Oden Institute REPORT 19-15

August 2019

Fast integration of DPG matrices based on tensorization (Part II—Prismatic elements)

by

Jacob Badger, Stefan Henneking, and Leszek Demkowicz

Oden Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Jacob Badger, Stefan Henneking, and Leszek Demkowicz, "Fast integration of DPG matrices based on tensorization (Part II—Prismatic elements)," Oden Institute REPORT 19-15, Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, August 2019.

Fast integration of DPG matrices based on tensorization (Part II—Prismatic elements)

Jacob Badger*, Stefan Henneking, and 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 [6] 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 [6]. 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

*Corresponding author. E-mail: jcbadger@utexas.edu

presented in [6] 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.

This work follows closely the work of its prequel [6] (and earlier work by Kurtz [2]). The reader is directed there (and the contained references) for a brief review of shape functions, the construction of DPG systems, and sum factorization.

The present work builds on that of its prequel, presenting only results and details which are sufficiently different for the prismatic element than for the hexahedral element to merit additional discussion. This paper is organized 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 a future work.

2 Sum Factorization

Similar to its predecessor, 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. The Piola transforms (pullback maps) presented in [6] allow for the transfer of all element domains to the master domain. Since such maps are independent of element type or tensorized structure, we will forgo their definition and consider only spaces (and integrals) defined on the master domain $\hat{\mathcal{K}} = \mathcal{T} \otimes \mathcal{I}$ 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 $\mathbb{R}$. 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.

2.1 Exact sequences

The infinite-dimensional exact sequences in one, two, and three-dimensions were defined previously in [6] and are given here for reference:

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

where the ∇_{vts} and ∇_{stv} denote the 2D vector-to-scalar and scalar-to-vector curl operators respectively, defined by $\nabla_{vts} = \partial_1(\cdot)_2 - \partial_2(\cdot)_1$ and $\nabla_{stv} = (\partial_2, -\partial_1)$.

2.2 Tensor-product prismatic finite element shape functions

For each of the infinite dimensional exact sequences in (2.1), we define polynomial subspaces matching 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:

$$\begin{array}{c}
\hat{W}^p_{\mathcal{T}} = \mathcal{P}^p(\mathcal{I}) \subsetneq H^1(\mathcal{I}) \\
\Big\downarrow{\scriptstyle \partial} \\
\hat{Y}^p_{\mathcal{I}} = \mathcal{P}^{p-1}(\mathcal{I}) \subsetneq L^2(\mathcal{I})
\end{array}$$

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 [3] by:

$$\begin{array}{ccc}
\hat{W}^p_{\mathcal{T}} = \mathcal{P}^p(\mathcal{T}) \subsetneq H^1(\mathcal{T}) & \qquad & \hat{W}^p_{\mathcal{T}} = \mathcal{P}^p(\mathcal{T}) \subsetneq H^1(\mathcal{T}) \\
\Big\downarrow{\scriptstyle \nabla=(\partial_1,\partial_2)} & & \Big\downarrow{\scriptstyle \nabla_{stv}=(\partial_2,-\partial_1)} \\
\hat{Q}^p_{\mathcal{T}} = \mathcal{N}^p(\mathcal{T}) \subsetneq H(\mathrm{curl},\mathcal{T}) & & \hat{V}^p_{\mathcal{T}} = \mathcal{RT}^p(\mathcal{T}) \subsetneq H(\mathrm{div},\mathcal{T}) \\
\Big\downarrow{\scriptstyle \nabla_{vts}=\partial_1(\cdot)_2-\partial_2(\cdot)_1} & & \Big\downarrow{\scriptstyle \mathrm{div}=\partial_1(\cdot)_1+\partial_2(\cdot)_2} \\
\hat{Y}^p_{\mathcal{T}} = \mathcal{P}^{p-1}(\mathcal{T}) \subsetneq L^2(\mathcal{T}) & & \hat{Y}^p_{\mathcal{T}} = \mathcal{P}^{p-1}(\mathcal{T}) \subsetneq L^2(\mathcal{T}).
\end{array}$$

where $\mathcal{P}^p(\mathcal{T})$ is the space of bivariate polynomials 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:

$$\begin{aligned}
\mathcal{N}^p(\hat{\mathcal{T}}) &= \mathcal{P}^{p-1} \otimes \left\{ E \in \left(\widetilde{\mathcal{P}}^p\right)^N : x \cdot E(x) = 0 \text{ for all } x \in \mathbb{R}^N \right\}, \\
\mathcal{RT}^p(\hat{\mathcal{T}}) &= \mathcal{P}^{p-1} \otimes \left\{ V \in \left(\widetilde{\mathcal{P}}^p\right)^N : V(x) = \varphi(x)x \text{ with } \varphi \in \widetilde{\mathcal{P}}^{p-1} \text{ for all } x \in \mathbb{R}^N \right\},
\end{aligned}$$

where $\widetilde{\mathcal{P}}^p$ 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:

$$\begin{array}{l}
\hat{W}^p = \hat{W}^{p_{12}}_{\mathcal{T}} \otimes \hat{W}^{p_3}_{\mathcal{I}} \\
\qquad \Big\downarrow{\scriptstyle \nabla=(\partial_1,\partial_2,\partial_3)} \\
\hat{Q}^p = \hat{Q}^{p_{12}}_{\mathcal{T}} \otimes \hat{W}^{p_3}_{\mathcal{I}} \ \times \ \hat{W}^{p_{12}}_{\mathcal{T}} \otimes \hat{Y}^{p_3}_{\mathcal{I}} \\
\qquad \Big\downarrow{\scriptstyle \mathrm{curl}=(\partial_2(\cdot)_3-\partial_3(\cdot)_2,\partial_3(\cdot)_1-\partial_1(\cdot)_3,\partial_1(\cdot)_2-\partial_2(\cdot)_1)} \\
\hat{V}^p = \hat{Y}^{p_{12}}_{\mathcal{T}} \otimes \hat{W}^{p_3}_{\mathcal{I}} \ \times \ \hat{V}^{p_{12}}_{\mathcal{T}} \otimes \hat{Y}^{p_3}_{\mathcal{I}} \\
\qquad \Big\downarrow{\scriptstyle \mathrm{div}=\partial_1(\cdot)_1+\partial_2(\cdot)_2+\partial_3(\cdot)_3} \\
\hat{Y}^p = \hat{Y}^{p_{12}}_{\mathcal{T}} \otimes \hat{Y}^{p_3}_{\mathcal{I}}.
\end{array} \tag{2.2}$$

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 (2.2) 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 [6], and the construction for each space is presented in the order defined therein—deviating from the exact sequence order in favor of simplicity.

