Higher Order Fourier Finite Element Methods for Hodge Laplacian Problems on Axisymmetric Domains

Nicole E. Stock
Advisor: Minah Oh, Ph.D.

Abstract

In this paper, we construct a new family of higher order Fourier finite element spaces to discretize the axisymmetric Hodge Laplacian problems. We demonstrate that these new higher order Fourier finite element methods provide improved computational efficiency as well as increased accuracy.

1 Introduction

The finite element method (FEM) is a numerical technique for approximating solutions to complex partial differential equations (PDEs). Many PDEs cannot be solved using analytical techniques; rather their solutions must be approximated. The FEM is an ideal candidate for approximating solutions due to its efficiency, well developed theory, adaptability, and accuracy.

An axisymmetric problem is a three-dimensional (3D) problem that is symmetrical about the z-axis (the axis of rotation). Beyond mathematical research and advancement, axisymmetric problems have a variety of applications in fields such as biomedical engineering, electromagnetism, and optics. For example, the numerical solution for the axisymmetric Maxwell equations can be used to design efficient antennas for hepatic microwave ablation, an alternative treatment to various types of cancer where a small antenna is inserted into the tumor to burn it [3].

Restated, an axisymmetric problem is a 3D problem such that if we take a two-dimensional (2D) slice of the problem domain, then we can regain the entire 3D domain by rotating the 2D slice about the z-axis. This slicing method allows one to perform a dimension reduction, transforming a 3D problem into a set of 2D problems, where the solution to each 2D problem is a Fourier mode of the 3D solution. For any computational method, solving a 2D problem is much more efficient and less complex than solving a 3D problem. Thus, performing a dimension reduction significantly reduces computational efforts. One important thing to note about this dimension reduction is that the 2D problems will be posed in weighted function spaces, with the weight being the radial component \( r \).

1James Madison University, VA (stockne@dukes.jmu.edu)
2Department of Mathematics and Statistics, James Madison University, VA (ohmx@jmu.edu)
This work was partially supported by the National Science Foundation grant number DMS 1913050.
We define the 2D slice on the meridian half-plane, $\mathbb{R}_2^2 = \{(r,z) \in \mathbb{R}^2 : r > 0\}$, as the domain, $\Omega$, and its boundary as $\partial \Omega$. We use $\Gamma_1$ to denote the part of $\partial \Omega$ that is not on the $z$-axis and $\Gamma_0$ to denote the artificial boundary $\partial \Omega \setminus \Gamma_1$.

Given an axisymmetric problem, Fourier FEMs use an appropriate FEM to approximate the solution, where the solution to each 2D problem is the $n$th Fourier mode of the 3D solution.

Given a scalar function $u$ defined on an axisymmetric domain $\tilde{\Omega} \subset \mathbb{R}^3$, $u$ can be written as in [18,19],

$$u = u_0 + \sum_{n=1}^{\infty} u_n \cos(n\theta) + \sum_{n=1}^{\infty} u_{-n} \sin(n\theta).$$

(1.1)

For a vector-valued function, we let $u = u_r e_r + u_\theta e_\theta + u_z e_z$, where $e_r, e_\theta, e_z$ are the cylindrical basis. Then, we divide $u$ into its symmetric and anti-symmetric parts, $u = u_s + u_a$. From [18,19], its Fourier series decomposition is

$$u_s = \begin{bmatrix} u_0^r \\ 0 \\ u_0^z \end{bmatrix} + \sum_{n=1}^{\infty} \begin{bmatrix} u_n^r \cos(n\theta) \\ u_n^\theta \sin(n\theta) \\ u_n^z \cos(n\theta) \end{bmatrix},$$

$$u_a = \begin{bmatrix} 0 \\ u_0^\theta \\ 0 \end{bmatrix} + \sum_{n=1}^{\infty} \begin{bmatrix} u_n^r \sin(n\theta) \\ u_n^\theta \cos(n\theta) \\ u_n^z \sin(n\theta) \end{bmatrix}.$$  

(1.2)

By applying the usual grad, curl, and div operators in cylindrical coordinates to Eq. (1.1) or (1.2), we get the operators for the $n$th Fourier mode,

$$\text{grad}^n_{rz} u = \begin{bmatrix} \partial_r u_r \\ -\frac{n}{r} u_r \\ \partial_z u \end{bmatrix},$$

$$\text{curl}^n_{rz} \begin{bmatrix} u_r \\ u_\theta \\ u_z \end{bmatrix} = \begin{bmatrix} \frac{1}{r} (\frac{\partial_z u_\theta - \partial_\theta u_z}{r} - \frac{\partial_r u_z + \partial_z u_\theta}{r}) \\ \frac{\partial_r u_r - \partial_\theta u_z}{r} \\ u_\theta \\ \partial_r u_r + \frac{u_r - nu_\theta}{r} + \partial_z u_z \end{bmatrix}.$$  

(1.3)

We assume $n > 0$ throughout this paper.

