



Sum factorization for fast integration of DPG matrices on prismatic elements

Jacob Badger^{*}, Stefan Henneking, Leszek Demkowicz

Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 E 24th St, Austin, TX 78712, USA

ABSTRACT

Higher order finite element (FE) methods provide significant advantages in a number of applications such as wave propagation, where high order shape functions help to mitigate pollution (dispersion) error. However, classical assembly of higher order systems is computationally burdensome, requiring the evaluation of many point quadrature schemes. When the Discontinuous Petrov-Galerkin (DPG) FE methodology is employed, the use of an enriched test space further increases the computational burden of system assembly, increasing the relevance of improved assembly techniques. Sum factorization—a technique that exploits the tensor-product structure of shape functions to accelerate numerical integration—was proposed in Ref. [10] for the assembly of DPG matrices on hexahedral elements that reduced the computational complexity from order $\mathcal{O}(p^9)$ to $\mathcal{O}(p^7)$ (where p denotes polynomial order). In this work we extend the concept of sum factorization to the construction of DPG matrices on prismatic elements by expressing prism shape functions as tensor products of 2D simplex and 1D interval shape functions. Unexpectedly, the resulting sum factorization routines on partially-tensorized prism shape functions achieve the same $\mathcal{O}(p^7)$ complexity as sum factorization on fully-tensorized hexahedra shape functions (as products of 1D interval shape functions) presented in Ref. [10]. Throughout this work we adhere to the theory of exact sequence energy spaces, proposing sum factorization routines for each of the 3D FE exact sequence energy spaces— H^1 , $H(\text{curl})$, $H(\text{div})$, and L^2 . Computational results for construction of the DPG Gram matrix on a prismatic element in each exact sequence energy space are presented, corroborating the expected $\mathcal{O}(p^7)$ complexity. Additionally, construction of the DPG system for an ultraweak Maxwell problem on a prismatic element is considered and a partially-tensorized sum factorization for hexahedral elements is proposed to improve implementational compatibility between hexahedral and prismatic elements.

1. Introduction

Construction of Finite Element (FE) systems relies on the accurate evaluation of integrals. Evaluation of quadrature schemes to approximate integrals can consume a significant portion of the total computational expense—especially when high-order elements are employed. In the case of the Discontinuous Petrov-Galerkin (DPG) methodology, the non-trivial expense of system assembly is further increased by use of an enriched test space. Thus, algorithms for efficient quadrature evaluation can result in significant computational savings in the construction of DPG systems. One such algorithm for the assembly of DPG systems presented in Ref. [10] achieves $\mathcal{O}(p^7)$ computational complexity for hexahedral-type elements (compared to $\mathcal{O}(p^9)$ complexity of standard routines) by decomposing the 3D hexahedron into the tensor product of three 1D line segments. In this article we extend these results to prismatic elements, achieving a similar $\mathcal{O}(p^7)$ complexity through the decomposition of prismatic elements into tensor products of 2D triangle, and 1D interval shape functions.

The present work builds on that of [10] (and earlier work by Kurtz [4]), presenting only results and details which are sufficiently different for the prismatic element than for the hexahedral element to merit additional discussion. The reader is directed there (and the contained refer-

ences) for a rigorous review of shape functions, the construction of DPG systems, and sum factorization. We begin this work by providing a less formal overview of the DPG method that may appeal to a more general audience. After providing a rough overview of the DPG method we will proceed as follows: In Section 2, polynomial subspaces with the desired tensor structure are defined and sum-factorization is outlined for each of the exact-sequence energy spaces. In Section 3, computational results are presented for both sum-factorized and standard assembly routines and the desired $\mathcal{O}(p^7)$ complexity is verified. We conclude in Section 4 with a summary of findings and suggestions for future work.

1.1. DPG: a rough sketch

The DPG methodology presented in this work—known fully as the *practical discontinuous Petrov-Galerkin method of optimal test functions*—can be understood in three contexts: as a method of optimal testing, as a generalized minimum residual method, and as a mixed method. To understand DPG as a method of optimal testing, consider the generalized variational problem

$$\begin{cases} \text{find } u \in U \text{ s.t.} \\ b(u, v) = l(v) \quad \forall v \in V \end{cases} \quad (1.1)$$

^{*} Corresponding author.

E-mail address: jcbadger@utexas.edu (J. Badger).

where U and V are the trial and test spaces respectively. Supposing bilinear functional $b(\cdot, \cdot)$ and linear functional $l(\cdot)$ satisfy continuity and compatibility constraints, the inf-sup condition,

$$\inf_{u \in U} \sup_{v \in V} \frac{|b(u, v)|}{\|u\|_U \|v\|_V} \geq \gamma > 0 \quad (1.2)$$

guarantees the well-posedness of this problem with stability constant γ . However, in a computational setting, finite-dimensional bases are used to represent spaces U and V , thus for this discretize problem to be well posed (1.2) must be satisfied when restricted to finite-dimensional subspaces U_h and V_h . Substituting U_h for U poses no problem for stability as restricting the infimum to a smaller set can only improve the stability constant γ . The same is not true for restricting V to V_h , thus a natural question would be to ask if there is a way to choose V_h so that the stability constant γ is not affected. This is the essence of the DPG method and, given a discrete trial space U_h and a test norm $\|\cdot\|_V$, the DPG methodology identifies the corresponding discrete test space V_h that realizes the supremum in (1.2) and essentially equates stability of the discrete problem to stability of the continuous (infinite-dimensional) problem. Identification of the optimal test space V_h is achieved by solving for the Riesz representation of each $u \in U_h$. In general, we cannot solve this Riesz representation problem on the original infinite-dimensional space V , so we introduce an enriched test space V^r by simply raising the polynomial degree p by a factor Δp (typically $\Delta p = 1, 2$; see Ref. [3] for details) resulting in an enriched or total polynomial order $p_r = p + \Delta p$. In this finite-dimensional space V^r the Riesz map is fully represented by the Gram matrix G ; thus solving the representation problem essentially reduces to constructing and inverting a Gram matrix on the enriched test basis.

The role of the Gram matrix in DPG methods can be observed directly by formulating (1.1) as a mixed problem. We defer to Ref. [3] for details of this formulation, but essentially yields a discrete system of coefficients of the form

$$\begin{pmatrix} G & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} s \\ u \end{pmatrix} = \begin{pmatrix} l \\ 0 \end{pmatrix} \quad (1.3)$$

where B is the stiffness matrix and l is the load. Note the analogy of system (1.3) with the system

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

for solving a general overdetermined system of equations $Ax = b$ in the least squares sense with residual r . Thus DPG can be identified as a generalized minimum residual method (see Ref. [8]) and two important properties can be observed from (1.3): i) DPG methods *always* produce hermitian positive definite systems of equations, and ii) DPG methods can be used to obtain a dual representation of the error which in turn can be used to obtain a robust error indicator for adaptive mesh refinement.

The primary benefits of DPG methods can then summarized as: stability, hermitian positive definiteness, and adaptability. These properties do not depend on a specific linear operator associated with a given variational problem, but are attained for any linear operator by virtue employing the DPG methodology. In addition to these remarkable properties, the DPG methodology permits additional variational formulations to be considered, including those with asymmetric functional settings such as the so-called primal and ultraweak variational formulations (see Ref. [2]).

The benefits of the DPG methodology are, however, not obtained for free. The cost of solving the full system (1.3) would be considerably higher than the cost of solving a typical Galerkin discretization. The practical DPG method reduces this cost by “breaking” the test space V_h (see Refs. [1,12] for details) to provide G a block diagonal form that can be statically condensed on an element level to obtain a reduced system

of equations (an example will be provided in Sec. 3.2). Such breaking of the test space necessitates the introduction of trace degrees of freedom, resulting in a condensed system that typically possesses roughly twice as many degrees of freedom as a classical FE discretizations. The benefits of DPG can however be leveraged to reduce this cost. For example, the hermitian positive definite structure of DPG systems permits use of advanced solution techniques including preconditioned conjugate gradient (CG) methods. One such CG solver proposed in Ref. [13] further leveraged the hierarchical structure of DPG-enabled adaptivity in a multigrid preconditioning scheme to produce a parallel linear-scaling solver for general DPG systems. We conclude this informal introduction by emphasizing—when stability, hermitian positive definiteness, and adaptivity are properly exploited, the DPG methodology can prove an efficient, robust, and general technique for solving very challenging problems.

2. Sum factorization

Similar to Ref. [10], this work follows the concept of exact sequence energy spaces. Thus, after a brief review of infinite-dimensional energy spaces, we define finite-dimensional polynomial subspaces with the desired tensor structure on which we can compute.

2.1. Exact sequence energy spaces

Let $X_0, X_1, \dots, X_{N_s}$ (with $N_s < \infty$) be a family of vector spaces, and for each $i = 1, \dots, N_s$, let $A_i : X_{i-1} \rightarrow X_i$ be a linear operator. The sequence or *complex* formed by these operators

$$X_0 \xrightarrow{A_1} X_1 \xrightarrow{A_2} \dots \xrightarrow{A_{N_s}} X_{N_s}$$

is said to be *exact* if, for $i = 1, 2, \dots, N_s - 1$, it holds that $R(A_i) = N(A_{i+1})$ where R and N denote the operator range and nullspace. By energy space we simply refer to a Hilbert space in which the solution of a variational problem is sought. Exact sequences can be formed (as detailed in Ref. [10]) for energy spaces in one, two, and three-dimensions as:

$$\begin{aligned} \text{1D :} \quad & H^1(\Omega) \xrightarrow{\partial} L^2(\Omega) \\ \text{2D :} \quad & H^1(\Omega) \xrightarrow{\nabla} H(\text{curl}, \Omega) \xrightarrow{\nabla_{\text{vts}}} L^2(\Omega) \\ & H^1(\Omega) \xrightarrow{\nabla_{\text{stv}}} H(\text{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega) \\ \text{3D :} \quad & H^1(\Omega) \xrightarrow{\nabla} H(\text{curl}, \Omega) \xrightarrow{\text{curl}} H(\text{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega). \end{aligned} \quad (2.1)$$

where the ∇_{vts} and ∇_{stv} denote the 2D vector-to-scalar and scalar-to-vector curl operators respectively, defined by $\nabla_{\text{vts}} = \partial_1(\cdot)_2 - \partial_2(\cdot)_1$ and $\nabla_{\text{stv}} = (\partial_2, -\partial_1)$, and Ω denotes a bounded, simply connected domain in $\mathbb{R}^N$ with $N = 1, 2, 3$ according to context.

