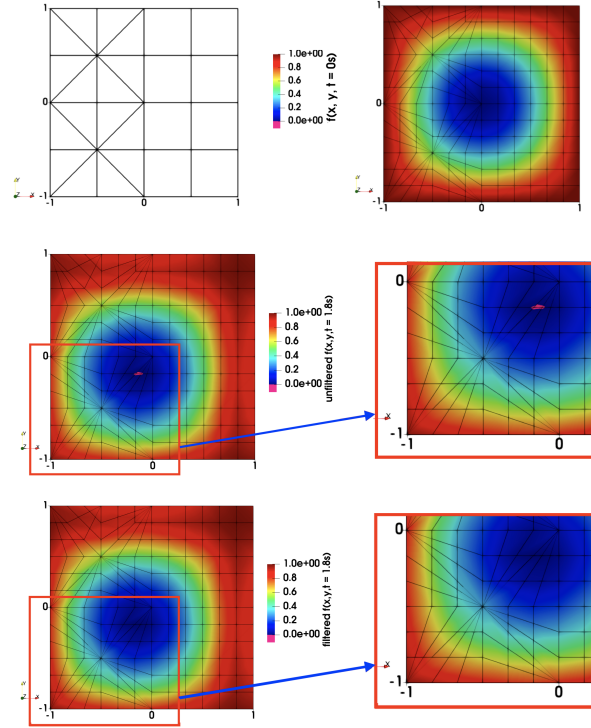


FIGURE 6. *Row 1, Left:* A 2D composite mesh used for solving (19). The mesh contains 16 triangles and 8 quadrilaterals. *Row 1, Right:* Initial state of the system, which is a projection of (20) on the mesh. *Row 2:* Unfiltered solution at timestep 1800. *Row 3:* Filtered solution at timestep 1800.

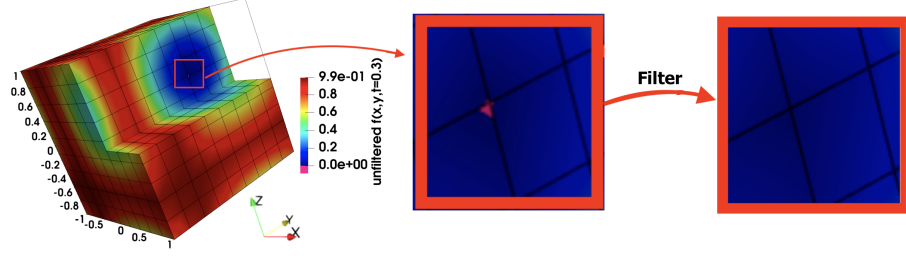


FIGURE 7. The state of the simulation at $t = 0.3$ seconds (timestep = 3000) using initial condition Equation (21). *Left and middle:* The highlighted region and its zoomed counterpart shows the points in the elements that lose positivity structure. *Right:* The filtered values at the same elements that preserve the positivity structure.

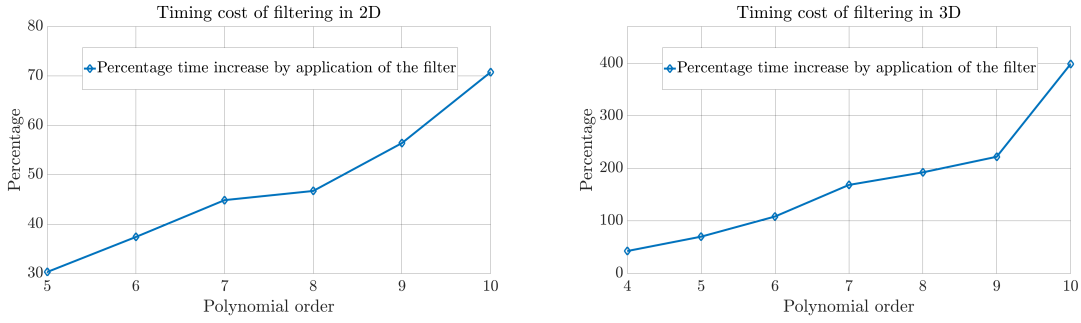


FIGURE 8. Percentage increase in time to complete the experiment by application of the filter. *Left:* 2D *Right:* 3D

chemical species involved in the process of thrombosis.