Axisymmetric problems with axisymmetric data have been studied in many previous works including in [2,9,10,12]. Additionally, axisymmetric problems with general data, where the functions are not axisymmetric, have also been studied in [7,8,13,16]. In [1], the Hodge Laplacian and its discretization are studied in a uniform framework. The author of [17] applied the theory developed in [1] to axisymmetric problems with axisymmetric data. The work in [17] was extended in [19] to the Hodge Laplacian problem and its discretization for general axisymmetric problems with non-axisymmetric data. The scalar and vector Laplacian with various boundary conditions are both examples of Hodge Laplacian problems. Additionally, these are important problems because they are fundamental to many problems which arise in mathematical models.
In [19], Oh constructs a new family of lowest order Fourier finite element spaces to approximate Hodge Laplacian problems on axisymmetric domains. Our research expands these results by constructing another family of Fourier finite element spaces using higher order Fourier FEMs. Our higher order Fourier FEMs are based on higher order basis functions and provide improved computational efficiency and accuracy when compared with the lowest order Fourier FEMs.

The following sections of this paper are organized as follows. In section 2, we define the relevant standard finite element spaces and construct the new first order family of Fourier finite element spaces. Section 3 presents the Hodge Laplacian on axisymmetric domains and the four main problems solved in this paper. In section 4, we present the discrete mixed formulation of each problem and describe how matrices and vectors are constructed and used in the FEM program for the $k = 1$ case. Section 5 compares the numerical results of the four first order programs in this work with the lowest order programs in [19].

2 Fourier Finite Element Spaces for Axisymmetric Problems

In the following subsections and throughout this paper, we define a finite element triangulation as $T_h$ on a given domain $\Omega$ such that $T_h$ satisfies the usual geometric conformity conditions in [6]. We assume that $T_h$ is quasi-uniform with a representative mesh size $h$.

The author of [17] constructs the new family of Fourier finite element spaces, denoted by $A_{h0}$, $B_{h0}$, $C_{h0}$, and $D_{h0}$, by using the $P_1$ space and the lowest order Nédelec, Raviart Thomas, and $P_k$ spaces (denoted by $\text{ND}_0$, $\text{RT}_0$, and $P_0$ respectively). The $P_k$ space consists of polynomials of at most degree $k$. Both the Nédelec and Raviart Thomas spaces consist of vector-valued functions. In 2D, the Nédelec and Raviart Thomas spaces are rotations of each other. There are known FEM examples, as in section 2.3 of [1], where the $P_k$ approximation converges to the wrong solution, while the Raviart Thomas approximation converges to the correct solution, thus motivating our use of these spaces. For more details on the Nédelec and Raviart Thomas spaces, refer to [15,20].

The $A_{h0}$ space is constructed from the $P_1$ space, the $B_{h0}$ space from the $P_1$ and the $\text{ND}_0$ spaces, the $C_{h0}$ space from the $\text{RT}_0$ and the $P_0$ spaces, and the $D_{h0}$ space from the $P_0$ space. The main motivation for these spaces is that they satisfy the exact sequence property and the so-called commuting diagram property. Refer to [17,19] for more information on the construction and motivation behind this lowest order family of Fourier finite element spaces which are defined as,

\begin{align*}
A_{h0} &= \{ u \in H_v(\text{grad}^n, \Omega) : u|_T \in A_1 \quad \forall \ T \in T_h \}, \\
B_{h0} &= \{ u \in H_v(\text{curl}^n, \Omega) : u|_T \in B_1 \quad \forall \ T \in T_h \}, \\
C_{h0} &= \{ u \in H_v(\text{div}^n, \Omega) : u|_T \in C_1 \quad \forall \ T \in T_h \}, \\
D_{h0} &= \{ u \in L^2(\Omega) : u|_T \in D_1 \quad \forall \ T \in T_h \},
\end{align*}