2.3 Space L^2

Let $\{\upsilon\}_{I=0}^{\dim Y^p-1}$ be a basis for Y^p, where $\dim Y^p = \frac{1}{2}(p_{12}+1)p_{12}p_3$. Elements of basis $\{\upsilon_I\}$ in Y^p are represented in the tensorized form of (2.2) as $\upsilon_I = \upsilon_{i_{12}}\nu_{i_3}$ where $\upsilon_{i_{12}} \in \hat{Y}_{\mathcal{T}}^{p_{12}}$ with $i_{12} = 0, ..., \frac{1}{2}(p_{12}+1)p_{12}-1$; where $\nu_{i_3} \in \hat{Y}_{\mathcal{I}}^{p_3}$ with $i_3 = 0, ..., p_3-1$; and where I is some unique integer identifier dependant on i_{12} and i_3 such that $0 \le I < \dim Y^p$. The L^2 Gram matrix is then constructed for ordered pairs of basis elements (υ_I, υ_J) as:

$$\begin{aligned}
\mathsf{G}_{IJ} &= (\upsilon_I, \upsilon_J)_{\mathcal{K}} \\
&= \int\limits_{\hat{\mathcal{K}}} \hat{\upsilon}_I(\boldsymbol{\xi})\hat{\upsilon}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|^{-1}d^3\boldsymbol{\xi} \\
&= \int\limits_{\mathcal{I}}\int\limits_{\mathcal{T}} \hat{\upsilon}_I(\xi_1,\xi_2,\xi_3)\hat{\upsilon}_J(\xi_1,\xi_2,\xi_3)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|^{-1}d^2(\xi_1,\xi_2)d\xi_3 \\
&= \int\limits_{\mathcal{I}} \nu_{i_3}(\xi_3)\nu_{j_3}(\xi_3)\left\{\int\limits_{\mathcal{T}} \upsilon_{i_{12}}(\xi_1,\xi_2)\upsilon_{j_{12}}(\xi_1,\xi_2)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|^{-1}d^2(\xi_1,\xi_2)\right\}d\xi_3
\end{aligned} \tag{2.3}$$

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 [6]) 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:

$$\begin{aligned}
\mathcal{G}^A_{i_{12}j_{12}}(\xi_3) &:= \int\limits_{\mathcal{T}} \upsilon_{i_{12}}(\xi_1,\xi_2)\upsilon_{j_{12}}(\xi_1,\xi_2)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|^{-1}d^2(\xi_1,\xi_2) \\
\Rightarrow \mathsf{G}_{IJ} = \mathcal{G}_{i_{12}j_{12}i_3j_3} &:= \int\limits_{\mathcal{I}} \nu_{i_3}(\xi_3)\nu_{j_3}(\xi_3)\mathcal{G}^A_{i_{12}j_{12}}(\xi_3)d\xi_3
\end{aligned} \tag{2.4}$$

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

$$\begin{aligned}
\mathcal{G}^A_{i_{12}j_{12}}(\xi_3) &\approx \sum_{m=1}^{M} \upsilon_{i_{12}}(\xi_1^m,\xi_2^m)\upsilon_{j_{12}}(\xi_1^m,\xi_2^m)|\mathcal{J}(\xi_1^m,\xi_2^m,\xi_3)|^{-1}w_{12}^m \\
\mathcal{G}_{i_{12}j_{12}i_3j_3} &\approx \sum_{l=1}^{L} \nu_{i_3}(\xi_3^l)\nu_{j_3}(\xi_3^l)\mathcal{G}^A_{i_{12}j_{12}}(\xi_3^l)w_3^l.
\end{aligned} \tag{2.5}$$

We note here that our definition of auxiliary functions and matrices is not unique. Indeed, a reverse factorization of (2.3) with the area integral on the outside would lead to a definition of $\mathcal{G}^A$ dependant 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 $\mathcal{O}(p^7)$.

The evaluation of the auxiliary sequence (2.5) follows similar logic as was presented for the hexahedra: a quadrature point ξ_3^m is fixed, $\mathcal{G}^A$ is evaluated, then $\mathcal{G}$ is evaluated. The process is iterated until each quadrature point ξ_3^m has been evaluated. The assembly procedure for the L^2 Gram matrix is outlined in Algorithm 1. For subsequent 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 [6].

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 2D simplex
    call setquadrature1D(i_el, p_3 − 1; L, {ζ^l, w^l})
    call setquadrature2D(i_el, p_12 − 1; M, {(ζ_1^m, ζ_2^m), w^m})
    𝒢 ← 0                                                      ▷ Initialize Gram Matrix
    for l = 1 to L do
        call Shape1L2(ζ^l, p_3; {ν_i3(ζ^l)})                   ▷ Evaluate 1D shape functions at ζ^l
        𝒢^A ← 0
        for m = 1 to M do
            call Shape2L2(ζ^m, p_12; {v_i12(ζ_1^m, ζ_2^m)})     ▷ Evaluate 2D shape functions at ζ^m
            ξ_lm ← (ζ_1^m, ζ_2^m, ζ^l)
            call geometry(ξ_lm, i_el; x, 𝒥(ξ_lm), 𝒥^{-1}(ξ_lm), |𝒥|)    ▷ Compute x and Jacobian
            for j_12 = 0 to N_12 do
                for i_12 = j_12 to N_12 do
                    𝒢^A_{i12 j12} ← 𝒢^A_{i12 j12} + v_i12(ζ_1^m, ζ_2^m) v_j12(ζ_1^m, ζ_2^m) |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_{i12 j12 i3 j3} ← 𝒢_{i12 j12 i3 j3} + ν_i3(ζ^l) ν_j3(ζ^l) 𝒢^A_{i12 j12}(ζ^l) w^l
    return G
```

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 G.

A similar analysis of sum-factorization algorithms for the hexahedra in [6] 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

Algorithm 1 (Naked)