Although the details of the individual species in the PAC process are beyond the scope of this paper, [13] has a detailed explanation of the nature and the evolution of the chemical species. The model can be summarized as a comprehensive collection of ODEs and PDEs that track all the chemical species in various phases during the process. The results in [13] use a finite-difference method with a specified limiter to truncate nonpositive concentration values to zero. In order to implement this model using the FEM without loss of structure, an alternative approach to truncation, such as the one presented in this paper, is needed.

In this section, we implement the model using the dG method. Instead of a truncation limiter, we use the structure-preserving filter. For simplicity, the focus is on the system of equations that track the chemical species that advect, diffuse, and react. One such species is fluid-phase thrombin (FP e_2), an essential component of the thrombosis process. For this section, the focus is to track the evolution, advection, diffusion, and decay of thrombin in the fluid-phase. For the detailed results of evolution of all the chemical species, including thrombin, under various circumstances in the original model, refer to [13].

We set up a version of the PAC model that solves the species evolution problem by a combination of advection, diffusion, and reaction (ADR) PDEs. Unlike the original model, the velocity \mathbf{u} of the fluid medium is not reported by a modified Navier-Stokes solver. Instead, the velocity is constant $\mathbf{u} = [u_x, u_y] = [5, 0]$. To solve this system of PDEs on a sample blood vessel represented by a rectangular domain $\Omega = [0, 300] \times [0, 20]$, we employ the dG FEM. The domain is tessellated using 2048 quadrilaterals. The bottom wall has an injury site spanning from $x = 20$ to $x = 60$. We use polynomial

order = 4 and timestep of $\Delta t = 10^{-2}$ to solve the problem.

The total number of species tracked in our implementation of the model is 56. Since the focus is on the behavior of thrombin, we present the PDE describing the evolution of fluid-phase thrombin (FP e_2) (23). For the details on the other chemical species involved in (23), refer to [13]. The boundary conditions vary depending on the chemical being tracked. For FP e_2 , we have a Dirichlet zero boundary on the left, and a no-flux boundary on the right and top. The bottom boundary has two kinds of conditions: a robin boundary condition on the injury site and Dirichlet zero everywhere else.