(2.1)

where $A_1$, $B_1$, $C_1$, and $D_1$ are lowest order polynomial spaces with the following form,
where \( \alpha_i, \beta_j, \gamma_l \in \mathbb{R} \).

\[
A_1 = \{ \alpha_1 r^2 + \alpha_2 rz + \alpha_3 r \}, \\
B_1 = \left\{ \begin{bmatrix} -\frac{1}{r} \beta_1 + \beta_4 r - \frac{1}{r} \beta_3 z - \beta_6 r z \\ \beta_1 + \beta_2 r + \beta_3 z \\ \beta_5 r + \beta_6 r^2 \end{bmatrix} \right\}, \\
C_1 = \left\{ \begin{bmatrix} \gamma_1 + \gamma_2 r \\ \frac{1}{r} \gamma_1 + \gamma_3 r \\ \gamma_4 + \gamma_2 z \end{bmatrix} \right\}, \\
D_1 = \{ \alpha_1 \}.
\]

(2.2)

The weighted Hilbert spaces used in Eq. (2.1) are defined as:

\[
L^2(\Omega) = \left\{ u : \iint_{\Omega} u(r,z)^2 r \, drdz < \infty \right\}, \\
H_r(\text{grad}, \Omega) = \{ u \in L^2(\Omega) : \text{grad}_r u \in L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega) \}, \\
H_r(\text{curl}, \Omega) = \{ u \in L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega) : \text{curl}_r u \in L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega) \}, \\
H_r(\text{div}, \Omega) = \{ u \in L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega) : \text{div}_r u \in L^2(\Omega) \}.
\]

Additionally, the corresponding standard (non-weighted) function spaces are defined as:

\[
L^2(\Omega) = \left\{ u : \iint_{\Omega} u(r,z)^2 \, dA < \infty \right\}, \\
H^1(\Omega) = \{ u \in L^2(\Omega) : \text{grad} u \in L^2(\Omega) \times L^2(\Omega) \}, \\
H(\text{curl}, \Omega) = \{ u \in L^2(\Omega) \times L^2(\Omega) : \text{curl} u \in L^2(\Omega) \}, \\
H(\text{div}, \Omega) = \{ u \in L^2(\Omega) \times L^2(\Omega) : \text{div} u \in L^2(\Omega) \}.
\]

Throughout the paper, we will use \( N \) to denote the number of vertices, \( N_m \) for the number of midpoints, \( N_e \) for the number of edges, and \( N_t \) for the number of triangles all on a particular mesh. We also use \( v_i \) to denote the \( i^{\text{th}} \) vertex, \( e_i \) to denote the \( i^{\text{th}} \) edge, and \( T_i \) to denote the \( i^{\text{th}} \) triangle. We use the vertices \((x_{i_1}, y_{i_1})\) and \((x_{i_2}, y_{i_2})\) to denote the endpoints of edge \( e_i \). Lastly, we use \( \vec{n}^{(i)} \) to denote the unit outward normal vector to \( e_i \) and \( \vec{t}^{(i)} \) to denote the unit tangent vector to \( e_i \).

We will construct a new first order family of Fourier finite element spaces from the P\(_2\) space and the first order spaces ND\(_1\), RT\(_1\), and P\(_1\) in a similar way to how [17, 19] construct the lowest order family of spaces. We define the P\(_2\), ND\(_1\), RT\(_1\), and P\(_1\) spaces in the following subsections.

### 2.1 P\(_2\) Space

The P\(_2\) space is defined as

\[
P_2 = \left\{ u \in H^1(\Omega) : u|_T = \alpha_1 r^2 + \alpha_2 rz + \alpha_3 z^2 + \alpha_4 r + \alpha_5 z + \alpha_6 \quad \forall \, T \in T_h \right\}.
\]
For each triangle, there are 6 degrees of freedom for the \(P_2\) space; 1 for each vertex and 1 for each edge midpoint. Let us use \(\{\psi_i\}_{i=1}^{N+N_m}\) to denote the basis of \(P_2\).