Finite element methods introduce families of shape functions that span finite-dimensional subspaces of these energy spaces, thus it is natural to ask whether these discrete energy subspaces can be constructed to preserve exactness in the resulting sequence. Such exact discretizations—including the one employed here—have diverse applications in approximability theory. For example, in Ref. [9], exact sequence discretizations were used to produce geometry-preserving discretizations to improve coherence of plasma simulations. However, as noted in Ref. [10], the sum-factorization algorithms presented here only require that discrete energy subspaces form a complex, i.e. $R(A_i) \subset N(A_{i+1})$. This complex structure is assumed in the following definition of Piola transforms (pullback maps) which will enable integrals defined on element domains to be evaluated on a master domain.

2.2. Piola transforms (pullback maps)

Let $\hat{\mathcal{K}}$ denote the master domain of definition for prismatic shape functions, $\hat{\mathcal{K}} = \mathcal{T} \otimes \mathcal{I}$ where $\mathcal{T}$ is the 2D simplex $\{x \in \mathbb{R}^2 : x_1 > 0, x_2 > 0, x_1 + x_2 < 1\}$ and $\mathcal{I}$ is the 1D interval $(0, 1)$ in $\mathbb{R}$. Let $\mathbf{x}_{\mathcal{K}} : \hat{\mathcal{K}} \rightarrow \mathcal{K}$ be a diffeomorphic map, transforming the master element into a physical space element $\mathcal{K}$ with Jacobian

$$\mathcal{J} := \frac{\partial \mathbf{x}_{\mathcal{K}}}{\partial \xi}, \quad (2.2)$$

where ξ is the position vector in the master domain. For the remainder of this work, hats $\hat{\cdot}$ will be used to denote functions, domains, and vectorspaces defined on master domain $\hat{\mathcal{K}}$.

The Piola transforms (pullback maps) presented in Ref. [10] define transformations $T^{\text{grad}}, T^{\text{curl}}, T^{\text{div}}, T$ that transform corresponding energy spaces defined on $\hat{\mathcal{K}}$ to those defined on $\mathcal{K}$ as follows:

$$H^1(\hat{\mathcal{K}}) \ni \hat{u} \mapsto T^{\text{grad}} \hat{u} := \hat{u} \circ \mathbf{x}_{\mathcal{K}}^{-1} = u \in H^1(\mathcal{K}) \quad (2.3)$$

$$H(\text{curl}, \hat{\mathcal{K}}) \ni \hat{E} \mapsto T^{\text{curl}} \hat{E} := (\mathcal{J}^{-T} \hat{E}) \circ \mathbf{x}_{\mathcal{K}}^{-1} = E \in H(\text{curl}, \mathcal{K}) \quad (2.4)$$

$$H(\text{div}, \hat{\mathcal{K}}) \ni \hat{V} \mapsto T^{\text{div}} \hat{V} := (|\mathcal{J}|^{-1} \hat{V}) \circ \mathbf{x}_{\mathcal{K}}^{-1} = V \in H(\text{div}, \mathcal{K}) \quad (2.5)$$

$$L^2(\hat{\mathcal{K}}) \ni \hat{q} \mapsto T \hat{q} := (|\mathcal{J}|^{-1} \hat{q}) \circ \mathbf{x}_{\mathcal{K}}^{-1} = q \in L^2(\mathcal{K}). \quad (2.6)$$

These transformations permit integrals defined on element domains $\mathcal{K}$ to be transferred to the master domain $\hat{\mathcal{K}}$. Since application of these transformations is independent of element type or tensor structure, we defer to the derivation in Ref. [10] and will thus consider only spaces (and integrals) defined on the master domain $\hat{\mathcal{K}}$. The remainder of this section is dedicated to the outline of sum-factorization routines for assembly of the Gram matrix on prismatic elements in each of the considered energy spaces.

2.3. Tensor-product prismatic finite element shape functions

For each of the infinite dimensional energy spaces in (2.1), we define polynomial subspaces with the tensorized structure of the domain $\hat{\mathcal{K}} = \mathcal{T} \otimes \mathcal{I}$. Beginning with the 1D exact sequence on $\mathcal{I}$, we define polynomial subspaces:

$$\hat{W}_{\mathcal{I}}^p = \mathcal{P}^p(\mathcal{I}) \subsetneq H^1(\mathcal{I})$$

$$\downarrow \partial$$

$$\hat{Y}_{\mathcal{I}}^p = \mathcal{P}^{p-1}(\mathcal{I}) \subsetneq L^2(\mathcal{I})$$

where $\mathcal{P}^p(\mathcal{I})$ is the space of univariate polynomials on $\mathcal{I}$ with degree less than or equal to p .

Polynomial subspaces corresponding to each of the 2D exact sequences are then defined on the 2D simplex domain $\mathcal{T}$ as in Ref. [5] by:

$$\hat{W}_{\mathcal{T}}^p = \mathcal{P}^p(\mathcal{T}) \subsetneq H^1(\mathcal{T}) \quad \hat{W}_{\mathcal{T}}^p = \mathcal{P}^p(\mathcal{T}) \subsetneq H^1(\mathcal{T})$$

$$\downarrow \nabla = (\partial_1, \partial_2) \quad \downarrow \nabla_{\text{stv}} = (\partial_2, -\partial_1)$$

$$\hat{Q}_{\mathcal{T}}^p = \mathcal{N}^p(\mathcal{T}) \subsetneq H(\text{curl}, \mathcal{T}) \quad \hat{Y}_{\mathcal{T}}^p = \mathcal{RT}^p(\mathcal{T}) \subsetneq H(\text{div}, \mathcal{T})$$

$$\downarrow \nabla_{\text{vts}} = \partial_1(\cdot)_2 - \partial_2(\cdot)_1 \quad \downarrow \text{div} = \partial_1(\cdot)_1 + \partial_2(\cdot)_2$$

$$\hat{Y}_{\mathcal{T}}^p = \mathcal{P}^{p-1}(\mathcal{T}) \subsetneq L^2(\mathcal{T}) \quad \hat{Y}_{\mathcal{T}}^p = \mathcal{P}^{p-1}(\mathcal{T}) \subsetneq L^2(\mathcal{T}).$$

where $\mathcal{P}^p(\mathcal{T})$ is the space of bivariate polynomials on $\mathcal{T}$ of total order less than or equal to p and $\mathcal{N}^p, \mathcal{RT}^p$ denote the Nédélec and Raviart-Thomas spaces (respectively) for simplices with definitions:

$$\mathcal{N}^p(\mathcal{T}) = \mathcal{P}^{p-1} \otimes \left\{ E \in (\tilde{\mathcal{P}}^p)^2 : x \cdot E(x) = 0 \text{ for all } x \in \mathbb{R}^2 \right\},$$

$$\mathcal{RT}^p(\mathcal{T}) = \mathcal{P}^{p-1} \otimes \left\{ V \in (\tilde{\mathcal{P}}^p)^2 : V(x) = \phi(x)x \text{ with} \right.$$

$$\left. \phi \in \tilde{\mathcal{P}}^{p-1} \text{ for } V \text{ all } x \in \mathbb{R}^2 \right\},$$

where $\tilde{\mathcal{P}}^p$ denotes the space of homogeneous polynomials of order p .

A polynomial subspace can then be constructed for the prism's exact sequence by employing each of the 1D and 2D exact sequence polynomial subspaces:

$$\begin{aligned} \hat{W}^p &= \hat{W}_{\mathcal{T}}^{p_{12}} \otimes \hat{W}_{\mathcal{I}}^{p_3} \\ \downarrow \nabla &= (\partial_1, \partial_2, \partial_3) \\ \hat{Q}^p &= \hat{Q}_{\mathcal{T}}^{p_{12}} \otimes \hat{W}_{\mathcal{I}}^{p_3} \times \hat{W}_{\mathcal{T}}^{p_{12}} \otimes \hat{Y}_{\mathcal{I}}^{p_3} \\ \downarrow \text{curl} &= (\partial_2(\cdot)_3 - \partial_3(\cdot)_2, \partial_3(\cdot)_1 - \partial_1(\cdot)_3, \partial_1(\cdot)_2 - \partial_2(\cdot)_1) \\ \hat{V}^p &= \hat{Y}_{\mathcal{T}}^{p_{12}} \otimes \hat{W}_{\mathcal{I}}^{p_3} \times \hat{W}_{\mathcal{T}}^{p_{12}} \otimes \hat{Y}_{\mathcal{I}}^{p_3} \\ \downarrow \text{div} &= \partial_1(\cdot)_1 + \partial_2(\cdot)_2 + \partial_3(\cdot)_3 \\ \hat{Y}^p &= \hat{Y}_{\mathcal{T}}^{p_{12}} \otimes \hat{Y}_{\mathcal{I}}^{p_3}. \end{aligned} \quad (2.7)$$

The use of superscript p_{12} to denote the order of 2D simplex spaces and p_3 to denote the order of 1D interval spaces here is used to indicate that p_{12} is the order in the first and second spatial dimensions, while p_3 is the order in the third spatial dimension of the master domain. Such a use of subscripts will follow throughout this document, with its purpose becoming increasingly evident when Gram matrix assembly routines are introduced. Before moving to the construction of the Gram matrix for DPG systems for the various energy spaces, we emphasize the importance of the 3D exact sequence polynomial subspaces in (2.7) as they define the structure of prismatic shape functions used to compute in the various energy spaces.

The Gram matrix considered here for each finite dimensional Hilbert space $\mathcal{H}$ is defined identically to Ref. [10], and the construction for each space is presented in the order defined therein—deviating from the exact sequence order in favor of simplicity.

2.4. Space L^2

Let $\{v\}_{l=0}^{\dim Y^p-1}$ be a basis for Y^p , where $\dim Y^p = \frac{1}{2}(p_{12} + 1)p_{12}p_3$. Elements of basis $\{v_l\}$ in Y^p are represented in the tensorized form of (2.7) as $v_l = v_{i_{12}i_3}$ where $v_{i_{12}} \in \hat{Y}_{\mathcal{T}}^{p_{12}}$ with $i_{12} = 0, \dots, \frac{1}{2}(p_{12} + 1)p_{12} - 1$; where $v_{i_3} \in \hat{Y}_{\mathcal{I}}^{p_3}$ with $i_3 = 0, \dots, p_3 - 1$; and where l is some unique integer identifier dependent on i_{12} and i_3 such that $0 \leq l < \dim Y^p$. The L^2 Gram matrix is then constructed for ordered pairs of basis elements (v_l, v_j) as:

$$\begin{aligned} G_{IJ} &= (v_l, v_j)_{\mathcal{K}} \\ &= \int_{\hat{\mathcal{K}}} \hat{v}_l(\xi) \hat{v}_j(\xi) |\mathcal{J}(\xi)|^{-1} d^3 \xi \\ &= \int_{\mathcal{I}} \int_{\mathcal{T}} \hat{v}_l(\xi_1, \xi_2, \xi_3) \hat{v}_j(\xi_1, \xi_2, \xi_3) |\mathcal{J}(\xi_1, \xi_2, \xi_3)|^{-1} d^2(\xi_1, \xi_2) d\xi_3 \\ &= \int_{\mathcal{I}} v_{i_3}(\xi_3) v_{j_3}(\xi_3) \left\{ \int_{\mathcal{T}} v_{i_{12}}(\xi_1, \xi_2) v_{j_{12}}(\xi_1, \xi_2) |\mathcal{J}(\xi_1, \xi_2, \xi_3)|^{-1} d^2(\xi_1, \xi_2) \right\} d\xi_3 \end{aligned} \quad (2.8)$$

where the evaluation of all inner-products and shape functions have been transferred to the master domain $\hat{\mathcal{K}}$ by means of Piola transforms (see (2.16) in Ref. [10]) and the integral in the last line has been factored according to Fubini's theorem.