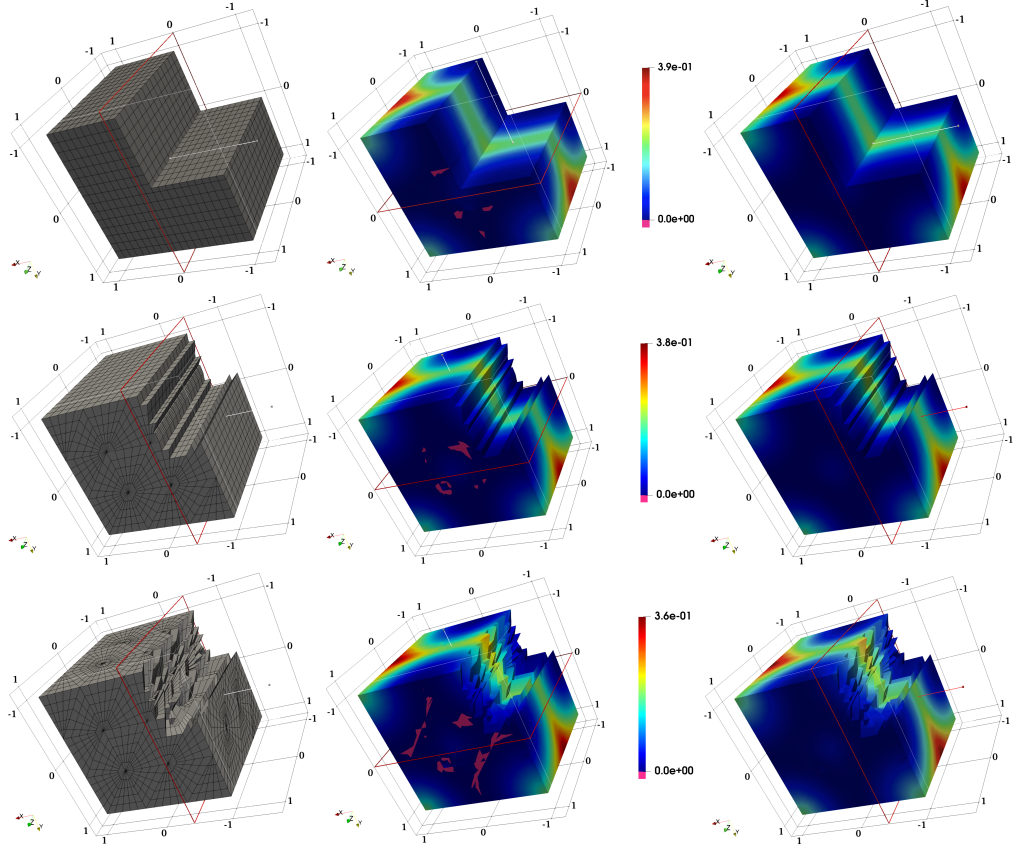


FIGURE 9. A summary of the advection experiment using polynomial order = 4 and $\Delta t = 10^{-3}$ on the 3D cube domains meshed using different canonical elements individually. *Left to right*: The mesh profile, the unfiltered solution, and the filtered solution at timestep = 1000, respectively. *Top row*: Cube mesh with 27 hexahedra, *Middle row*: Cube mesh with 54 prisms, *Bottom row*: Cube mesh with 162 tetrahedra.

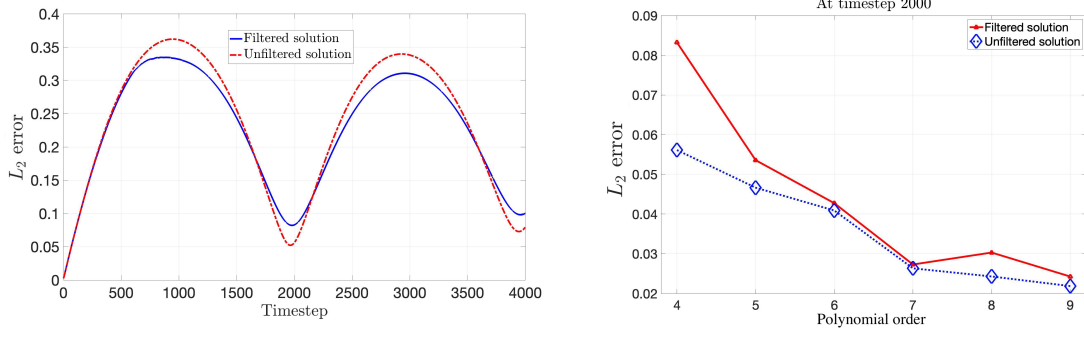


FIGURE 10. *Left:* Per timestep L_2 error for advection problem for polynomial order = 4 on a cube mesh of 24 tetrahedral elements. The timestep = 10^{-3} , and the total steps = 4000. At each timestep, for the L_2 error computation, we consider 10 quadrature points in each direction. *Right:* p-convergence for filtered and unfiltered L_2 error at timestep 2000 for the advection problem on a cube mesh of 24 tetrahedral elements.