The following must hold for each of the degrees of freedom used to construct the basis for the \(P_2\) space, where \(v_j\) represents each vertex and edge midpoint in the mesh,

\[
\psi_i(v_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.
\] (2.3)

### 2.2 ND\(_1\) Space

The ND\(_1\) space is defined as

\[
\text{ND}_1 = \left\{ u \in H(\text{curl}, \Omega) : u|_T = \left[ \beta_1 r + \beta_2 z + \beta_3 - \beta_7 rz - \beta_8 z^2 \right] \quad \forall T \in T_h \right\}.
\]

For each triangle, there are 8 degrees of freedom for the ND\(_1\) space; 2 for each edge and 2 for the triangular element. Let us use \(\{\Phi_i\}_{i=1}^{2N_e+2N_t}\) to denote the basis of ND\(_1\).

The following must hold for each of the degrees of freedom used to construct the basis for the ND\(_1\) space, where \(T_j\) represents each triangle in the mesh and \((x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2})\) represent the pair of endpoints for each edge in the mesh,

\[
\int_0^1 f(x(t), y(t)) \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},
\]

\[
\int_0^1 f(x(t), y(t)t) \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},
\]

\[
\int_{T_j} \Phi_i \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},
\]

\[
\int_{T_j} \Phi_i \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},
\] (2.4)

where

\[ f(x, y) = \Phi_i \cdot \vec{f}(i), \]

\[ \langle x(t), y(t) \rangle = \langle x_{i_1}, y_{i_1} \rangle (1 - t) + \langle x_{i_2}, y_{i_2} \rangle t, \text{ for } 0 \leq t \leq 1, \]

\[ x'(t) = \frac{d}{dt} x(t), \]

\[ y'(t) = \frac{d}{dt} y(t). \]

### 2.3 RT\(_1\) Space

The RT\(_1\) space is defined as

\[
\text{RT}_1 = \left\{ u \in H(\text{curl}, \Omega) : u|_T = \left[ \gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz \right] \quad \forall T \in T_h \right\}.
\]

For each triangle, there are 8 degrees of freedom for the RT\(_1\) space; 2 for each edge and 2 for the triangular element. Let us use \(\{\phi_i\}_{i=1}^{2N_e+2N_t}\) to denote the basis of RT\(_1\).
The following must hold for each of the degrees of freedom used to construct the basis for the RT$_1$ space, where $T_j$ represents each triangle in the mesh and $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$ represent the pair of endpoints for each edge in the mesh,

$$\int_0^1 f(x(t), y(t)) \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\int_0^1 f(x(t), y(t)) t \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\int_{T_j} \phi_i \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\int_{T_j} \phi_i \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

(2.5)

where

$$f(x, y) = \phi_i \cdot \vec{n}^{(i)},$$

$$\langle x(t), y(t) \rangle = \langle x_{j1}, y_{j1} \rangle (1 - t) + \langle x_{j2}, y_{j2} \rangle t,$$ for $0 \leq t \leq 1,$

$$x'(t) = \frac{d}{dt} x(t),$$

$$y'(t) = \frac{d}{dt} y(t).$$

### 2.4 P$_1$ Space

The P$_1$ space is defined as

$$P_1 = \{ u \in H^1(\Omega) : u|_T = \alpha_1 r + \alpha_2 z + \alpha_3 \, \forall \, T \in T_h \}.$$  

For each triangle, there are 3 degrees of freedom for the P$_1$ space; 1 for each vertex. Let us use $\{\chi_i\}_{i=1}^{N}$ to denote the basis of P$_1$.

The following must hold for each of the degrees of freedom used to construct the basis for the P$_1$ space, where $v_j$ represents each vertex in the mesh,

$$\chi_i(v_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

(2.6)

### 2.5 New Family of Fourier Finite Element Spaces

We will use the notation $A_{h_1}, B_{h_1}, C_{h_1},$ and $D_{h_1}$ to represent the new first order family of Fourier finite element spaces that we construct in this section. We construct the new first order family of Fourier finite element spaces in a consistent way to what the author of [17] did for the lowest order family of Fourier finite element spaces. The lowest order family satisfies the so-called commuting diagram property (Theorem 4.1 of [17]), and the first order family will continue to satisfy this property.
We define the first order family of Fourier finite element spaces as
\[
A_{h_1} = \{ u \in H_r(\text{grad}^n, \Omega) : u|_T \in A_2 \ \forall \ T \in T_h \}, \\
B_{h_1} = \{ u \in H_r(\text{curl}^n, \Omega) : u|_T \in B_2 \ \forall \ T \in T_h \}, \\
C_{h_1} = \{ u \in H_r(\text{div}^n, \Omega) : u|_T \in C_2 \ \forall \ T \in T_h \}, \\
D_{h_1} = \{ u \in L^2_r(\Omega) : u|_T \in D_2 \ \forall \ T \in T_h \},
\]
where \(A_2, \ B_2, \ C_2,\) and \(D_2\) are first order polynomial spaces with the following form and \(\alpha_i, \beta_j, \gamma_l \in \mathbb{R}\).