Sum factorization proceeds by computing and storing the inner area integral first for all combinations of 2D simplicial shape functions in $Y_{\mathcal{T}}^{p_{12}}$ before evaluating any outer integral terms. To accomplish this, we

introduce a sequence of auxiliary functions to compute the final Gram matrix:

$$\begin{aligned} \mathcal{G}_{i_{12}j_{12}}^A(\xi_3) &:= \int_{\mathcal{T}} v_{i_{12}}(\xi_1, \xi_2) v_{j_{12}}(\xi_1, \xi_2) |J(\xi_1, \xi_2, \xi_3)|^{-1} d^2(\xi_1, \xi_2) \\ \Rightarrow \mathbf{G}_U &= \mathcal{G}_{i_{12}j_{12}i_3j_3} := \int_I v_{i_3}(\xi_3) v_{j_3}(\xi_3) \mathcal{G}_{i_{12}j_{12}}^A(\xi_3) d\xi_3 \end{aligned} \quad (2.9)$$

Discretizing the preceding integrals by means of a quadrature rule leads to corresponding auxiliary matrices:

$$\begin{aligned} \mathcal{G}_{i_{12}j_{12}}^A(\xi_3) &\approx \sum_{m=1}^M v_{i_{12}}(\xi_1^m, \xi_2^m) v_{j_{12}}(\xi_1^m, \xi_2^m) |J(\xi_1^m, \xi_2^m, \xi_3)|^{-1} w_{12}^m \\ \mathcal{G}_{i_{12}j_{12}i_3j_3} &\approx \sum_{l=1}^L v_{i_3}(\xi_3^l) v_{j_3}(\xi_3^l) \mathcal{G}_{i_{12}j_{12}}^A(\xi_3^l) w_3^l. \end{aligned} \quad (2.10)$$

We note here that our definition of auxiliary functions and matrices is not unique. Indeed, a reverse factorization of (2.8) with the area integral on the outside would lead to a definition of $\mathcal{G}^A$ dependent on ξ_1 and ξ_2 . As will become clear shortly however, this factorization leads to order $\mathcal{O}(p^8)$ complexity instead of the desired $\mathcal{O}(p^7)$.

Evaluation of the auxiliary sequence (2.10) follows similar logic as was presented for the hexahedra: all three components of a quadrature point ξ_{lm} are fixed, $\mathcal{G}^A$ is evaluated, then $\mathcal{G}$ is evaluated. This process is iterated until each quadrature point has been evaluated. Note in particular that $\mathcal{G}^A$ is evaluated once for each 1D quadrature point ξ_3^l .

The assembly procedure for the L^2 Gram matrix is outlined in Algorithm 1. In the following, standard Gaussian quadrature rules for the unit interval and triangle are used. Constants L and M will denote the number of 1D interval and 2D triangle quadrature points respectively (thus L is $\mathcal{O}(p)$ and M is $\mathcal{O}(p^2)$). Indices l and m will be used as in (2.10) to traverse 1D and 2D quadrature points respectively. Finally, indices j_{12} and j_3 traverse test space degrees of freedom on the triangle and interval respectively. In the context of the Gram matrix, indices i_{12} and i_3 also traverse test space degrees of freedom (see Remark 2.4); however, for assembling the stiffness matrix $\mathbf{B}$, these indices would instead traverse trial space degrees of freedom. The physical coordinate $\mathbf{x}_{lm}$ corresponding to quadrature point ξ_{lm} returned by the function “geometry” is not used during assembly of the Gram matrix; it is however used in assembly of the stiffness matrix to obtain material data. Thus, Algorithm 1 can be adapted for assembly of the stiffness matrix by incorporating the trial space and material data as described. For subsequent energy spaces, such algorithms will not be outlined explicitly as they follow rather naturally from the loop structure of Algorithm 1 and from the corresponding sum-factorization algorithms for the hexahedra presented in Ref. [10].

Remark 2.1 When applying the DPG methodology there can be some confusion over the definition of trial and test spaces. This confusion arises because we are essentially solving two variational problems simultaneously: first the Riesz problem to identify the optimal test functions, and second the original variational problem in which we employ the optimal test functions. Some clarity can be achieved by considering the difference between the (optimal) test space and the enriched test space. However, the Riesz problem (corresponding to the Gram matrix) has a symmetric functional setting in which both trial and test space correspond to the enriched test space. To avoid confusion ‘trial space’ will denote the trial space of the original variational problem, and ‘test space’ will denote the enriched test space. We again emphasize that in the following algorithm for assembly of the Gram matrix both i and j indices loop over test degrees of freedom.

To clearly illustrate how the preceding algorithm achieves $\mathcal{O}(p^7)$ complexity, in Algorithm 1 (Naked) we identify only the ‘naked’ loops with their corresponding complexity. Consideration of the naked loops immediately reveals the $\mathcal{O}(p^7)$ scaling both for the computation of the auxiliary matrix $\mathcal{G}^A$, and for the computation of the final Gram matrix $\mathbf{G}$.

A similar analysis of sum-factorization algorithms for the hexahedra in Ref. [10] reveals instead a scaling of $\mathcal{O}(p^5)$ for computation of auxiliary matrix $\mathcal{G}^A$, $\mathcal{O}(p^6)$ for computation of the additional auxiliary matrix $\mathcal{G}^B$, and $\mathcal{O}(p^7)$ for the final computation of the Gram matrix. Thus, while the computation of auxiliary matrices for the full sum-factorization of the hexahedra has an order less computational-complexity than for the partial sum-factorization of the prism, both achieve a total $\mathcal{O}(p^7)$ complexity. We reiterate here that if (2.8) was factored differently (as discussed previously in this section) that the resulting complexity would be $\mathcal{O}(p^8)$, as could be seen by inverting the order of loops on l and m in Algorithm 1 (Naked).

2.5. Space H^1

To consider the H^1 energy space, let $\{\phi_I\}_{I=0}^{\dim W^p-1}$ be a basis for W^p , where $\dim W^p = \frac{1}{2}(p_{12}+2)(p_{12}+1)(p_3+1)$. Elements of basis $\{\phi_I\}$ can again be represented in the tensorized form of (2.7) as $\phi_I = u_{i_{12}} \chi_{i_3}$ where $u_{i_{12}} \in \hat{W}_T^{p_{12}}$ with $i_{12} = 0, \dots, \frac{1}{2}(p_{12}+2)(p_{12}+1) - 1$; where $\chi_{i_3} \in \hat{W}_I^{p_3}$ with $i_3 = 0, \dots, p_3$; and where I is a unique integer identifier dependent on i_{12} and i_3 such that $0 \leq I < \dim W^p$. The H^1 Gram matrix is then constructed for ordered pairs of basis elements (ϕ_I, ϕ_J) as:

$$\begin{aligned} \mathbf{G}_{IJ}^{\text{grad}} &= (\phi_I, \phi_J)_{H^1(\mathcal{K})} \\ &= \int_{\hat{\mathcal{K}}} \hat{\phi}_I(\xi) \hat{\phi}_J(\xi) |J(\xi)| d^3 \xi + \int_{\hat{\mathcal{K}}} \left[\hat{\nabla} \hat{\phi}_I(\xi) \right]^T \mathcal{D}(\xi) \left[\hat{\nabla} \hat{\phi}_J(\xi) \right] |J(\xi)| d^3 \xi \\ &= \int_I \int_{\mathcal{T}} \hat{\phi}_I(\xi) \hat{\phi}_J(\xi) |J(\xi)| d(\xi_1, \xi_2) d\xi_3 \\ &\quad + \int_I \int_{\mathcal{T}} \left[\hat{\nabla} \hat{\phi}_I(\xi) \right]^T \mathcal{D}(\xi) \left[\hat{\nabla} \hat{\phi}_J(\xi) \right] |J(\xi)| d^2(\xi_1, \xi_2) d\xi_3 \end{aligned} \quad (2.11)$$

where $\mathcal{D}(\xi) := J^{-1}(\xi) J^{-T}(\xi)$ and

$$\hat{\nabla} \hat{\phi}_I = \begin{pmatrix} \partial_{\xi_1} u_{i_{12}}(\xi_1, \xi_2) \chi_{i_3}(\xi_3) \\ \partial_{\xi_2} u_{i_{12}}(\xi_1, \xi_2) \chi_{i_3}(\xi_3) \\ u_{i_{12}}(\xi_1, \xi_2) \chi'_{i_3}(\xi_3) \end{pmatrix}. \quad (2.12)$$

The first integral in (2.11) closely resembles that in (2.8) for the L_2 case (note however the lack of the inverse on $|J(\xi)|$ in the H^1 case) and its factorization will not be repeated here. The sum-factorization of this integral is approximated by the auxiliary matrix sequence:

$$\begin{aligned} \mathcal{G}_{i_{12}j_{12}}^{\text{grad}A}(\xi_3) &\approx \sum_{n=1}^N v_{i_{12}}(\xi_1^n, \xi_2^n) v_{j_{12}}(\xi_1^n, \xi_2^n) |J(\xi_1^n, \xi_2^n, \xi_3)| w_{12}^n \\ \mathcal{G}_{i_{12}j_{12}i_3j_3}^{\text{grad}} &\approx \sum_{m=1}^M v_{i_3}(\xi_3^m) v_{j_3}(\xi_3^m) \mathcal{G}_{i_{12}j_{12}}^{\text{grad}A}(\xi_3^m) w_3^m. \end{aligned} \quad (2.13)$$

To factor the second integral in (2.11) according to Fubini’s theorem we first write (2.12) as a product of 3×3 and 3×1 arrays:

$$\widehat{\mathbf{v}}_I = U_{i_{12}}(\xi_1, \xi_2) X_{i_3}(\xi_3) = \begin{pmatrix} \partial_{\xi_1} u_{i_{12}}(\xi_1, \xi_2) & - & - \\ - & \partial_{\xi_2} u_{i_{12}}(\xi_1, \xi_2) & - \\ - & - & u_{i_{12}}(\xi_1, \xi_2) \end{pmatrix} \begin{pmatrix} \chi_{i_3}(\xi_3) \\ \chi_{i_3}(\xi_3) \\ \chi'_{i_3}(\xi_3) \end{pmatrix} \quad (2.14)$$