```
for l = 1 to L do                                   O(p)
    for m = 1 to M do                               O(p^2)
        for j12 = 0 to N12 − 1 do                   O(p^2)
            for i12 = j12 to N12 − 1 do             O(p^2)      ▷ Total: O(p^7)
    for j3 = 0 to N3 − 1 do                         O(p)
        for i3 = j3 to N3 − 1 do                    O(p)
            for j12 = 0 to N12 − 1 do               O(p^2)
                for i12 = j12 to N12 − 1 do         O(p^2)      ▷ Total: O(p^7)
```

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 (2.3) 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).

2.4 Space H^1

To consider the H^1 energy space, let $\{\varphi\}_{I=0}^{\dim W^p-1}$ 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 $\{\varphi_I\}$ can again be represented in the tensorized form of (2.2) as $\varphi_I = u_{i_{12}}\chi_{i_3}$ where $u_{i_{12}} \in \hat{W}_{\mathcal{T}}^{p_{12}}$ with $i_{12} = 0, ..., \frac{1}{2}(p_{12}+2)(p_{12}+1)-1$; where $\chi_{i_3} \in \hat{W}_{\mathcal{I}}^{p_3}$ with $i_3 = 0, ..., p_3$; and where I is a unique integer identifier dependant on i_{12} and i_3 such that $0 \le I < \dim W^p$. The H^1 Gram matrix is then constructed for ordered pairs of basis elements (φ_I, φ_J) as:

$$\begin{aligned}
\mathsf{G}_{IJ}^{\mathrm{grad}} &= (\varphi_I, \varphi_J)_{H^1(\mathcal{K})} \\
&= \int_{\hat{\mathcal{K}}} \hat{\varphi}_I(\boldsymbol{\xi})\hat{\varphi}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|d^3\boldsymbol{\xi} + \int_{\hat{\mathcal{K}}} \left[\hat{\nabla}\hat{\varphi}_I(\boldsymbol{\xi})\right]^{\mathsf{T}} \mathcal{D}(\boldsymbol{\xi}) \left[\hat{\nabla}\hat{\varphi}_J(\boldsymbol{\xi})\right] |\mathcal{J}(\boldsymbol{\xi})|d^3\boldsymbol{\xi} \\
&= \int_{\mathcal{I}}\int_{\mathcal{T}} \hat{\varphi}_I(\boldsymbol{\xi})\hat{\varphi}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|d(\xi_1,\xi_2)d\xi_3 + \int_{\mathcal{I}}\int_{\mathcal{T}} \left[\hat{\nabla}\hat{\varphi}_I(\boldsymbol{\xi})\right]^{\mathsf{T}} \mathcal{D}(\boldsymbol{\xi}) \left[\hat{\nabla}\hat{\varphi}_J(\boldsymbol{\xi})\right] |\mathcal{J}(\boldsymbol{\xi})|d^2(\xi_1,\xi_2)d\xi_3
\end{aligned} \tag{2.6}$$

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

$$\hat{\nabla}\hat{\varphi}_I = \begin{pmatrix} \partial_{\xi_1} u_{i_{12}}(\xi_1,\xi_2)\chi_{i_3}(\xi_3) \\ \partial_{\xi_2} u_{i_{12}}(\xi_1,\xi_2)\chi_{i_3}(\xi_3) \\ u_{i_{12}}(\xi_1,\xi_2)\chi'_{i_3}(\xi_3) \end{pmatrix}. \tag{2.7}$$

The first integral in (2.6) closely resembles that in (2.3) for the L_2 case (note however the lack of the inverse on $|\mathcal{J}(\boldsymbol{\xi})|$ 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:

$$\begin{aligned}
\mathcal{G}_{i_{12}j_{12}}^{\mathrm{grad}\,A}(\xi_3) &\approx \sum_{n=1}^{N} v_{i_{12}}(\xi_1^n,\xi_2^n)v_{j_{12}}(\xi_1^n,\xi_2^n)|\mathcal{J}(\xi_1^n,\xi_2^n,\xi_3)|w_{12}^n \\
\mathcal{G}_{i_{12}j_{12}i_3j_3}^{\mathrm{grad}} &\approx \sum_{m=1}^{M} \nu_{i_3}(\xi_3^m)\nu_{j_3}(\xi_3^m)\mathcal{G}_{i_{12}j_{12}}^{\mathrm{grad}\,A}(\xi_3^m)w_3^m.
\end{aligned} \tag{2.8}$$

To factor the second integral in (2.6) according to Fubini's theorem we first write (2.7) as a product of 3×3 and 3×1 arrays:

$$\hat{\nabla}\hat{\varphi}_I = U_{i_{12}}(\xi_1,\xi_2)X_{i_3}(\xi_3) = \begin{pmatrix} \partial_{\xi_1}u_{i_{12}}(\xi_1,\xi_2) & - & - \\ - & \partial_{\xi_2}u_{i_{12}}(\xi_1,\xi_2) & - \\ - & - & u_{i_{12}}(\xi_1,\xi_2) \end{pmatrix} \begin{pmatrix} \chi_{i_3}(\xi_3) \\ \chi_{i_3}(\xi_3) \\ \chi'_{i_3}(\xi_3) \end{pmatrix} \tag{2.9}$$

Such a definition naturally leads to the factorization of (2.6) as:

$$\begin{aligned} &\int_{\mathcal{I}}\int_{\mathcal{T}} \left[\hat{\nabla}\hat{\varphi}_I(\boldsymbol{\xi})\right]^{\mathsf{T}} \mathcal{D}(\boldsymbol{\xi}) \left[\hat{\nabla}\hat{\varphi}_J(\boldsymbol{\xi})\right] |\mathcal{J}(\boldsymbol{\xi})| d^2(\xi_1,\xi_2)d\xi_3 \\ &\qquad = \int_{\mathcal{I}} X_{i_3}^{\mathsf{T}}(\xi_3)\left\{\int_{\mathcal{T}} U_{i_{12}}^{\mathsf{T}}(\xi_1,\xi_2)\mathcal{D}(\boldsymbol{\xi})U_{i_{12}}(\xi_1,\xi_2)|\mathcal{J}(\boldsymbol{\xi})|d^2(\xi_1,\xi_2)\right\}X_{i_3}(\xi_3)d\xi_3 \end{aligned} \tag{2.10}$$