\[
A_2 = \{ \alpha_1 r^3 + \alpha_2 r^2 z + \alpha_3 rz^2 + \alpha_4 r^2 + \alpha_5 rz + \alpha_6 r \}, \\
B_2 = \left\{ \begin{bmatrix}
\frac{1}{n} (\beta_1 r^2 + \beta_2 rz + \beta_3 r - \beta_4 r^2 z - \beta_5 rz^2 - \beta_6 r - \beta_7 r^2 - \beta_8 rz - \beta_9 z^2 - \beta_{10} z - \beta_{11}) \\
\frac{1}{n} (\beta_6 r^2 + \beta_7 rz + \beta_8 r - \beta_9 r^2 z - \beta_{10} rz^2 - \beta_{11} r - \beta_{12} r^2 - \beta_{13} r - \beta_{14}) \\
\frac{1}{n} (\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_4 r z^2 + \gamma_{12} r z + \gamma_{13} z + \gamma_{14}) \\
\frac{1}{n} (\gamma_4 r + \gamma_5 z + \gamma_6 + \gamma_7 r z + \gamma_{14}^2 z^2)
\end{bmatrix} : \right\}, \\
C_2 = \left\{ \begin{bmatrix}
\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_4 r z^2 + \gamma_{12} r z + \gamma_{13} z + \gamma_{14}
\gamma_4 r + \gamma_5 z + \gamma_6 + \gamma_7 r z + \gamma_{14} z^2
\end{bmatrix} : \right\}, \\
D_2 = \{ \alpha_1 r + \alpha_2 z + \alpha_3 \}.
\]

The construction of these spaces is given in more detail in the following subsections.

### 2.6 \(A_{h_1}:\) Weighted Fourier \(P_2\) Space

There are 6 local degrees of freedom in the \(A_{h_1}\) space; 1 for each vertex and 1 for each edge midpoint. In comparison, there are 3 degrees of freedom for each triangle in the \(A_{h_0}\) space; 1 for each vertex. Let us use \(\{\psi_i^n\}_{i=1}^{n+3m}\) to denote the basis of \(A_{h_1}\).

The following must hold for each of the degrees of freedom used to construct the basis for the \(A_{h_1}\) space, where \(v_j\) represents each vertex and edge midpoint in the mesh,

\[
\frac{n}{r} \psi_i^n(v_j) = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j
\end{cases}
\]

The basis functions on each triangle have the following form and satisfy Eq. (2.9),

\[
\psi_i^n = \frac{r}{n} i \\
= \frac{r}{n} (\alpha_1 r^2 + \alpha_2 r z + \alpha_3 z^2 + \alpha_4 r + \alpha_5 z + \alpha_6) \\
= \frac{1}{n} (\alpha_1 r^3 + \alpha_2 r^2 z + \alpha_3 rz^2 + \alpha_4 r^2 + \alpha_5 rz + \alpha_6 r),
\]

where \(\psi_i\) represents the basis functions for the \(P_2\) space.

### 2.7 \(B_{h_1}:\) Weighted Fourier \(ND_1\) and \(P_2\) Space

There are 14 local degrees of freedom in the \(B_{h_1}\) space; 2 for each edge, 1 for each vertex and edge midpoint, and 2 for the triangular element. The lowest order space,
$B_{h_0}$, has only 6 local degrees of freedom; 1 for edge and 1 for each vertex. Let us use \( \{ \zeta_i \}_{i=1}^{2N_e+2N_t+N+N_m} = \{ \Phi_i \}_{j=1}^{2N_e+2N_t} \cup \{ \Psi_j \}_{j=1+\left(2N_e+2N_t\right)}^{N+N_m+(2N_e+2N_t)} \) to denote the basis of $B_{h_1}$.