Such a definition naturally leads to the factorization of (2.11) as:

$$\begin{aligned} & \int_{\mathcal{I}} \int_{\mathcal{T}} [\widehat{\mathbf{v}}_I(\xi)]^T \mathcal{D}(\xi) [\widehat{\mathbf{v}}_J(\xi)] |\mathcal{J}(\xi)| d^2(\xi_1, \xi_2) d\xi_3 \\ &= \int_{\mathcal{I}} X_{i_3}^T(\xi_3) \left\{ \int_{\mathcal{T}} U_{i_{12}}^T(\xi_1, \xi_2) \mathcal{D}(\xi) U_{i_{12}}(\xi_1, \xi_2) |\mathcal{J}(\xi)| d^2(\xi_1, \xi_2) \right\} X_{i_3}(\xi_3) d\xi_3 \end{aligned} \quad (2.15)$$

To proceed with sum-factorization we introduce an auxiliary function sequence for the computation of (2.15):

$$\begin{aligned} \overline{\mathcal{G}}_{i_{12}j_{12}}^{\text{gradA}}(\xi_3) &:= \int_{\mathcal{T}} U_{i_{12}}^T(\xi_1, \xi_2) \mathcal{D}(\xi) U_{j_{12}}(\xi_1, \xi_2) |\mathcal{J}(\xi)| d^2(\xi_1, \xi_2) \\ \overline{\mathcal{G}}_{i_{12}j_{12}i_{3j_3}}^{\text{grad}} &:= \int_{\mathcal{I}} X_{i_3}^T(\xi_3) \overline{\mathcal{G}}_{i_{12}j_{12}}^{\text{gradA}}(\xi_3) X_{j_3}(\xi_3) d\xi_3 \end{aligned} \quad (2.16)$$

Note in particular that the integral for $\overline{\mathcal{G}}^{\text{gradA}}$ in (2.16) evaluates a 3×3 matrix. In addition to discretizing integrals through a quadrature rule we introduce indices $a, b \in \{1, 2, 3\}$ to store matrix components of $\overline{\mathcal{G}}^{\text{gradA}}$. The resulting discretized auxiliary function sequence can be represented in terms of arrays as:

$$\begin{aligned} \overline{\mathcal{G}}_{abi_{12}j_{12}}^{\text{gradA}}(\xi_3) &\approx \sum_{m=1}^M U_{i_{12}aa}(\xi_1^m, \xi_2^m) \mathcal{D}_{ab}(\xi_1^m, \xi_2^m, \xi_3) U_{j_{12}bb}(\xi_1^m, \xi_2^m) |\mathcal{J}(\xi_1^m, \xi_2^m, \xi_3)| w_{12}^m \\ \overline{\mathcal{G}}_{i_{12}j_{12}i_{3j_3}}^{\text{grad}} &\approx \sum_{l=1}^L \sum_{a=1}^3 \sum_{b=1}^3 X_{i_3a}(\xi_3^l) \overline{\mathcal{G}}_{abi_{12}j_{12}}^{\text{gradA}}(\xi_3^l) X_{j_3b}(\xi_3^l) w_{13}^l. \end{aligned} \quad (2.17)$$

were subscripts a, b denote vector indices.

The Gram matrix is finally calculated by the addition of (2.13) and (2.17):

$$\mathbf{G}_{IJ}^{\text{grad}} = \mathcal{G}_{i_{12}j_{12}i_{3j_3}}^{\text{grad}} + \overline{\mathcal{G}}_{i_{12}j_{12}i_{3j_3}}^{\text{grad}}. \quad (2.18)$$

2.6. Space $H(\text{div})$

In considering the final two energy spaces, some additional difficulty is presented by the structure of the polynomial subspaces in (2.7) from which the shape functions are defined. In particular, in both $H(\text{div})$ and

$$\begin{aligned} \mathbf{G}_{IJ}^{\text{div}} &= (\partial_I, \partial_J)_{H(\text{div})} \\ &= \int_{\widehat{\mathcal{K}}} \widehat{\partial}_I(\xi)^T \mathcal{C}(\xi) \widehat{\partial}_J(\xi) |\mathcal{J}(\xi)|^{-1} d^3 \xi + \int_{\widehat{\mathcal{K}}} \widehat{\text{div}} \widehat{\partial}_I(\xi) \widehat{\text{div}} \widehat{\partial}_J(\xi) |\mathcal{J}(\xi)|^{-1} d^3 \xi \\ &= \int_{\mathcal{I}} \int_{\mathcal{T}} \widehat{\partial}_I(\xi)^T \mathcal{C}(\xi) \widehat{\partial}_J(\xi) |\mathcal{J}(\xi)|^{-1} d^2(\xi_2, \xi_1) d\xi_3 + \int_{\mathcal{I}} \int_{\mathcal{T}} \widehat{\text{div}} \widehat{\partial}_I(\xi) \widehat{\text{div}} \widehat{\partial}_J(\xi) |\mathcal{J}(\xi)|^{-1} d^2(\xi_2, \xi_1) d\xi_3. \end{aligned} \quad (2.19)$$

$H(\text{curl})$ spaces, shape functions come from two families of shape functions (note that the definition of families here only loosely coincides with the definition in Ref. [5]).

The definition of prism shape function families is indicated naturally by the definition of polynomial subspace $\widehat{\mathcal{V}}^p$ in (2.7):

$$\widehat{\mathcal{V}}^p = \underbrace{\widehat{\mathcal{V}}_{\mathcal{T}}^{p_{12}} \otimes \widehat{\mathcal{Y}}_{\mathcal{I}}^{p_3}}_{\text{Family 1}} \times \underbrace{\widehat{\mathcal{Y}}_{\mathcal{T}}^{p_{12}} \otimes \widehat{\mathcal{W}}_{\mathcal{I}}^{p_3}}_{\text{Family 2}},$$

Various implementations are possible for incorporating the two family structure of this space including: decomposition of Gram matrix into blocks based on family interactions (i.e. [fam 1, fam1], [fam1, fam2], ...), or by sequential treatment of simplicial shape functions and a logical treatment of interval shape functions. The second approach is outlined here due to its relatively compact representation.

Let $\{\widehat{\partial}_I\}_{I=0}^{\dim \widehat{\mathcal{V}}^p - 1}$ be a basis of prismatic shape functions spanning $\widehat{\mathcal{V}}^p$ that is partitioned into two families as depicted above. Let $N_{12}^1 = \dim \widehat{\mathcal{V}}_{\mathcal{T}}^{p_{12}}$ denote the number of 2D simplicial shape functions in $\widehat{\mathcal{V}}_{\mathcal{T}}^{p_{12}}$ used in defining family 1, and $N_{12}^2 = \dim \widehat{\mathcal{Y}}_{\mathcal{T}}^{p_{12}}$ denote the number of 2D simplicial shape functions in $\widehat{\mathcal{Y}}_{\mathcal{T}}^{p_{12}}$ used in defining family 2. The

approach is outlined as follows: we define $i_{12} = 0, \dots, N_{12}^1 + N_{12}^2 - 1$ to enumerate all simplicial components of shape functions in $\widehat{\mathcal{V}}^p$. The appropriate univariate shape function space ($\widehat{\mathcal{Y}}_{\mathcal{I}}^{p_3}$ or $\widehat{\mathcal{W}}_{\mathcal{I}}^{p_3}$) is then determined by i_{12} , and a Gram index I can be uniquely defined given i_{12} and i_3 . Table 1 defines the shape functions, their divergence, and indices. Note in particular that values $i_{12} < N_{12}^1$ correspond to the first family of shape functions while values $i_{12} \geq N_{12}^1$ correspond to the second family of shape functions.

The Gram matrix $\mathbf{G}_{IJ}^{\text{div}}$ is then calculated for ordered pairs of shape functions (∂_I, ∂_J) as:

where $\mathcal{C}$ is the symmetric matrix given by $\mathcal{C}(\xi) := \mathcal{J}^T(\xi) \mathcal{J}(\xi) = \mathcal{D}(\xi)^{-1}$.

The first integral in (2.19) resembles that of (2.15) and can be factored similarly by writing $\widehat{\partial}_I$ as a product of a 3×3 array $W_{i_{12}}(\xi_1, \xi_2)$ and 3×1 array $X_{i_3}(\xi_3, i_{12})$ as:

$$\widehat{\partial}_I = W_{i_{12}}(\xi_1, \xi_2) X_{i_3}(\xi_3, i_{12})$$

Table 1
Definition of two families of prismatic shape functions for $\hat{V}^p$

Family 1	Family 2
$\hat{\theta}_I = \begin{pmatrix} V_{i_{12},1}(\xi_1, \xi_2) \nu_{i_3}(\xi_3) \\ V_{i_{12},2}(\xi_1, \xi_2) \nu_{i_3}(\xi_3) \\ 0 \end{pmatrix}$ $\widehat{\text{div}} \hat{\theta}_I = (\partial_x V_{i_{12}}(\xi_1, \xi_2) + \partial_y V_{i_{12}}(\xi_1, \xi_2)) \nu_{i_3}(\xi_3)$ $= \text{div}(V_{i_{12}}(\xi_1, \xi_2)) \nu_{i_3}(\xi_3)$ where $V_{i_{12}} \in \hat{V}_T^{p_{12}}, N_{12}^1 = (p_{12} + 2)p_{12}, 0 \leq i_{12} < N_{12}^1$ and $\nu_{i_3} \in \hat{Y}_T^{p_3}, N_3^1 = p_3, 0 \leq i_3 < N_3^1$;	$\hat{\theta}_I = \begin{pmatrix} 0 \\ 0 \\ \nu_{i_{12}}(\xi_1, \xi_2) \chi_{i_3}(\xi_3) \end{pmatrix}$ $\widehat{\text{div}} \hat{\theta}_I = \nu_{i_{12}}(\xi_1, \xi_2) \chi'_{i_3}(\xi_3)$ where $\nu_{i_{12}} \in \hat{Y}_T^{p_{12}}, N_{12}^2 = \frac{1}{2}p_{12}(p_{12} + 1), N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2$ and $\chi_{i_3} \in \hat{W}_T^{p_3}, N_3^2 = p_3 + 1, 0 \leq i_3 < N_3^2$;