To proceed with sum-factorization we introduce an auxiliary function sequence for the computation of (2.10):

$$\begin{aligned} \bar{\mathcal{G}}^{\text{grad}\,A}_{i_{12}j_{12}}(\xi_3) &:= \int_{\mathcal{T}} U_{i_{12}}^{\mathsf{T}}(\xi_1,\xi_2)\mathcal{D}(\xi_1,\xi_2,\xi_3)U_{j_{12}}(\xi_1,\xi_2)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|d^2(\xi_1,\xi_2) \\ \bar{\mathcal{G}}^{\text{grad}}_{i_{12}j_{12}i_3j_3} &:= \int_{\mathcal{I}} X_{i_3}^{\mathsf{T}}(\xi_3)\bar{\mathcal{G}}^{\text{grad}\,A}_{i_{12}j_{12}}(\xi_3)X_{j_3}(\xi_3)d\xi_3 \end{aligned} \tag{2.11}$$

Note in particular that the integral for $\bar{\mathcal{G}}^{\text{grad}\,A}$ in (2.11) evaluates to 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 $\bar{\mathcal{G}}^{\text{grad}\,A}$. The resulting discretized auxiliary function sequence can be represented in terms of arrays as:

$$\begin{aligned} \bar{\mathcal{G}}^{\text{grad}\,A}_{abi_{12}j_{12}}(\xi_3) &\approx \sum_{m=1}^{M} U_{i_{12}aa}(\xi_1^m,\xi_2^m)\mathcal{D}_{ab}(\xi_1^m,\xi_2^m,\xi_3)U_{j_{12}bb}(\xi_1^m,\xi_2^m)|\mathcal{J}(\xi_1^m,\xi_2^m,\xi_3)|w_{12}^m \\ \bar{\mathcal{G}}^{\text{grad}}_{i_{12}j_{12}i_3j_3} &\approx \sum_{l=1}^{L}\sum_{a=1}^{3}\sum_{b=1}^{3} X_{i_3a}(\xi_3^l)\bar{\mathcal{G}}^{\text{grad}\,A}_{abi_{12}j_{12}}(\xi_3^l)X_{j_3b}(\xi_3^l)w_3^l. \end{aligned} \tag{2.12}$$

were subscripts a,b denote vector indices.

The Gram matrix is finally calculated by the addition of (2.8) and (2.12):

$$\mathsf{G}^{\text{grad}}_{IJ} = \mathcal{G}^{\text{grad}}_{i_{12}j_{12}i_3j_3} + \bar{\mathcal{G}}^{\text{grad}}_{i_{12}j_{12}i_3j_3}. \tag{2.13}$$

2.5 Space $\boldsymbol{H}(\text{div})$

In considering the final two energy spaces some additional difficulty is presented by the structure of the polynomial subspaces in (2.2) from which the shape functions are defined. In particular, in both $H(\text{div})$ and $H(\text{curl})$ spaces shape functions will come from two families of shape functions (note that the definition of families here only loosely coincides with the definition in [3]).

The definition of prism shape function families is indicated naturally by the definition of polynomial subspace $\hat{V}^p$ in (2.2):

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

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 $\{\hat{\vartheta}_I\}_{I=0}^{\dim \hat{V}^p}$ be a basis of prismatic shape functions spanning $\hat{V}^p$ that is partitioned into two families as depicted above. Let $N_{12}^1 = \dim \hat{V}_{\mathcal{T}}^{p_{12}}$ denote the number of 2D simplicial shape functions in $\hat{V}_{\mathcal{T}}^{p_{12}}$ used in defining family 1, and $N_{12}^2 = \dim \hat{Y}_{\mathcal{T}}^{p_{12}}$ denote the number of 2D simplicial shape functions in $\hat{Y}_{\mathcal{T}}^{p_{12}}$ used in defining family 2. The approach is outlined as follows: we define $i_{12} = 0, ..., N_{12}^1 + N_{12}^2 - 1$ to enumerate all simplicial components of shape functions in $\hat{V}^p$. The appropriate univariate shape function space ($\hat{Y}_{\mathcal{I}}^{p_3}$ or $\hat{W}_{\mathcal{I}}^{p_3}$) 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.

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

Family 1	**Family 2**
$\hat{\vartheta}_I = \begin{pmatrix} V_{i_{12},1}(\xi_1,\xi_2)\nu_{i_3}(\xi_3) \\ V_{i_{12},2}(\xi_1,\xi_2)\nu_{i_3}(\xi_3) \\ 0 \end{pmatrix}$	$\hat{\vartheta}_I = \begin{pmatrix} 0 \\ 0 \\ v_{i_{12}}(\xi_1,\xi_2)\chi_{i_3}(\xi_3) \end{pmatrix}$
$\begin{aligned} \widehat{\text{div}}\hat{\vartheta}_I &= \left(\partial_x V_{i_{12}}(\xi_1,\xi_2) + \partial_y V_{i_{12}}(\xi_1,\xi_2)\right)\nu_{i_3}(\xi_3) \\ &= \text{div}(V_{i_{12}}(\xi_1,\xi_2))\nu_{i_3}(\xi_3) \end{aligned}$	$\widehat{\text{div}}\hat{\vartheta}_I = v_{i_{12}}(\xi_1,\xi_2)\chi'_{i_3}(\xi_3)$