The following must hold for each of the degrees of freedom used to construct the basis for the $B_{h_1}$ space, where $v_j$ represents each vertex and edge midpoint in the mesh, $T_j$ represents each triangle in the mesh, and $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2})$ represent the pair of endpoints for each edge in the mesh,

\[
\zeta_{ij}^n(v_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.11.1)
\]

\[
\int_0^1 f(x(t), y(t)) \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.11.2)
\]

\[
\int_0^1 f(x(t), y(t)) t \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.11.3)
\]

\[
\int_{T_j} n \zeta_{ij}^n + \zeta_{ij}^n \frac{r}{n} \frac{n}{n} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.11.4)
\]

\[
\int_{T_j} n \zeta_{ij}^n + \zeta_{ij}^n \frac{r}{n} \frac{n}{n} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \, dA = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.11.5)
\]

where

\[
f(x, y) = \begin{bmatrix} n \zeta_{ij}^n + \zeta_{ij}^n \\ n \zeta_{ij}^n + \zeta_{ij}^n \frac{r}{n} \end{bmatrix} \cdot t^{(i)},
\]

\[
\langle x(t), y(t) \rangle = \langle x_{j_1}, y_{j_1} \rangle (1-t) + \langle x_{j_2}, y_{j_2} \rangle t, \quad \text{for } 0 \leq t \leq 1.
\]

We note that Eq. (2.11.1) is the same as what must hold for the basis of $P_2$, $\{ \psi_j \}_{j=1}^{N+N_m}$. Therefore, for the subset of basis functions on vertices and midpoints in the $B_{h_1}$ space, $\{ \Psi_j \}_{j=1+\left(2N_e+2N_t\right)}^{N+N_m+(2N_e+2N_t)}$, we let

\[
\Psi_{j_0}^n = \psi_j = \alpha_1 r^2 + \alpha_2 r z + \alpha_3 z^2 + \alpha_4 r + \alpha_5 z + \alpha_6.
\]

Furthermore, to satisfy Eq. (2.11.2 - 2.11.5) for this subset of basis functions, we let

\[
\Psi_{j_0}^n = -\frac{1}{n} \Psi_{j_0}^n = -\frac{1}{n} (\alpha_1 r^2 + \alpha_2 r z + \alpha_3 z^2 + \alpha_4 r + \alpha_5 z + \alpha_6),
\]

and

\[
\Psi_{j_2}^n = 0,
\]

so that

\[
\begin{bmatrix} n \Psi_{j_0}^n + \Psi_{j_0}^n \\ n \Psi_{j_0}^n + \Psi_{j_0}^n \frac{r}{n} \end{bmatrix} = \begin{bmatrix} n(-\frac{1}{n} \Psi_{j_0}^n + \Psi_{j_0}^n) \\ \frac{r}{n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

At this point, we have defined the subset of basis functions, $\{ \Psi_j \}_{j=1+\left(2N_e+2N_t\right)}^{N+N_m+(2N_e+2N_t)}$, so that they satisfy Eq. (2.11.1 - 2.11.5).

Now for the subset of basis functions on edges and triangles in the $B_{h_1}$ space, $\{ \Phi_i \}_{i=1}^{2N_e+2N_t}$. If we let $\Phi_{i_0}^n = 0$, then we guarantee that the basis functions will satisfy Eq. (2.11.1).
Furthermore, with $\Phi^n_{i\theta} = 0$, then
\[
\begin{bmatrix}
\frac{n}{r} (\Phi^n_{ir} + \Phi^n_{i\theta}) \\
\frac{r}{n} \Phi^n_{i\theta}
\end{bmatrix} = \begin{bmatrix}
\frac{r}{n} \Phi^n_{ir} \\
\frac{r}{n} \Phi^n_{i\theta}
\end{bmatrix}.
\]