Algorithm 1 Computation of the L^2 Gram Matrix - (partial) sum factorization.

```

procedure L2GRAMTENSOR( $i_{el}, G$ ) ▷ Compute G for element No.  $i_{el}$  - Partial sum factorization
   $N_{12} = 1/2(p_{12} + 1)p_{12}$  ▷ Calculate DoFs for 2D simplex
   $N_3 = p_3$  ▷ Calculate DoFs for 1D simplex
  call setquadrature1D( $i_{el}, p_3 - 1; L, \{\xi_3^l, w^l\}$ )
  call setquadrature2D( $i_{el}, p_{12} - 1; M, \{(\xi_1^m, \xi_2^m), w^m\}$ )
   $G \leftarrow 0$  ▷ Initialize Gram Matrix
  for  $l = 1$  to  $L$  do
    call Shape1L2( $\xi_3^l, p_3; \{\nu_{i_3}(\xi_3^l)\}$ ) ▷ Evaluate 1D shape functions at  $\xi_3^l$ 
     $G^A \leftarrow 0$ 
    for  $m = 1$  to  $M$  do
      call Shape2L2( $\xi_1^m, p_{12}; \{v_{i_{12}}(\xi_1^m, \xi_2^m)\}$ ) ▷ Evaluate 2D shape functions at  $(\xi_1^m, \xi_2^m)$ 
       $\xi_{lm} \leftarrow (\xi_1^m, \xi_2^m, \xi_3^l)$ 
      call geometry( $\xi_{lm}, i_{el}; \mathbf{x}_{lm}, \mathcal{J}(\xi_{lm}), \mathcal{J}^{-1}(\xi_{lm}), |\mathcal{J}|$ ) ▷ Compute  $\mathbf{x}_{lm} = \mathbf{x}_K(\xi_{lm})$  and Jacobian
      for  $j_{12} = 0$  to  $N_{12}$  do
        for  $i_{12} = j_{12}$  to  $N_{12}$  do
           $G_{i_{12}j_{12}}^A \leftarrow G_{i_{12}j_{12}}^A + v_{i_{12}}(\xi_1^m, \xi_2^m) v_{j_{12}}(\xi_1^m, \xi_2^m) |\mathcal{J}|^{-1} w^m$ 
      for  $j_3 = 0$  to  $N_3 - 1$  do
        for  $i_3 = j_3$  to  $N_3 - 1$  do
          for  $j_{12} = 0$  to  $N_{12} - 1$  do
            for  $i_{12} = j_{12}$  to  $N_{12} - 1$  do
               $G_{i_{12}j_{12}i_3j_3} \leftarrow G_{i_{12}j_{12}i_3j_3} + \nu_{i_3}(\xi_3^l) \nu_{j_3}(\xi_3^l) G_{i_{12}j_{12}}^A(\xi_3^l) w^l$ 
    return G

```

where

$$W_{i_{12}}(\xi_1, \xi_2) = \begin{cases} \begin{pmatrix} V_{i_{12},1}(\xi_1, \xi_2) & - & - \\ - & V_{i_{12},2}(\xi_1, \xi_2) & - \\ - & - & 0 \end{pmatrix} & \text{if } 0 \leq i_{12} < N_{12}^1, \\ \begin{pmatrix} 0 & - & - \\ - & 0 & - \\ - & - & \nu_{i_{12}}(\xi_1, \xi_2) \end{pmatrix} & \text{if } N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2, \end{cases} \quad (2.20)$$

Algorithm 1: (Naked)

```

for  $l = 1$  to  $L$  do  $\mathcal{O}(p)$ 
  for  $m = 1$  to  $M$  do  $\mathcal{O}(p^2)$ 
    for  $j_{12} = 0$  to  $N_{12} - 1$  do  $\mathcal{O}(p^2)$ 
      for  $i_{12} = j_{12}$  to  $N_{12} - 1$  do  $\mathcal{O}(p^2)$  ▷ Total:  $\mathcal{O}(p^7)$ 
    for  $j_3 = 0$  to  $N_3 - 1$  do  $\mathcal{O}(p)$ 
      for  $i_3 = j_3$  to  $N_3 - 1$  do  $\mathcal{O}(p)$ 
        for  $j_{12} = 0$  to  $N_{12} - 1$  do  $\mathcal{O}(p^2)$ 
          for  $i_{12} = j_{12}$  to  $N_{12} - 1$  do  $\mathcal{O}(p^2)$  ▷ Total:  $\mathcal{O}(p^7)$ 

```

and

$$X_{i_3}(\xi_3, i_{12}) = \begin{cases} \begin{pmatrix} v_{i_3}(\xi_3) \\ v_{i_3}(\xi_3) \\ 0 \end{pmatrix} & \text{if } 0 \leq i_{12} < N_{12}^1, \\ \begin{pmatrix} 0 \\ 0 \\ \chi_{i_3}(\xi_3) \end{pmatrix} & \text{if } N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2. \end{cases} \quad (2.21)$$

Sum factorization then proceeds by introducing the auxiliary function sequence for computation of the first integral term in (2.19) as:

$$\begin{aligned} \mathcal{G}_{i_{12}j_{12}}^{\text{divA}}(\xi_3) &:= \int_{\mathcal{T}} W_{i_{12}}^T(\xi_1, \xi_2) C(\xi_1, \xi_2, \xi_3) W_{j_{12}}(\xi_1, \xi_2) |J(\xi_1, \xi_2, \xi_3)|^{-1} d^2(\xi_1, \xi_2) \\ \mathcal{G}_{i_{12}j_{12}i_3j_3}^{\text{div}} &:= \int_I X_{i_3}^T(\xi_3, i_{12}) \mathcal{G}_{i_{12}j_{12}}^{\text{divA}}(\xi_3) X_{j_3}(\xi_3, j_{12}) d\xi_3 \end{aligned} \quad (2.22)$$

Discretization of this auxiliary sequence is accomplished similar to (2.17) by introducing indices $a, b \in \{1, 2, 3\}$ and will not be repeated here.

Computation of the second integral term in (2.19) can be simplified by introducing functions $w_{i_{12}}(\xi_1, \xi_2)$ and $x_{i_3}(\xi_3, i_{12})$ to treat both families of shape functions simultaneously:

$$w_{i_{12}}(\xi_1, \xi_2) = \begin{cases} \text{div}(V_{i_{12}}(\xi_1, \xi_2)) & \text{if } 0 \leq i_{12} < N_{12}^1, \\ v_{i_{12}}(\xi_1, \xi_2) & \text{if } N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2, \end{cases}$$

$$\begin{aligned} G_{IJ}^{\text{curl}} &= (\psi_I, \psi_J)_{H(\text{curl})} \\ &= \int_{\hat{\mathcal{K}}} \hat{\psi}_I(\xi)^T \mathcal{D}(\xi) \hat{\psi}_J(\xi) |J(\xi)| d^3\xi + \int_{\hat{\mathcal{K}}} [\widehat{\text{curl}} \hat{\psi}_I(\xi)]^T C(\xi) [\widehat{\text{curl}} \hat{\psi}_J(\xi)] |J(\xi)|^{-1} d^3\xi \\ &= \int_I \int_{\mathcal{T}} \hat{\psi}_I(\xi)^T \mathcal{D}(\xi) \hat{\psi}_J(\xi) |J(\xi)| d^2(\xi_2, \xi_1) d\xi_3 \\ &\quad + \int_I \int_{\mathcal{T}} [\widehat{\text{curl}} \hat{\psi}_I(\xi)]^T C(\xi) [\widehat{\text{curl}} \hat{\psi}_J(\xi)] |J(\xi)|^{-1} d^2(\xi_2, \xi_1) d\xi_3 \end{aligned} \quad (2.25)$$

and

$$x_{i_3}(\xi_3, i_{12}) = \begin{cases} v_{i_3}(\xi_3) & \text{if } 0 \leq i_{12} < N_{12}^1, \\ \chi'_{i_3}(\xi_3) & \text{if } N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2. \end{cases}$$

An auxiliary function sequence (similar to that in (2.9) for the L_2 case) can be introduced for the computation of this term.

$$\begin{aligned} \bar{\mathcal{G}}_{i_{12}j_{12}}^{\text{divA}}(\xi_3) &:= \int_{\mathcal{T}} w_{i_{12}}(\xi_1, \xi_2) w_{j_{12}}(\xi_1, \xi_2) |J(\xi_1, \xi_2, \xi_3)|^{-1} d^2(\xi_1, \xi_2) \\ \bar{\mathcal{G}}_{i_{12}j_{12}i_3j_3}^{\text{div}} &:= \int_I x_{i_3}(\xi_3, i_{12}) \bar{\mathcal{G}}_{i_{12}j_{12}}^{\text{divA}}(\xi_3) x_{j_3}(\xi_3, j_{12}) d\xi_3 \end{aligned} \quad (2.23)$$

The $H(\text{div})$ Gram matrix can finally be computed by summing the contribution from each term as:

$$G_{IJ}^{\text{div}} = \mathcal{G}_{i_{12}j_{12}i_3j_3}^{\text{div}} + \bar{\mathcal{G}}_{i_{12}j_{12}i_3j_3}^{\text{div}}. \quad (2.24)$$

2.7. Space $H(\text{curl})$

We conclude this section by presenting a sum factorization for $H(\text{curl})$. However, due to the similarities between this and other spaces

we forgo explicit definitions of the auxiliary sequences. The definition of prism shape function families for polynomial subspace $\hat{Q}^p$ again follows from (2.7):

$$\hat{Q}^p = \underbrace{\hat{Q}_{\mathcal{T}}^{p_{12}} \otimes \hat{W}_{\mathcal{I}}^{p_3}}_{\text{Family 1}} \times \underbrace{\hat{W}_{\mathcal{T}}^{p_{12}} \otimes \hat{Y}_{\mathcal{I}}^{p_3}}_{\text{Family 2}}.$$

This two family structure closely resembles that of $\hat{V}^p$, thus we use the same indexing structure as before.

Let the basis $\{\hat{\psi}_I\}_{I=0}^{\dim \hat{Q}^p - 1}$ of prismatic shape functions spanning $\hat{Q}^p$ be partitioned into two families. As before, we allow i_{12} to take non-negative values up to $\dim \hat{Q}_{\mathcal{T}}^{p_{12}} + \dim W_{\mathcal{T}}^{p_{12}}$, uniquely enumerating 2D

simplicial components of the basis $\{\hat{\psi}_I\}$. The index i_{12} can then be used to identify the appropriate space for univariate shape functions ($W_{\mathcal{I}}^{p_3}$ or $Y_{\mathcal{I}}^{p_3}$) and the corresponding range of index i_3 . Gram index I can again be uniquely defined given values of i_{12} , i_3 . Table 2 defines the shape functions and indexing corresponding to each family.

The $H(\text{curl})$ Gram matrix can then be calculated for each ordered pair (ψ_I, ψ_J) as:

As can be seen in Table 2, both $\hat{\psi}_I$ and $\widehat{\text{curl}} \hat{\psi}_I$ are vector quantities. Thus, both integrals in (2.25) can be factored through Fubini's theorem by introducing array factors similar to (2.20) and (2.21). The auxiliary function sequences for the sum factorization of both integrals are then sufficiently similar to (2.22) to neglect explicit definition here.