where $V_{i_{12}} \in \hat{V}_{\mathcal{T}}^{p_{12}}, \quad N_{12}^1 = (p_{12}+2)p_{12},$ $0 \leq i_{12} < N_{12}^1;$	where $v_{i_{12}} \in \hat{Y}_{\mathcal{T}}^{p_{12}}, \quad N_{12}^2 = \frac{1}{2}p_{12}(p_{12}+1),$ $N_{12}^1 \leq i_{12} < N_{12}^1 + N_{12}^2$
and $\nu_{i_3} \in \hat{Y}_{\mathcal{I}}^{p_3}, \quad N_3^1 = p_3,$ $0 \leq i_3 < N_3^1;$	and $\chi_{i_3} \in \hat{W}_{\mathcal{I}}^{p_3}, \quad N_3^2 = p_3 + 1,$ $0 \leq i_3 < N_3^2;$

The Gram matrix $\mathsf{G}_{IJ}^{\text{div}}$ is then calculated for ordered pairs of shape functions $(\vartheta_I, \vartheta_J)$ as:

$$\begin{aligned} \mathsf{G}_{IJ}^{\text{div}} &= (\vartheta_I, \vartheta_J)_{H(\text{div})} \\ &= \int_{\hat{\mathcal{K}}} \hat{\vartheta}_I(\boldsymbol{\xi})^\mathsf{T}\mathcal{C}(\boldsymbol{\xi})\hat{\vartheta}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|^{-1}d^3\boldsymbol{\xi} + \int_{\hat{\mathcal{K}}} \widehat{\text{div}}\hat{\vartheta}_I(\boldsymbol{\xi})\widehat{\text{div}}\hat{\vartheta}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|^{-1}d^3\boldsymbol{\xi} \\ &= \int_{\mathcal{I}}\int_{\mathcal{T}} \hat{\vartheta}_I(\boldsymbol{\xi})^\mathsf{T}\mathcal{C}(\boldsymbol{\xi})\hat{\vartheta}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|^{-1}d^2(\xi_2,\xi_1)d\xi_3 + \int_{\mathcal{I}}\int_{\mathcal{T}} \widehat{\text{div}}\hat{\vartheta}_I(\boldsymbol{\xi})\widehat{\text{div}}\hat{\vartheta}_J(\boldsymbol{\xi})|\mathcal{J}(\boldsymbol{\xi})|^{-1}d^2(\xi_2,\xi_1)d\xi_3. \end{aligned} \quad (2.14)$$

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

The first integral in (2.14) resembles that of (2.10) and can be factored similarly by writing $\hat{\vartheta}_I$ 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:

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

where

$$
W_{i_{12}}(\xi_1,\xi_2) = \begin{cases} \begin{pmatrix} V_{i_{12},1}(\xi_1,\xi_2) & - & - \\ - & V_{i_{12},2}(\xi_1,\xi_2) & - \\ - & - & 0 \end{pmatrix} & \text{if } 0 \le i_{12} < N^1_{12}, \\ \\ \begin{pmatrix} 0 & - & - \\ - & 0 & - \\ - & - & v_{i_{12}}(\xi_1,\xi_2) \end{pmatrix} & \text{if } N^1_{12} \le i_{12} < N^1_{12} + N^2_{12}, \end{cases} \tag{2.15}
$$

and

$$
X_{i_3}(\xi_3, i_{12}) = \begin{cases} \begin{pmatrix} \nu_{i_3}(\xi_3) \\ \nu_{i_3}(\xi_3) \\ 0 \end{pmatrix} & \text{if } 0 \le i_{12} < N^1_{12}, \\ \\ \begin{pmatrix} 0 \\ 0 \\ \chi_{i_3}(\xi_3) \end{pmatrix} & \text{if } N^1_{12} \le i_{12} < N^1_{12} + N^2_{12}. \end{cases} \tag{2.16}
$$

Sum factorization then proceeds by introducing the auxiliary function sequence for computation of the first integral term in (2.14) as:

$$
\begin{aligned}
\mathcal{G}^{\mathrm{div}\,A}_{i_{12}j_{12}}(\xi_3) &:= \int\limits_{\mathcal{T}} W^{\mathsf{T}}_{i_{12}}(\xi_1,\xi_2)\mathcal{C}(\xi_1,\xi_2,\xi_3)W_{j_{12}}(\xi_1,\xi_2)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|^{-1}d^2(\xi_1,\xi_2) \\
\mathcal{G}^{\mathrm{div}}_{i_{12}j_{12}i_3j_3} &:= \int\limits_{\mathcal{I}} X^{\mathsf{T}}_{i_3}(\xi_3,i_{12})\mathcal{G}^{\mathrm{div}\,A}_{i_{12}j_{12}}(\xi_3)X_{j_3}(\xi_3,j_{12})d\xi_3
\end{aligned} \tag{2.17}
$$

Discretization of this auxiliary sequence is accomplished similar to (2.12) by introducing indices $a, b \in \{1,2,3\}$ and will not be repeated here.

Computation of the second integral term in (2.14) 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:

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

and

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

An auxiliary function sequence (similar to that in (2.4) for the L_2 case) can be introduced for the computation of this term.

$$
\begin{aligned}
\bar{\mathcal{G}}^{\mathrm{div}\,A}_{i_{12}j_{12}}(\xi_3) &:= \int\limits_{\mathcal{T}} w_{i_{12}}(\xi_1,\xi_2)w_{j_{12}}(\xi_1,\xi_2)|\mathcal{J}(\xi_1,\xi_2,\xi_3)|^{-1}d^2(\xi_1,\xi_2) \\