We note that Eq. (2.11.2 - 2.11.5) is very similar to what must hold for the basis of $\text{ND}_1$, $\{\Phi_i\}_{i=1}^{2N_e+2N_t}$, as in Eq. (2.4). Thus, if we let
\[
\begin{bmatrix}
\frac{r}{n} \Phi^n_{ir} \\
\frac{r}{n} \Phi^n_{i\theta}
\end{bmatrix} = \begin{bmatrix}
\phi_{ir} \\
\phi_{i\theta}
\end{bmatrix},
\]
then Eq. (2.11.2 - 2.11.5) matches Eq. (2.4). We define
\[
\Phi^n_{ir} = \frac{r}{n} \Phi_{ir},
\]
\[
= \frac{1}{n} (\beta_1 r + \beta_2 z + \beta_3 - \beta_7 r z - \beta_8 z^2)
\]
\[
= \frac{1}{n} (\beta_1 r^2 + \beta_2 r z + \beta_3 r - \beta_7 r^2 z - \beta_8 r z^2),
\]
and
\[
\Phi^n_{i\theta} = \frac{r}{n} \Phi_{i\theta},
\]
\[
= \frac{1}{n} (\beta_4 r + \beta_5 z + \beta_6 + \beta_7 r^2 + \beta_8 r z)
\]
\[
= \frac{1}{n} (\beta_4 r^2 + \beta_5 r z + \beta_6 r + \beta_7 r^3 + \beta_8 r^2 z),
\]
to satisfy Eq. (2.11.2 - 2.11.5) for the subset of basis functions $\{\Phi^n_{i}\}_{i=1}^{2N_e+2N_t}$.

The basis functions for the $B_{h_1}$ space have the following form on each triangle,
\[
\Phi^n = \begin{bmatrix}
\frac{1}{n} (\beta_1 r^2 + \beta_2 r z + \beta_3 r - \beta_7 r^2 z - \beta_8 r z^2) \\
0
\end{bmatrix},
\]
\[
\Psi_j^n = \begin{bmatrix}
\frac{1}{n} (\alpha_1 r^2 + \alpha_2 r z + \alpha_3 z^2 + \alpha_4 r + \alpha_5 z + \alpha_6) \\
0
\end{bmatrix},
\]
(2.12)

2.8 $C_{h_1}$: Weighted Fourier RT$_1$ and P$_1$ Space

There are 11 local degrees of freedom in the $C_{h_1}$ space. On each triangle in the mesh, there are 2 basis functions for each edge, 1 for each vertex, and 2 for the triangular element. The $C_{h_0}$ space has 4 degrees of freedom for each triangle; 1 for each edge and 1 for the triangular element. Let us use $\{c^n_i\}_{i=1}^{2N_e+2N_t + N} = \{\phi^n_i\}_{i=1}^{2N_e+2N_t} \cup \{\chi^n_j\}_{j=1+(2N_e+2N_t)}^N$ to denote the basis of $C_{h_1}$.

The following must hold for each of the degrees of freedom used to construct the basis for the $C_{h_1}$ space, where $T_j$ represents each triangle in the mesh, $v_j$ represents each vertex
in the mesh, and \((x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2})\) represent the pair of endpoints for each edge in the mesh,

\[
\int_0^1 f(x(t), y(t)) \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \tag{2.13.1}
\]

\[
\int_0^1 f(x(t), y(t)) t \sqrt{x'(t)^2 + y'(t)^2} \, dt = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \tag{2.13.2}
\]

\[
\int_{T_i} \left[ \begin{array}{c} \xi_i^n \\ \eta_i^n \\ \zeta_i^n \end{array} \right] \cdot \left[ \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right] \, dA = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \tag{2.13.3}
\]

\[
\int_{T_i} \left[ \begin{array}{c} \xi_i^n \\ \eta_i^n \\ \zeta_i^n \end{array} \right] \cdot \left[ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right] \, dA = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \tag{2.13.4}
\]

\[
\frac{n \xi_{i_0}^n - \xi_i^n}{r} (v_j) = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}, \tag{2.13.5}
\]

where

\[
f(x, y) = \left[ \begin{array}{c} \xi_i^n \\ \eta_i^n \\ \zeta_i^n \end{array} \right] \cdot \vec{n}^{(i)},
\]

\[
\langle x(t), y(t) \rangle = \langle x_{j_1}, y_{j_1} \rangle (1 - t) + \langle x_{j_2}, y_{j_2} \rangle t, \text{ for } 0 \leq t \leq 1.
\]

Notice that Eq. (2.13.1 - 2.13.4) is the same as Eq. (2.5), which details what must hold for the basis of \(RT_1, \{\phi_i\}_{i=1}^{2N_c+2N_t}\). Therefore, for the subset of basis functions on edges and triangles in the \(C_h\) space, \(\{\phi_i^n\}_{i=1}^{2N_c+2N_t}\), we let

\[
\phi_i^n = \phi_{i_r} = \gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz,
\]

and

\[
\phi_i^n = \phi_{i_z} = \gamma_4 r + \gamma_5 z + \gamma_6 + \gamma_7 rz + \gamma_8 z^2.
\]