3. Results

We begin our exposition of numerical results by reporting computational times both for conventional and sum-factorized Gram matrix assembly in each of the previously presented energy spaces. Next, the construction of DPG matrices G (Gram), B (stiffness), and l (load) for an ultraweak formulation of a Maxwell problem employing a scaled adjoint graph norm is considered. In all cases order $\mathcal{O}(p^7)$ complexity is observed.

We conclude this section by considering a partial tensorization of hexahedral-type elements (as the tensor product of a 2D square and 1D interval), showing that the corresponding sum-factorization routine achieves order $\mathcal{O}(p^7)$ complexity however with a slightly higher computational expense compared to full tensorization. Such a partially tensorized formulation may be desirable for a number of reasons including improved ease and brevity of implementation, as well as increased

Table 2
Definition of two families of prismatic shape functions for $\hat{Q}^p$

Family 1	Family 2
$\hat{\psi}_I = \begin{pmatrix} E_{i_{12},1}(\xi_1, \xi_2) \chi_{i_3}(\xi_3) \\ E_{i_{12},2}(\xi_1, \xi_2) \chi_{i_3}(\xi_3) \\ 0 \end{pmatrix}$	$\hat{\psi}_I = \begin{pmatrix} 0 \\ 0 \\ u_{i_{12}}(\xi_1, \xi_2) v_{i_3}(\xi_3) \end{pmatrix}$
$\widehat{\text{curl}} \hat{\psi}_I = \begin{pmatrix} -E_{i_{12},2}(\xi_1, \xi_2) \chi'_{i_3}(\xi_3) \\ E_{i_{12},1}(\xi_1, \xi_2) \chi'_{i_3}(\xi_3) \\ \text{curl}(E_{i_{12}}(\xi_1, \xi_2)) \chi_{i_3}(\xi_3) \end{pmatrix}$	$\widehat{\text{curl}} \hat{\psi}_I = \begin{pmatrix} \partial_y u_{i_{12}}(\xi_1, \xi_2) v_{i_3}(\xi_3) \\ -\partial_x u_{i_{12}}(\xi_1, \xi_2) v_{i_3}(\xi_3) \\ 0 \end{pmatrix}$
where $E_{i_{12}} \in \hat{Q}_T^{p_{12}}, N_{12}^1 = (p_{12} + 2)p_{12}, 0 \leq i_{12} < N_{12}^1$;	where $u_{i_{12}} \in \hat{W}_T^{p_{12}}, N_{12}^2 = \frac{1}{2}(p_{12} + 2)(p_{12} + 1), N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2$
and $\chi_{i_3} \in \hat{W}_I^{p_3}, N_3^1 = p_3 + 1, 0 \leq i_3 < N_3^1$;	and $v_{i_3} \in \hat{V}_I^{p_3}, N_3^2 = p_3, 0 \leq i_3 < N_3^2$;

Table 3

Computational times (seconds) for conventional and sum factorized Gram matrix G assembly for a prismatic element in each exact sequence energy space. The observed order was calculated using only the three highest order elements $p_r = 6, 7, 8$ to better capture asymptotic behavior.

p_r	L^2		H^1	
	Conventional	Sum Factorized	Conventional	Sum Factorized
2	4.1×10^{-5}	3.8×10^{-5}	7.7×10^{-5}	5.1×10^{-5}
3	1.4×10^{-4}	1.3×10^{-4}	3.4×10^{-4}	1.8×10^{-4}
4	3.8×10^{-4}	3.2×10^{-4}	1.3×10^{-3}	5.4×10^{-4}
5	1.1×10^{-3}	7.8×10^{-4}	5.2×10^{-3}	1.5×10^{-3}
6	3.1×10^{-3}	1.6×10^{-3}	2.1×10^{-2}	3.2×10^{-3}
7	1.2×10^{-2}	3.1×10^{-3}	6.0×10^{-2}	9.0×10^{-3}
8	3.5×10^{-2}	6.5×10^{-3}	2.2×10^{-1}	1.9×10^{-2}
Observed Order	8.3	4.9	8.1	6.1

p_r	$H \text{ div}$		$H \text{ curl}$	
	Conventional	Sum Factorized	Conventional	Sum Factorized
2	9.9×10^{-5}	6.3×10^{-5}	2.2×10^{-4}	1.7×10^{-4}
3	5.7×10^{-4}	2.7×10^{-4}	1.9×10^{-3}	9.3×10^{-4}
4	3.6×10^{-3}	1.1×10^{-3}	1.3×10^{-2}	5.1×10^{-3}
5	2.5×10^{-2}	4.4×10^{-3}	6.8×10^{-2}	1.9×10^{-2}
6	1.0×10^{-1}	1.3×10^{-2}	2.7×10^{-1}	5.5×10^{-2}
7	3.7×10^{-1}	3.2×10^{-2}	8.3×10^{-1}	1.4×10^{-1}
8	1.3×10^0	7.3×10^{-2}	2.8×10^0	3.2×10^{-1}
Observed Order	8.6	6.3	8.1	6.1

implementational compatibility in applications where both hexahedral and prismatic elements are used.

In each of the following examples, both sum-factorized and conventional element assembly routines were implemented. Sum factorization routines were then verified by direct comparison of matrices with those produced by standard construction routines. In every case, the resulting matrices were verified to be identical within machine precision. All experiments were performed 50 times to reduce statistical variation; only averages are reported here.

3.1. Gram matrix assembly in various energy spaces

Assembly of the Gram Matrix was performed for each of the exact sequence energy spaces (in the associated norm). Computational times for assembly on a prismatic element of various enriched orders (p_r) are presented in Table 3—revealing a roughly $10 \times$ computational advantage of sum factorization in the case of enriched order $p_r = 8$ in each energy space. The observed order reported in Table 3 for each energy space was calculated using regression on the three highest order elements $p_r = 6, 7, 8$. Note in particular that the observed orders both for conventional and for sum factorized assembly appear slightly less than theory would suggest; the reason behind this aberration will become apparent in further discussion.

Fig. 1 provides a graphical representation of the computational data in Table 3 with additional reference lines corresponding to the expected $\mathcal{O}(p^9)$ and $\mathcal{O}(p^7)$ rates. Consideration of Fig. 1 reveals that a pre-asymptotic regime for low-orders p_r is to blame for the seemingly deficient observed orders reported in Table 3. Especially in the case of L^2 and H^1 energy spaces, the pre-asymptotic region is observed to persist well into the high-order regime. Note however that for experiments in which pre-asymptotic behavior is especially apparent, computational times are small, typically on the order of milliseconds. The relatively small relatively small computational times as well as presence of pre-asymptotic behavior in both conventional and sum-factorized routines, suggest that this pre-asymptotic behavior is due to computational and memory overhead and could be implementation dependent. To minimize computational overhead, arrays for Gram matrix G and auxiliary matrices $\mathcal{G}^A$ were dynamically allocated in contiguous memory; however only minor improvements in pre-asymptotic behavior were observed.

Despite the presence of a pre-asymptotic region, sum factorization was observed to reduce over-all computational cost in each energy space—demonstrating improved assembly cost for all order elements.

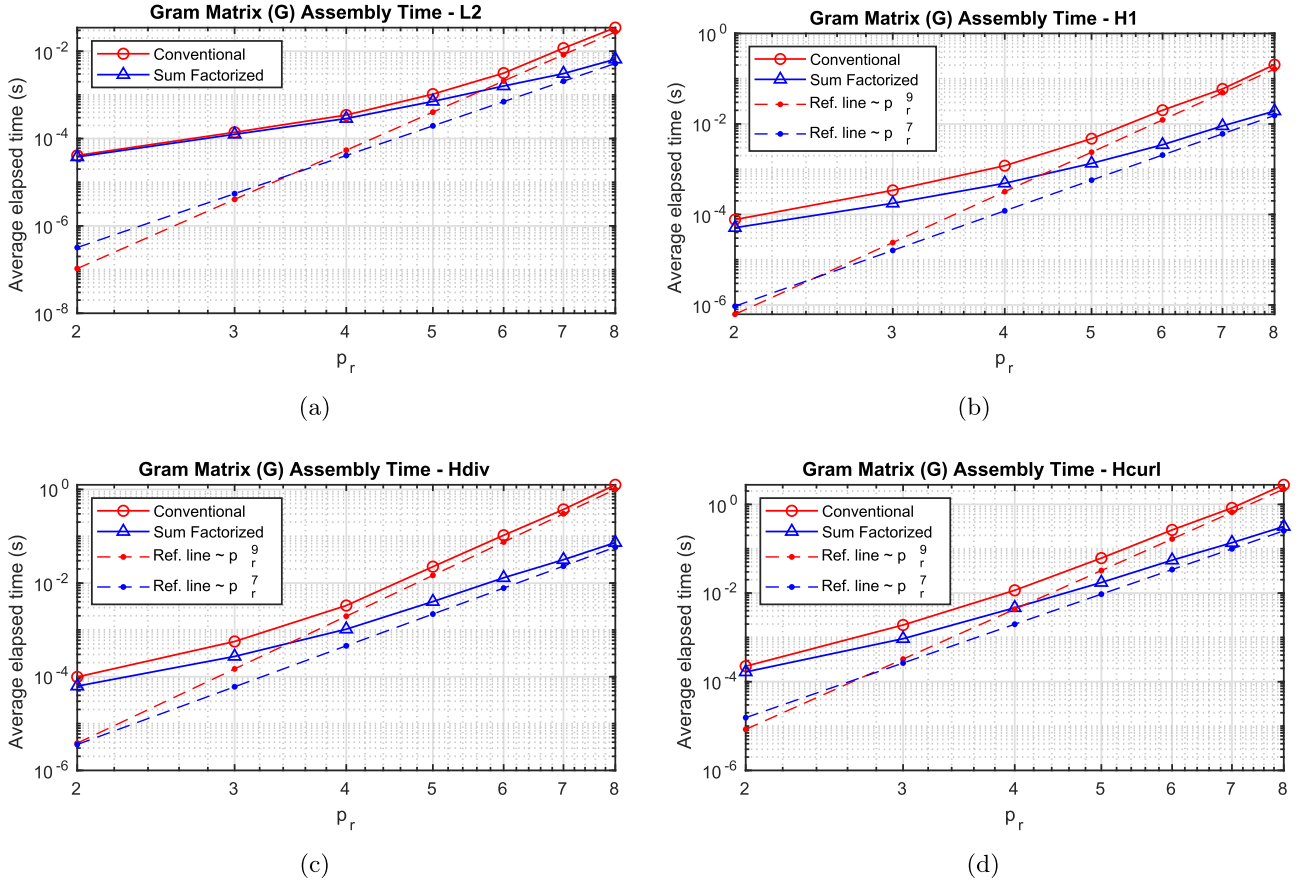


Fig. 1. Computational times for conventional and sum factorized Gram matrix G assembly for a prismatic element in each exact sequence energy space.