\bar{\mathcal{G}}^{\mathrm{div}}_{i_{12}j_{12}i_3j_3} &:= \int\limits_{\mathcal{I}} x_{i_3}(\xi_3,i_{12})\bar{\mathcal{G}}^{\mathrm{div}\,A}_{i_{12}j_{12}}(\xi_3)x_{j_3}(\xi_3,j_{12})d\xi_3
\end{aligned} \tag{2.18}
$$

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

$$
\mathsf{G}^{\mathrm{grad}}_{IJ} = \mathcal{G}^{\mathrm{div}}_{i_{12}j_{12}i_3j_3} + \bar{\mathcal{G}}^{\mathrm{div}}_{i_{12}j_{12}i_3j_3}. \tag{2.19}
$$

2.6 Space $\boldsymbol{H}(\mathrm{curl})$

We conclude this section by presenting a sum factorization for $H(\mathrm{curl})$. 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 (2.2):

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

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

Let the basis $\{\hat{\psi}_I\}_{I=0}^{\dim \hat{Q}^p}$ 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 $\dim Q_{\mathcal{T}}^{p_{12}} + \dim W_{\mathcal{T}}^{p_{12}}$, 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 each families prism shape functions as well as their indexing.

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

Family 1	**Family 2**
$\hat{\psi}_I = \begin{pmatrix} E_{i_{12},1}(\xi_1,\xi_2)\chi_{i_3}(\xi_3) \\ E_{i_{12},2}(\xi_1,\xi_2)\chi_{i_3}(\xi_3) \\ 0 \end{pmatrix}$	$\hat{\psi}_I = \begin{pmatrix} 0 \\ 0 \\ u_{i_{12}}(\xi_1,\xi_2)\nu_{i_3}(\xi_3) \end{pmatrix}$
$\widehat{\mathrm{curl}}\hat{\psi}_I = \begin{pmatrix} -E_{i_{12},2}(\xi_1,\xi_2)\chi'_{i_3}(\xi_3) \\ E_{i_{12},1}(\xi_1,\xi_2)\chi'_{i_3}(\xi_3) \\ \mathrm{curl}(E_{i_{12}}(\xi_1,\xi_2))\chi_{i_3}(\xi_3) \end{pmatrix}$	$\widehat{\mathrm{curl}}\hat{\psi}_I = \begin{pmatrix} \partial_y u_{i_{12}}(\xi_1,\xi_2)\nu_{i_3}(\xi_3) \\ -\partial_x u_{i_{12}}(\xi_1,\xi_2)\nu_{i_3}(\xi_3) \\ 0 \end{pmatrix}$
where $E_{i_{12}} \in \hat{Q}_{\mathcal{T}}^{p_{12}}$, $N_{12}^1 = (p_{12}+2)p_{12}$, $0 \le i_{12} < N_{12}^1$;	where $u_{i_{12}} \in \hat{W}_{\mathcal{T}}^{p_{12}}$, $N_{12}^2 = \frac{1}{2}(p_{12}+2)(p_{12}+1)$, $N_{12}^1 \le i_{12} < N_{12}^1 + N_{12}^2$
and $\chi_{i_3} \in \hat{W}_{\mathcal{I}}^{p_3}$, $N_3^1 = p_3 + 1$, $0 \le i_3 < N_3^1$;	and $\nu_{i_3} \in \hat{Y}_{\mathcal{I}}^{p_3}$, $N_3^2 = p_3$, $0 \le i_3 < N_3^2$;

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

$$\begin{aligned}
\mathsf{G}_{IJ}^{\mathrm{curl}} &= (\psi_I, \psi_J)_{H(\mathrm{curl})} \\
&= \int\limits_{\hat{\mathcal{K}}} \hat{\psi}_I(\boldsymbol{\xi})^\mathsf{T} \mathcal{D}(\boldsymbol{\xi}) \hat{\psi}_J(\boldsymbol{\xi}) |\mathcal{J}(\boldsymbol{\xi})| d^3\boldsymbol{\xi} + \int\limits_{\hat{\mathcal{K}}} \left[\widehat{\mathrm{curl}}\hat{\psi}_I(\boldsymbol{\xi})\right]^\mathsf{T} \mathcal{C}(\boldsymbol{\xi}) \left[\widehat{\mathrm{curl}}\hat{\psi}_J(\boldsymbol{\xi})\right] |\mathcal{J}(\boldsymbol{\xi})|^{-1} d^3\boldsymbol{\xi} \\
&= \int\limits_{\mathcal{I}} \int\limits_{\mathcal{T}} \hat{\psi}_I(\boldsymbol{\xi})^\mathsf{T} \mathcal{D}(\boldsymbol{\xi}) \hat{\psi}_J(\boldsymbol{\xi}) |\mathcal{J}(\boldsymbol{\xi})| d^2(\xi_2,\xi_1) d\xi_3 \\
&\qquad + \int\limits_{\mathcal{I}} \int\limits_{\mathcal{T}} \left[\widehat{\mathrm{curl}}\hat{\psi}_I(\boldsymbol{\xi})\right]^\mathsf{T} \mathcal{C}(\boldsymbol{\xi}) \left[\widehat{\mathrm{curl}}\hat{\psi}_J(\boldsymbol{\xi})\right] |\mathcal{J}(\boldsymbol{\xi})|^{-1} d^2(\xi_2,\xi_1) d\xi_3
\end{aligned} \tag{2.20}$$

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 (2.20) can be factored through Fubini's theorem by introducing array factors similar to (2.15) and (2.16). The auxiliary function sequences for the sum factorization of both integrals are then sufficiently similar to (2.17) 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 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.

Figure 3.1 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. 3.1 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 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 dependant. To minimize computational overhead, arrays for Gram matrix G and auxiliary matrices $\mathcal{G}^A$ were dynamically allocated in contiguous memory; however only minor