$$\begin{aligned}
 \frac{\partial e_2}{\partial t} = & \underbrace{-\mathbf{u} \cdot \nabla e_2 + \nabla \cdot (D \nabla e_2)}_{\text{Transport by Advection Diffusion}} \\
 & + \underbrace{k_{e_2}^{on} e_2 (N_2 P^{b,a} + N_2 P^{se,a} - z_2^{mtot} - e_2^{mtot}) + k_{e_2}^{off} e_2^m}_{\text{Binding to platelet receptor}} \\
 (23) \quad & + \underbrace{(k_{z_5:e_2}^{cat} + k_{z_5:e_2}^-) [Z_5 : E_2] - k_{z_5:e_2}^+ z_5 e_2}_{\text{Activation of V}} \\
 & + \underbrace{(k_{z_7:e_2}^{cat} + k_{z_7:e_2}^-) [Z_7 : E_2] - k_{z_7:e_2}^+ z_7 e_2}_{\text{Activation of VII}} \\
 & + \underbrace{(k_{z_8:e_2}^{cat} + k_{z_8:e_2}^-) [Z_8 : E_2] - k_{z_8:e_2}^+ z_8 e_2}_{\text{Activation of VIII}}
 \end{aligned}$$

Figure 11 shows filtered and unfiltered runs of our implementation of this model and the resulting concentration profile of FP e_2 . At timestep = 2, we get the invalid (negative) values around the injury site because of the sharp concentration changes (refer to the slice at $y = 5$ in bottom left part of Figure 11). Therefore, the fluid phase thrombin concentration does not adhere to the desired structure (positivity), and the simulation does not succeed. If this experiment is allowed to continue, the effect of invalid values will compound over time, rendering the simulation results nonphysical and useless. At timestep 38, the simulation fails because of a blow-up resulting from the accumulation and propagation of negative values. At this point, the code for this simulation reports invalid values such as *NaN* (datatype: Not a Number).

When the filter is applied, it runs for significantly more timesteps (total 674) and terminates naturally. The right side of Figure 11 shows the state of FP e_2 at the depletion point of the chemicals involved.

6. CONCLUSIONS

We present a formalism that solves the problem of structure-preservation for PDE solutions in 2D and 3D. The construction and design of a post-processing structure-preserving filter are detailed and applied to multidimensional dG FEM solutions to different PDEs. A geometric interpretation of the mathematical foundation behind the filter is presented, followed by an algorithm to apply the proposed filter in a timestepping PDE framework. At the core of the filter lies the expensive requirement for global minimization on a weighted distance function that corresponds to the objective we optimize. We employ gradient descent with backtracking line search for the minimization in order to reduce the cost