3.2. Assembly of DPG matrices for ultraweak Maxwell problem

To further illustrate the utility of sum factorization for the construction of DPG systems, we consider as a model problem the ultraweak variational form of Maxwell's equation. This problem and its DPG setting are discussed in depth in Refs. [1,11], but will be outlined here for completeness.

3.2.1. Problem definition

Consider the time-harmonic Maxwell system (of positive frequency $\omega > 0$) defined on an open bounded and connected domain $\Omega \subset \mathbb{R}^3$ given by:

$$\begin{cases} \text{curl } E + i\omega\mu H = 0 & \text{in } \Omega \\ \text{curl } H - i\omega\epsilon E = \mathcal{J}^{imp} & \text{in } \Omega \\ n \times E = n \times E_0 & \text{on } \Gamma_E \\ n \times H = n \times H_0 & \text{on } \Gamma_H \end{cases} \quad (3.1)$$

where the functions $E, H, \mathcal{J}^{imp} : \Omega \rightarrow \mathbb{C}$ represent electric field, magnetic field, and imposed current respectively, and Γ_E, Γ_H coincide with the disjoint portions of boundary $\partial\Omega$ on which electric and magnetic boundary conditions are imposed. Parameters μ, ϵ represent the electromagnetic properties of the domain and are assumed to be positive and element-wise constant on a mesh Ω_h . We denote by Γ_h the skeleton of mesh Ω_h .

Ultraweak variational forms (as defined in Ref. [1]) are obtained by expressing a system in first order form, then weakening each first-order equation by introducing a test function and integrating by parts. Such a formulation passes all differential operators—and corresponding regularity—to the test space. In the case of the first order Maxwell system (3.1), the ultraweak formulation is obtained by multiplying the

first and second lines by test functions $F, G \in H(\text{curl}, \Omega)$ respectively, integrating by parts, identifying the new unknowns—traces defined on mesh skeleton Γ_h , and incorporating boundary conditions on Γ_E and Γ_H . The following system is obtained:

$$\begin{cases} E, H \in (L^2(\Omega))^3, \quad \hat{E}_t, \hat{H}_t \in H^{-1/2}(\text{curl}, \Gamma_h), \\ (E, \text{curl } F) - \langle n \times \hat{E}_t, F \rangle_{\Gamma_h} + i\omega(\mu H, F) = 0 & F \in H(\text{curl}, \Omega_h), \\ (H, \text{curl } G) - \langle n \times \hat{H}_t, G \rangle_{\Gamma_h} - i\omega(\epsilon E, G) = (\mathcal{J}^{imp}, G) & G \in H(\text{curl}, \Omega_h), \\ \hat{E}_t = E_{0,t} & \text{on } \Gamma_E, \\ \hat{H}_t = H_{0,t} & \text{on } \Gamma_H, \end{cases} \quad (3.2)$$

where $(\cdot, \cdot)$ denotes the standard L^2 product, and $\langle \cdot, \cdot \rangle_{\Gamma_h}$ denotes the duality pairing between trace spaces $H^{-1/2}(\text{div}, \Gamma_h)$ and $H^{-1/2}(\text{curl}, \Gamma_h)$. The additional unknown functions $\hat{E}_t, \hat{H}_t$ denote tangential traces defined on the mesh skeleton Γ_h that arise through the use of the discontinuous test space $H(\text{curl}, \Omega_h)$ with no additional assumptions (i.e. electing to test on the boundary). System (3.2) can be expressed in abstract variational form by introducing bilinear functional

$$b((E, H, \hat{E}_t, \hat{H}_t), (F, G)) = b((E, H), (F, G)) + \tilde{b}((\hat{E}_t, \hat{H}_t), (F, G)) \quad (3.3)$$

where,

$$b((E, H), (F, G)) = (E, \text{curl } F) + (H, \text{curl } G) + i\omega(\mu H, F) - i\omega(\epsilon E, G),$$

$$\tilde{b}((\hat{E}_t, \hat{H}_t), (F, G)) = -\langle n \times \hat{E}_t, F \rangle_{\Gamma_h} - \langle n \times \hat{H}_t, G \rangle_{\Gamma_h},$$

and linear functional

$$\ell((F, G)) = (\mathcal{J}^{imp}, G). \quad (3.4)$$

Table 4

Computational times for assembly of the Gram matrix G alone and with additional DPG stiffness matrix B and load l for the ultraweak Maxwell problem on a prismatic element.

p_0	Δp	p_r	G Assembly Time (s)		G, B, l Assembly Time (s)	
			Conventional	Sum Factorized	Conventional	Sum Factorized
2	0	2	1.0×10^{-3}	5.7×10^{-4}	1.1×10^{-3}	6.6×10^{-4}
2	1	3	1.3×10^{-2}	4.6×10^{-3}	1.4×10^{-2}	4.7×10^{-3}
3	1	4	7.4×10^{-2}	1.3×10^{-2}	8.2×10^{-2}	1.4×10^{-2}
4	1	5	4.1×10^{-1}	4.8×10^{-2}	4.4×10^{-1}	5.2×10^{-2}
5	1	6	1.8×10^0	1.4×10^{-1}	2.0×10^0	1.5×10^{-1}
6	1	7	8.3×10^0	3.9×10^{-1}	1.2×10^1	4.7×10^{-1}
6	2	8	3.8×10^1	1.1×10^0	4.5×10^1	1.3×10^0
6	3	9	1.1×10^2	2.4×10^0	1.1×10^2	2.5×10^0
Observed Order			9.2	6.9	9.1	7.0

To simplify notation, we define group variables $u = (E, H)$; $\hat{u} = (\hat{E}_t, \hat{H}_t)$; and $v = (G, F)$ with corresponding spaces $\mathcal{U} = (L^2(\Omega))^6$; $\hat{\mathcal{U}} = (H^{-1/2}(\text{curl}, \Gamma_h))^2$; and $\mathcal{V} = (H(\text{curl}, \Omega_h))^2$.

Variational problem (3.2) can be cast as a mixed problem by introducing the error representation function ψ (detailed in Refs. [1,11]) as follows: find $\psi \in \mathcal{V}^r$, $u^h \in \mathcal{U}^h$, $\tilde{u}^h \in \hat{\mathcal{U}}^h$ such that

$$\begin{cases} (\psi, v)_{\mathcal{V}^r(\Omega_h)} + b(u^h, v) + \tilde{b}(\tilde{u}^h, v) = \ell(v) & \forall v \in \mathcal{V}^r(\Omega_h) \\ b(\delta u, \psi) = 0 & \forall \delta u \in \mathcal{U}^h \\ \tilde{b}(\delta \tilde{u}, \psi) = 0 & \forall \delta \tilde{u} \in \hat{\mathcal{U}}^h(\Gamma_h) \end{cases} \quad (3.5)$$

where $(\cdot, \cdot)_{\mathcal{V}^r(\Omega_h)}$ denotes the test inner product which, in the context of DPG, is assumed to be defined a priori—with a particular choice of test norm defining a particular DPG method. Here we employ the scaled adjoint test norm as described in Ref. [1]. Finally, defining a discrete trial subspace allows problem (3.5) to be formulated in discrete matrix form as

$$\begin{cases} Gs + Bu + \tilde{B}w = l \\ B^T s = 0 \\ \tilde{B}^T s = 0, \end{cases} \quad (3.6)$$

where s , u , and w represent degrees-of-freedom corresponding to ψ , u^h , and $\tilde{u}^h$ respectively. Matrix $\tilde{B}$ in (3.6) is composed of only trace terms and its assembly requires integration only over 2D faces—evaluation of which are computationally insignificant compared to the overall cost of assembly—and can be handled by conventional assembly methods. The remaining Gram matrix G , stiffness matrix B , and load vector l

in (3.6) involve only volume integrals and are amenable to the sum factorization techniques outlined previously.

The dimensionality of system (3.6) can be reduced by statically condensing variable s to obtain a system of the form

$$\begin{pmatrix} B^T G^{-1} B & B^T G^{-1} \tilde{B} \\ \tilde{B}^T G^{-1} B & \tilde{B}^T G^{-1} \tilde{B} \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} B^T G^{-1} l \\ \tilde{B}^T G^{-1} l \end{pmatrix}. \quad (3.7)$$

The block diagonal structure of G achieved by breaking test spaces in the practical DPG method (see Ref. [1]) allows this static condensation to be performed on an element level. This is efficiently achieved by performing a Cholesky factorization on G , then using back-substitution to obtain $G^{-1}B$, $G^{-1}\tilde{B}$, and $G^{-1}l$.

3.2.2. Computational results

Table 4 reports assembly times for the Gram matrix G , and for the full DPG system G , B , and l . Comparing assembly times for G to those for G , B , and l , it can be verified that the assembly of the Gram matrix G incurs the greatest computational expense in the construction of DPG systems—a result that reiterates the need for specialized Gram matrix assembly routines considered in this work. Indeed, in the case of enriched order $p_r = 9$ notice that the conventional assembly time of 110 s is reduced to a mere 2.4 s for sum factorized assembly. Additionally, it can be observed that in the case of a highly enriched test space ($\Delta p = 3$) the additional cost for assembling B and l becomes relatively negligible, requiring roughly 4% of overall cost for both conventional and sum factorized routines.

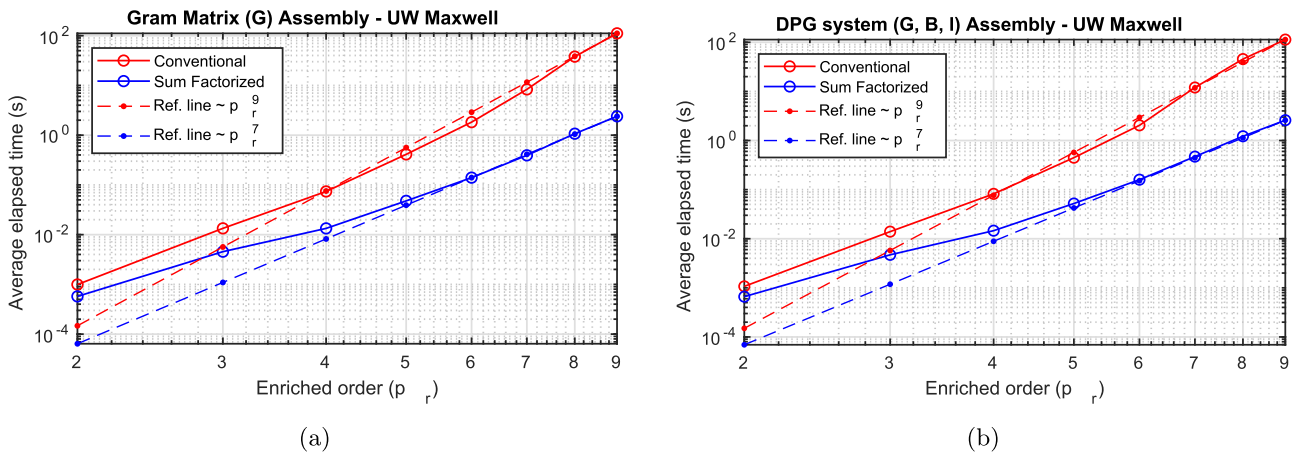


Fig. 2. Computational results for assembly of (a) Gram matrix G and (b) DPG system (neglecting trace terms) consisting of Gram matrix G , stiffness matrix B , and load l for the ultraweak Maxwell problem on a prismatic element.