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.

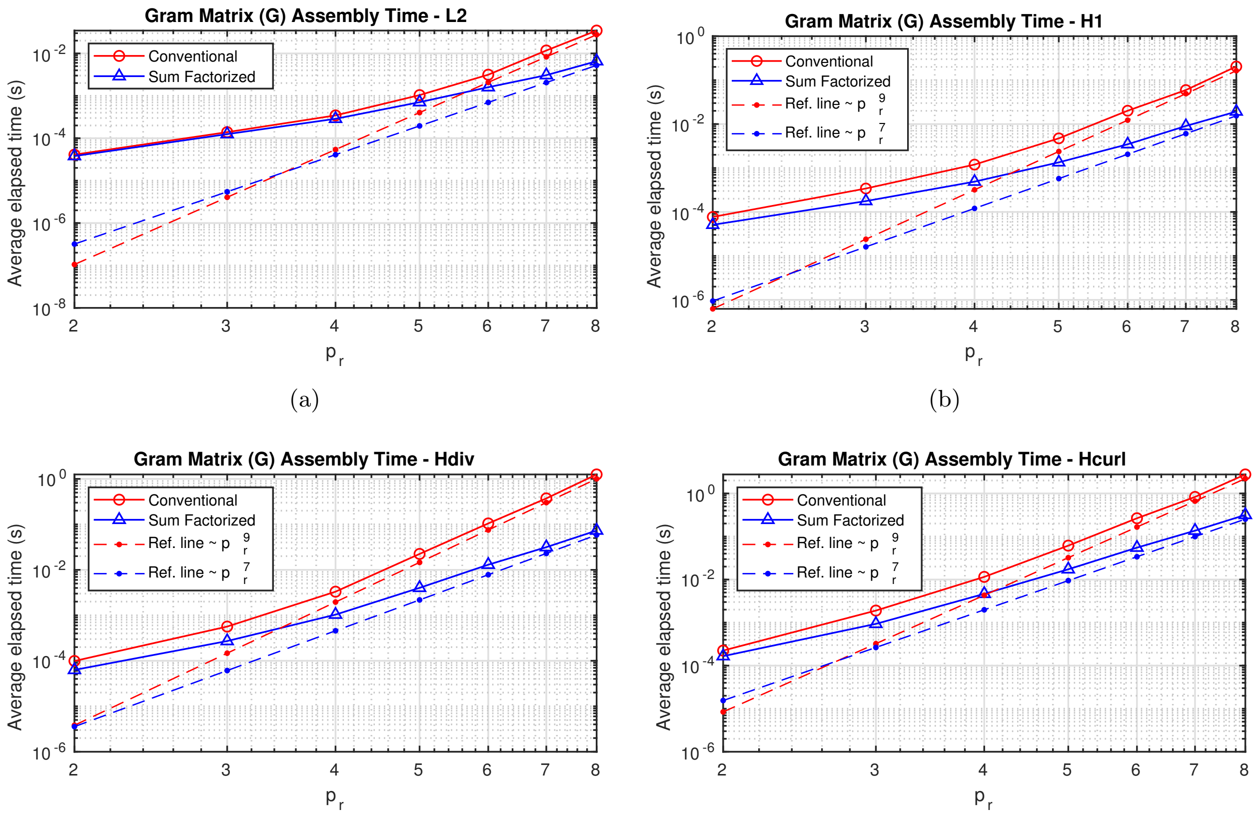
	L^2		H^1	
p_r	Conventional	Sum Factorized	Conventional	Sum Factorized
2	4.1×10^{-5}	3.8×10^{-5}	7.7×10^{-5}	5.1×10^{-5}
3	1.4×10^{-4}	1.3×10^{-4}	3.4×10^{-4}	1.8×10^{-4}
4	3.8×10^{-4}	3.2×10^{-4}	1.3×10^{-3}	5.4×10^{-4}
5	1.1×10^{-3}	7.8×10^{-4}	5.2×10^{-3}	1.5×10^{-3}
6	3.1×10^{-3}	1.6×10^{-3}	2.1×10^{-2}	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**
	$H\,\mathrm{div}$		$H\,\mathrm{curl}$	
p_r	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**

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.

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 outlined in depth in [1, 7], but will be outlined here for completeness.



(a) (b)

(c) (d)

Figure 3.1: Computational times for conventional and sum factorized Gram matrix G assembly for a prismatic element in each exact sequence energy space

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:

$$\left\{ \begin{array}{rll} \operatorname{curl} E + i\omega\mu H & = 0 & \text{in } \Omega \\ \operatorname{curl} H - i\omega\epsilon E & = \mathcal{J}^{imp} & \text{in } \Omega \\ n \times E & = n \times E_0 & \text{on } \Gamma_E \\ n \times H & = n \times H_0 & \text{on } \Gamma_H \end{array} \right. \tag{3.1}$$

where the functions $E, H, \mathcal{J}^{imp} : \Omega \to \mathbb{C}$ 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 [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 $F, G \in H(\mathrm{curl}, \Omega)$ 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:

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

where $(\cdot, \cdot)$ denotes the standard L^2 product, and $\langle \cdot, \cdot \rangle_{\Gamma_h}$ denotes the duality pairing between trace spaces $H^{-1/2}(\mathrm{div}, \Gamma_h)$ and $H^{-1/2}(\mathrm{curl}, \Gamma_h)$. 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 $H(\mathrm{curl}, \Omega_h)$ 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

$$\mathsf{b}\big((E, H, \hat{E}_t, \hat{H}_t), (F, G)\big) = b\big((E, H), (F, G)\big) + \widetilde{b}\big((\hat{E}_t, \hat{H}_t), (F, G)\big) \tag{3.3}$$

where,

$$\begin{aligned} b\big((E, H), (F, G)\big) &= (E, \mathrm{curl}\, F) + (H, \mathrm{curl}\, G) + i\omega(\mu H, F) - i\omega(\epsilon E, G), \\ \widetilde{b}\big((\hat{E}_t, \hat{H}_t), (F, G)\big) &= -\langle n \times \hat{E}_t, F\rangle_{\Gamma_h} - \langle n \times \hat{H}_t, G\rangle_{\Gamma_h}, \end{aligned}$$

and linear functional

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

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 $\mathcal{U} = \big(L^2(\Omega)\big)^6$; $\hat{\mathcal{U}} = \big(H^{-1/2}(\mathrm{curl}, \Gamma_h)\big)^2$; and $\mathcal{V} = \big(H(\mathrm{curl}, \Omega_h)\big)^2$.

Finally, variational problem (3.2) can be cast as a mixed problem by introducing the error representation function ψ (detailed in [1, 7]) as follows: find $\psi \in \mathcal{V}^r, u^h \in \mathcal{U}^h, \widetilde{u}^h \in \hat{\mathcal{U}}^h$ such that

$$\left\{\begin{array}{lll} (\psi, v)_{\mathcal{V}^r(\Omega_h)} - b(u^h, v) - \widetilde{b}(\widetilde{u}^h, v) & = -\ell(v) & \forall v \in \mathcal{V}^r(\Omega_h) \\ b(\delta u, \psi) & = 0 & \forall \delta u \in \mathcal{U}^h \\ \widetilde{b}(\delta \widetilde{u}, \psi) & = 0 & \forall \delta \widetilde{u} \in \hat{\mathcal{U}}^h(\Gamma_h) \end{array}\right. \tag{3.5}$$

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 prescribed in [1]. Finally, defining a discrete trial subspace allows problem (3.5) to be formulated in discrete matrix form as

$$\left\{\begin{array}{ll} \mathsf{Gs} - \mathsf{Bu} - \widetilde{\mathsf{B}}\mathsf{w} & = -\mathsf{l} \\ \mathsf{B}^\mathsf{T}\mathsf{s} & = 0 \\ \widetilde{\mathsf{B}}^\mathsf{T}\mathsf{s} & = 0, \end{array}\right. \tag{3.6}$$

where s, u, and w represent degrees-of-freedom corresponding to ψ, u^h, and $\hat{u}^h$ respectively. Matrix $\widetilde{\mathsf{B}}$ 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.