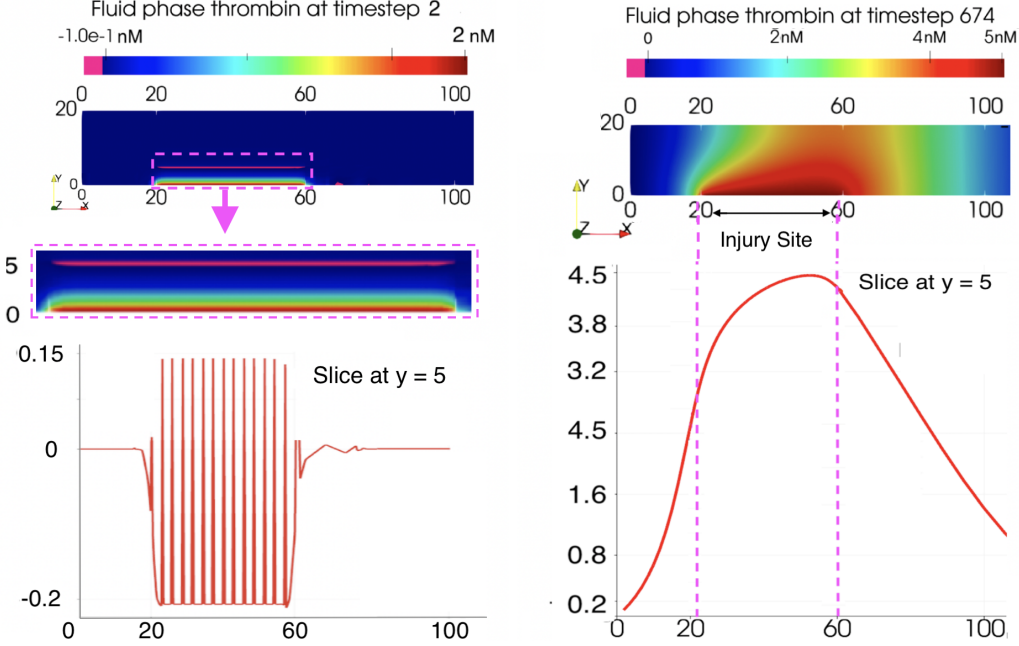


FIGURE 11. Top left: The unconstrained simulation solution becomes invalid, as shown in the inset at **timestep 2**. This causes a blowup at timestep 87, leading to the failure of the simulation. Top right: The constrained simulation runs for total **674 timesteps** until the depletion of the chemicals causes the model to terminate naturally. Bottom left: Profile of fluid phase thrombin in unconstrained simulation at timestep 2 on a domain slice at $y = 5$. Bottom right: Profile of fluid phase thrombin in constrained simulation at timestep 674 on a domain slice $y = 5$

of the filtering routine. To this end, we detail an investigative procedure to reduce the minimization cost by precomputing specific parameter values for the gradient descent approach. Using numerical examples, we compare the convergence rates with and without filter application for different problem sizes and domain structures. The percentage increase in time taken by the simulation experiments is computed to understand the cost of the filter and it is observed that the cost scales with the order and size of the problem. In the end, using a filtered solution to a mathematical-biology problem of platelet aggregation and blood coagulation, we provide evidence of the proposed method's efficacy and utility.

ACKNOWLEDGMENTS

V. Zala and R.M. Kirby acknowledge that their part of this research was sponsored by ARL under cooperative agreement number W911NF-12-2-0023. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. A. Narayan was partially supported by NSF DMS-1848508. This material is based upon work supported by both the National Science Foundation under Grant No. DMS-1439786 and the Simons Foundation Institute Grant Award ID 507536 while A. Narayan was in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the Spring 2020 semester.

REFERENCES

- [1] L. ALLEN AND R. C. KIRBY, *Bounds-constrained polynomial approximation using the Bernstein basis*, arXiv:2104.11819 [cs, math], (2021), <http://arxiv.org/abs/2104.11819>. arXiv: 2104.11819.