Table 5

Computational times for construction of the ultraweak Maxwell Gram matrix using conventional, partial sum factorization, and full sum factorization techniques.

p_r	G Assembly Time (s)		
	Conventional	Partial Sum Factorization	Full Sum Factorization
2	2.9×10^{-3}	1.5×10^{-3}	6.1×10^{-4}
3	4.0×10^{-2}	8.9×10^{-3}	2.5×10^{-3}
4	3.0×10^{-1}	4.1×10^{-2}	1.0×10^{-2}
5	1.7×10^0	1.6×10^{-1}	3.4×10^{-2}
6	7.9×10^0	5.5×10^{-1}	1.0×10^{-1}
7	3.5×10^1	1.6×10^0	2.7×10^{-1}
8	1.2×10^2	4.0×10^0	6.4×10^{-1}
Observed Order	9.2	6.9	6.7

The observed order reported in Table 4 was calculated using regression on the three highest enriched order elements $p_r = 7, 8, 9$ and verifies the respective $\mathcal{O}(p^9)$ and $\mathcal{O}(p^7)$ complexity for conventional and sum factorized assembly. Graphical representation of the data as depicted in Fig. 2 reveals that expected asymptotic rates are reached for relatively low polynomial orders p_r .

3.3. Partial tensorization of hexahedral elements

To conclude our exposition of results we briefly consider a partial tensorization of the hexahedral elements, based on the representation of the hexahedra as a tensor product of a 2D square domain and 1D interval. Such a construction produces auxiliary function sequences, shape function families, and computational loops with structures similar to those for the prism. Indeed, the primary benefit of this partially tensorized representation is that it allows for a symmetric implementation of prismatic and hexahedral elements. In the case of the authors' code base, this allowed a single sum factorization routine to handle assembly of both element types. As an added benefit, the partially tensorized representation reduces both the length and complexity of element assembly routines by eliminating the assembly of secondary auxiliary matrices (denoted $\mathcal{G}^B$ in Ref. [10]). Note however that this representation allows for polynomial anisotropy in only a single direction (the 2D square is assumed to be of uniform polynomial order) and therefore may not be suitable for routines requiring fully anisotropic polynomial refinements.

To provide a direct comparison of partially and fully tensorized sum factorization on hexahedral elements, both Gram matrix assembly routines were implemented for the ultraweak Maxwell problem considered previously. The results of this experiment are reported in Table 5 and depicted graphically in Fig. 3.

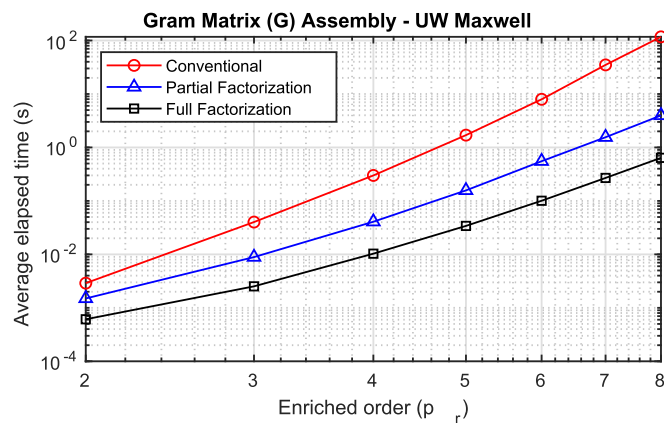


Fig. 3. Computational times for construction of the ultraweak Maxwell Gram matrix using conventional, partial sum factorization, and full sum factorization techniques.

Consideration of Table 5 reveals that both partial sum factorization and full sum factorization routines achieve the expected $\mathcal{O}(p^7)$ complexity. However, it can be seen both in Table 5 and in Fig. 3 that the partial sum factorization requires a roughly constant multiple of four to six times greater computational cost compared to full sum factorization. Despite the increased cost, the partial sum factorization significantly reduced assembly time compared to the conventional procedure—achieving a $10 \times$ speed-up in the case of modest enriched order $p_r = 5$ and a $30 \times$ speed-up in the case of enriched order $p_r = 8$. While the increased expense of the partially tensorized representation is certainly non-negligible, in applications where both prismatic and hexahedral elements are used sum factorization routines for prismatic elements can be rather trivially extended to support hexahedral elements. Additionally, the reduced length and complexity of a unified routine for treatment of hexahedral and prismatic elements may further justify the additional computational cost incurred for partial sum factorization.

4. Conclusions

Sum factorization routines for fast assembly of Gram matrices in the exact sequence energy spaces H^1 , $H(\text{curl})$, $H(\text{div})$, and L^2 were proposed based on the construction of prismatic shape functions as tensor-products of 2D simplex and 1D interval shape functions. The proposed algorithms for the partial tensorization of prismatic elements achieve the same $\mathcal{O}(p^7)$ complexity as the full tensorization of the hexahedra (as a product of three 1D intervals) proposed in Ref. [10]. This somewhat unexpected result is achieved since the final matrix maintains the same $\mathcal{O}(p^7)$ complexity but the complexity of auxiliary matrix assembly is increased from $\mathcal{O}(p^6)$ in the case of the $\mathcal{O}(p^6)$ in the case of the fully tensorized hexahedra to $\mathcal{O}(p^7)$ in the case of the prism. The proposed algorithms were verified to achieve the expected $\mathcal{O}(p^7)$ complexity in each energy space—a significant reduction over conventional $\mathcal{O}(p^9)$ assembly routines.

To further illustrate the efficiency of sum factorization routines, the ultraweak formulation of a Maxwell problem was considered. The sum factorized construction of DPG matrices on a prismatic element significantly reduced computational cost in the case of both low-order and high-order elements. Additionally, a partial factorization for hexahedral elements (as a product of 2D square and 1D interval) was proposed to mirror the structure of prismatic elements. Such a formulation allows for a symmetric treatment of prismatic and hexahedral elements—enabling the unification of element assembly routines for prismatic and hexahedral elements—but was observed to incur a roughly constant four to six times penalty in computational performance. Despite this significant penalty, the expected $\mathcal{O}(p^7)$ complexity was observed and significant computational savings were observed for all polynomial orders compared to conventional assembly routines—achieving a $30 \times$ reduced computational expense in the case of enriched order $p_r = 8$ elements.

Recently, in the context of modern manycore architectures, especially GPU accelerators, parallel assembly of element matrices has become a focus of study. While the sum factorization routines presented here were developed independent of parallel considerations, these routines are in fact amenable to parallelization. For example, evaluation of auxiliary matrices at 1D quadrature points could readily be distributed. It should be noted however that the additional memory required to store the auxiliary arrays for sum factorization—while relatively small—may inhibit efficient GPU acceleration. Investigation of the feasibility and efficacy of such techniques would be beneficial and may further reduce the cost of DPG system assembly.

Sum factorization routines for the construction of DPG systems have thus far been presented only for hexahedral and prismatic element types since their structure is amenable to a tensor product representation. While shape functions on the remaining tetrahedral and pyramid element types do not possess a natural tensor structure, a tensor structure may be imparted through use of Duffy transformations as noted in Ref. [6] and outlined in Refs. [7,14]. Sum factorized construction of DPG systems on the remaining element types may then be achieved by exploiting the resulting tensor structure. Such an extension of sum factorization to include all finite element types would enable considerable computational savings on more general geometries—especially in parallel element assembly routines where use of conventional assembly on a subset of elements produces a significant load imbalance. The extension of sum factorization to include all element types in each exact sequence energy space is left to future work.

Acknowledgments

This work has been supported by AFRL grant No. FA9550-17-1-0090.

References

- [1] C. Carstensen, L. Demkowicz, J. Gopalakrishnan, Breaking spaces and forms for the DPG method and applications including Maxwell equations, *Comput. Math. Appl.* 72 (3) (2016) 494–522.
- [2] L. Demkowicz, Various Variational Formulations and Closed Range Theorem, The University of Texas at Austin, 2015. ICES Report 15-03.
- [3] L. Demkowicz, J. Gopalakrishnan, Discontinuous Petrov-Galerkin (DPG) Method. ICES Report 15-20, The University of Texas at Austin, 2015.
- [4] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszyski, W. Rachowicz, A. Zdunek, Computing with Hp Finite Elements. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications, Chapman & Hall/CRC, New York, 2007.
- [5] F. Fuentes, B. Keith, L. Demkowicz, S. Nagaraj, Orientation embedded high order shape functions for the exact sequence elements of all shapes, *Comput. Math. Appl.* 70 (4) (2015) 353–458.
- [6] P. Gatto, L. Demkowicz, Construction of H1-conforming hierarchical shape functions for elements of all shapes and transfinite interpolation, *Finite Elem. Anal. Des.* 46 (6) (2010) 474–486.
- [7] G. Karniadakis, S.J. Sherwin, *Spectral/hp Element Methods for CFD*. Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 1999.
- [8] B. Keith, S. Petrides, F. Fuentes, L. Demkowicz, Discrete least-squares finite element methods, *Comput. Methods Appl. Mech. Eng.* 327 (2017) 226–255.
- [9] M. Kraus, K. Kormann, P.J. Morrison, E. Sonnendrcker, GEMPIC: geometric electromagnetic particle-in-cell methods, *J. Plasma Phys.* 83 (4) (2017).
- [10] J. Mora Paz, L. Demkowicz, Fast integration of DPG matrices based on sum factorization for all the energy spaces, *Comput. Methods Appl. Math.* 19 (3) (2019) 523–555.
- [11] S. Nagaraj, J. Grosek, S. Petrides, L. Demkowicz, J. Mora, A 3D DPG Maxwell approach to nonlinear Raman gain in fiber laser amplifiers, *J. Comp. Physiol.* 2 (2019) 100002.
- [12] S. Nagaraj, S. Petrides, L. Demkowicz, Construction of a DPG Fortin Operator, ICES Report, vols. 1522, The University of Texas at Austin, 2015.
- [13] S. Petrides, Adaptive Multilevel Solvers for the Discontinuous PetrovGalerkin Method with an Emphasis on High-Frequency Wave Propagation Problems, The University of Texas at Austin, Austin, Texas, 2019. PhD thesis.
- [14] S. Zaglmayr, High Order Finite Element Methods for Electromagnetic Field Computation, Johannes Kepler Universitt Linz, Linz, 2006. PhD thesis.