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 seconds is reduced to a mere 2.4 seconds 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.

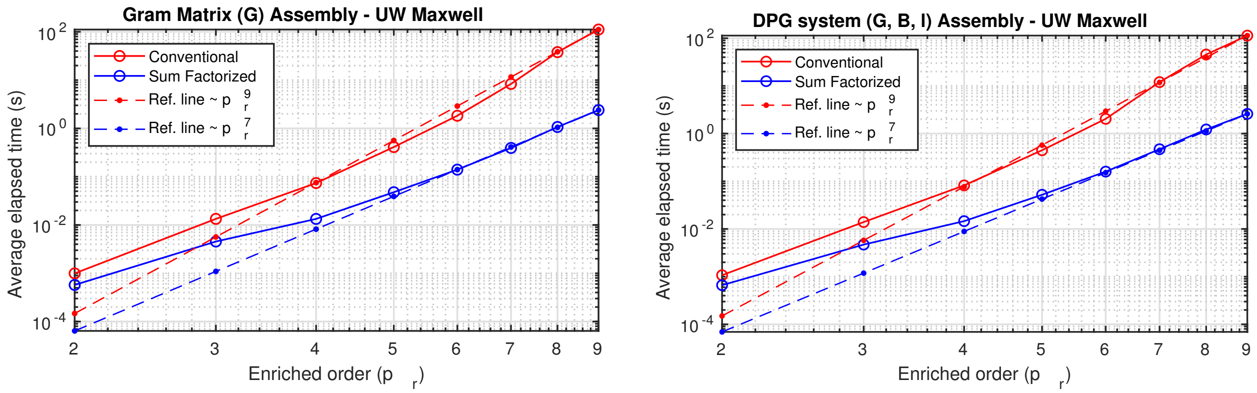
The observed order reported in Table 4 was calculated using regression on the three highest enriched order elements $p_r = 7, 8, 9$ and verifies the respective $\mathcal{O}(p^9)$ and $\mathcal{O}(p^7)$ complexity for conventional and sum factorized assembly. Graphical representation of the data as depicted in Fig. 3.2 reveals that expected asymptotic rates are reached for relatively low polynomial orders p_r.

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

			G Assembly Time (s)		G, B, l Assembly Time (s)	
p_0	Δp	p_r	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**

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 second auxiliary matrices (denoted $\mathcal{G}^B$ in [6]). Note however that this representation allows for polynomial anisotropy in only a single direction



(a) (b)

Figure 3.2: 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

(the 2D square is assumed to be of uniform polynomial order) and therefore may not be suitable for routines in which fully anisotropic polynomial refinements are required.

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. 3.3.

Table 5: Computational times for construction of the ultraweak Maxwell Gram matrix using conventional, partial sum factorization, and full sum factorization techniques

	G Assembly Time (s)		
p_r	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**

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. 3.3 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 computational premium incurred by partial sum factorization.

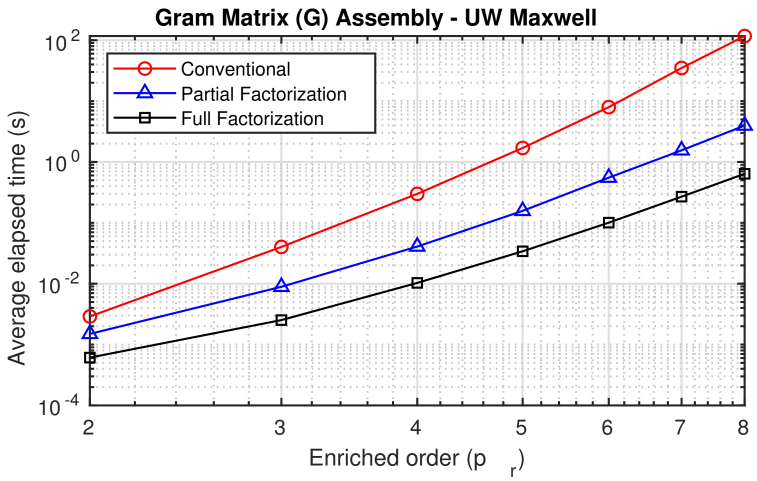


Figure 3.3: Computational times for construction of the ultraweak Maxwell Gram matrix using conventional, partial sum factorization, and full sum factorization techniques

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 [6]. This somewhat unexpected result is achieved since the final compilation of the Gram 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 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× reduced computational expense in the case of enriched order $p_r = 8$ elements.

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 [4] and outlined in [5, 8]. 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] Carstensen, C., Demkowicz, L., and Gopalakrishnan, J. (2016). Breaking spaces and forms for the DPG method and applications including Maxwell equations. *Comput. Math. Appl.*, 72(3):494–522.

[2] Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W., and Zdunek, A. (2007). *Computing with hp Finite Elements. II. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC, New York.

[3] Fuentes, F., Keith, B., Demkowicz, L., and Nagaraj, S. (2015). Orientation embedded high order shape functions for the exact sequence elements of all shapes. *Comput. Math. Appl.*, 70(4):353–458.

[4] Gatto, P. and Demkowicz, L. (2010). Construction of H1-conforming hierarchical shape functions for elements of all shapes and transfinite interpolation. *Finite Elements in Analysis and Design*, 46(6):474–486.

[5] Karniadakis, G. and Sherwin, S. J. (1999). *Spectral/hp Element Methods for CFD*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York.

[6] Mora Paz, J. and Demkowicz, L. (2019). Fast integration of DPG matrices based on sum factorization for all the energy spaces. *Computational Methods in Applied Mathematics*, 19(3):523–555.

[7] Nagaraj, S., Grosek, J., Petrides, S., Demkowicz, L., and J., M. P. (2019). A 3D DPG Maxwell approach to nonlinear Raman gain in fiber laser amplifiers. *J. Comp. Phys.*, 2:100002.

[8] Zaglmayr, S. (2006). *High Order Finite Element Methods for Electromagnetic Field Computation*. PhD thesis, Johannes Kepler Universität Linz, Linz.