- [2] R. ANDERSON, V. DOBREV, T. KOLEV, D. KUZMIN, M. Q. DE LUNA, R. RIEBEN, AND V. TOMOV, *High-order local maximum principle preserving (mpp) discontinuous galerkin finite element method for the transport equation*, Journal of Computational Physics, 334 (2017), pp. 102–124.
- [3] R. W. ANDERSON, V. A. DOBREV, T. V. KOLEV, AND R. N. RIEBEN, *Monotonicity in high-order curvilinear finite element arbitrary lagrangian–eulerian remap*, International Journal for Numerical Methods in Fluids, 77 (2015), pp. 249–273.
- [4] L. ARMIJO, *Minimization of functions having Lipschitz continuous first partial derivatives.*, Pacific Journal of Mathematics, 16 (1966), pp. 1 – 3, <https://doi.org/pjm/1102995080>, <https://doi.org/>.
- [5] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric lagrange interpolation*, SIAM Review, 46 (2004), pp. 501–517.
- [6] D. P. BERTSEKAS, *Nonlinear programming*, Journal of the Operational Research Society, 48 (1997), pp. 334–334.
- [7] S. BOYD, S. P. BOYD, AND L. VANDENBERGHE, *Convex optimization*, Cambridge university press, 2004.
- [8] C. D. CANTWELL, D. MOXEY, A. COMERFORD, A. BOLIS, G. ROCCO, G. MENGALDO, D. DE GRAZIA, S. YAKOVLEV, J.-E. LOMBARD, D. EKELSCHOT, ET AL., *Nektar++: An open-source spectral/hp element framework*, Computer physics communications, 192 (2015), pp. 205–219.
- [9] J. B. CROCKETT, H. CHERNOFF, ET AL., *Gradient methods of maximization.*, Pacific Journal of Mathematics, 5 (1955), pp. 33–50.
- [10] H. B. CURRY, *The method of steepest descent for non-linear minimization problems*, Quarterly of Applied Mathematics, 2 (1944), pp. 258–261.
- [11] M. P. LAIU, C. D. HAUCK, R. G. MCCLARREN, D. P. O’LEARY, AND A. L. TITS, *Positive Filtered P_n Moment Closures for Linear Kinetic Equations*, SIAM Journal on Numerical Analysis, 54 (2016), pp. 3214–3238, <https://doi.org/10.1137/15M1052871>.
- [12] E. LAUGHTON, V. ZALA, A. NARAYAN, R. M. KIRBY, AND D. MOXEY, *Fast barycentric-based evaluation over spectral/hp elements*, Accepted by SIAM Journal on Scientific Computing, (2021), <https://arxiv.org/abs/2103.03594>.
- [13] K. LEIDERMAN AND A. L. FOGELSON, *Grow with the flow: a spatial–temporal model of platelet deposition and blood coagulation under flow*, Mathematical Medicine and Biology: a journal of the IMA, 28 (2011), pp. 47–84.
- [14] D. LIGHT AND D. DURRAN, *Preserving nonnegativity in discontinuous galerkin approximations to scalar transport via truncation and mass aware rescaling (tmar)*, Monthly Weather Review, 144 (2016), pp. 4771–4786.
- [15] C. LIU, C. WANG, AND Y. WANG, *A structure-preserving, operator splitting scheme for reaction-diffusion equations with detailed balance*, Journal of Computational Physics, 436 (2021), p. 110253, <https://doi.org/https://doi.org/10.1016/j.jcp.2021.110253>, <https://www.sciencedirect.com/science/article/pii/S0021999121001480>.
- [16] X.-D. LIU AND S. OSHER, *Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes i*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 760–779.
- [17] C. LOHMANN, D. KUZMIN, J. N. SHADID, AND S. MABUZA, *Flux-corrected transport algorithms for continuous galerkin methods based on high order bernstein finite elements*, Journal of Computational Physics, 344 (2017), pp. 151–186.
- [18] T. S. MOTZKIN AND I. J. SCHOENBERG, *The Relaxation Method for Linear Inequalities*, Canadian Journal of Mathematics, 6 (1954), pp. 393–404, <https://doi.org/10.4153/CJM-1954-038-x>, <https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/relaxation-method-for-linear-inequalities/8EDDF8B5F08B146AC71A5CF4F304A560>. Publisher: Cambridge University Press.
- [19] R. SANDERS, *A third-order accurate variation nonexpansive difference scheme for single nonlinear conservation laws*, Mathematics of Computation, 51 (1988), pp. 535–558.
- [20] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of computational physics, 77 (1988), pp. 439–471.
- [21] J. VAN DER VEGT, Y. XIA, AND Y. XU, *Positivity preserving limiters for time-implicit higher order accurate discontinuous galerkin discretizations*, arXiv preprint arXiv:1811.08620, (2018).
- [22] M. J. ZAHR AND P.-O. PERSSON, *An optimization based discontinuous galerkin approach for high-order accurate shock tracking*, in 2018 AIAA Aerospace Sciences Meeting, 2018, p. 0063.
- [23] V. ZALA, M. KIRBY, AND A. NARAYAN, *Structure-preserving function approximation via convex optimization*, SIAM Journal on Scientific Computing, 42 (2020), pp. A3006–A3029.
- [24] V. ZALA, M. KIRBY, AND A. NARAYAN, *Structure-preserving nonlinear filtering for continuous and discontinuous galerkin spectral/hp element methods*, SIAM Journal on Scientific Computing (Under review), (2020).
- [25] X. ZHANG, *On positivity-preserving high order discontinuous galerkin schemes for compressible navier–stokes equations*, Journal of Computational Physics, 328 (2017), pp. 301–343.
- [26] X. ZHANG AND C.-W. SHU, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), pp. 3091 – 3120.