Then, we need to satisfy Eq. (2.13.5) for this subset of basis functions. This is done by solving the following

\[
\frac{n \phi_i^n - \phi_{i_0}^n}{r} = 0
\]

\[
\frac{n}{r} \phi_i^n - \frac{1}{r} (\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz) = 0
\]

\[
\frac{n}{r} \phi_i^n = \frac{1}{r} (\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz)
\]

\[
\phi_i^n = \frac{1}{n} (\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz)
\]

Thus, we let \(\phi_i^n = \frac{1}{n} (\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz)\).

Now, to solve for the subset of basis functions on vertices in the \(C_h\) space, \(\{\chi_j^n\}_{j=1}^{N_0+(2N_c+2N_t)}\), we begin by letting \(\chi_{j_r}^n, \chi_{j_z}^n = 0\) to satisfy Eq. (2.13.1 - 2.13.4). Then, we need to satisfy Eq. (2.13.5). We notice that by setting \(\chi_{j_r}^n = 0\), then

\[
\frac{n \chi_i^n - \chi_{j_r}^n}{r} = \frac{n}{r} \chi_{j_r}^n.
\]
Now the condition looks similar to that of the $P_1$ space in Eq. (2.6), so we let
\[ \chi^n_j = \frac{r}{n} \chi_j = \frac{1}{n} (\alpha_1 r^2 + \alpha_2 rz + \alpha_3 r), \]
where $\chi_j$ is the basis function for the $P_1$ space. We see that
\[ \frac{n \chi^n_j}{r} = \frac{n(\frac{1}{n})(\alpha_1 r^2 + \alpha_2 rz + \alpha_3 r)}{r} = \alpha_1 r + \alpha_2 z + \alpha_3 = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \]
which satisfies Eq. (2.13.5) in the same way that the basis functions for the $P_1$ space do.

The basis functions for the $C_{h_1}$ space have the following form on each triangle,
\[
\phi^n_i = \begin{bmatrix}
\gamma_1 r + \gamma_2 z + \gamma_3 + \gamma_7 r^2 + \gamma_8 rz \\
\gamma_4 r + \gamma_5 z + \gamma_6 + \gamma_7 r^2 + \gamma_8 z^2
\end{bmatrix}, \\
\chi^n_j = \begin{bmatrix}
\frac{1}{n}(\alpha_1 r^2 + \alpha_2 rz + \alpha_3 r) \\
0
\end{bmatrix}.
\] (2.14)

### 2.9 $D_{h_1}$: Weighted Fourier $P_1$ Space

Similarly to how $D_{h_0}$ was set to be the $P_0$ space, we let $D_{h_1}$ be the $P_1$ space. We use $\{\chi_i\}^N_{i=1}$ to denote the basis of $D_{h_1}$ as well as $P_1$, where $\chi_i$ has the following form on each triangle,
\[ \chi_i = \alpha_1 r + \alpha_2 z + \alpha_3. \] (2.15)

### 3 Hodge Laplacian Problems on Axisymmetric Domains

The main framework for the Hodge Laplacian with $k = 0, 1, 2, 3$ was built in [1]. A similar framework is established for axisymmetric problems in [17, 19] where the abstract Hodge Laplacian with $k = 0, 1, 2, 3$ is defined in the following way,
\[ L_k = d^{k-1} \delta_k + \delta_{k+1} d^k. \] (3.1)

We define $d^k$ as,
\[
\begin{align*}
d^0 u &= \text{grad}^n_{rz} u, \\
d^1 u &= \text{curl}^n_{rz} u, \\
d^2 u &= \text{div}^n_{rz} u, \\
d^3 u &= 0,
\end{align*}
\]
with the operators in (1.3). We define $\delta_k$ as,
\[
\begin{align*}
\delta_0 u &= 0, \\
\delta_1 u &= -\text{div}^n_{rz} u, \\
\delta_2 u &= \text{curl}^n_{rz} u, \\
\delta_3 u &= -\text{grad}^n_{rz} u,
\end{align*}
\]
with the following operators,

\[
\text{grad}^{n*}_{rz} u = \begin{bmatrix}
\frac{\partial u}{\partial r} \\
\frac{\partial u}{\partial \theta} \\
\frac{\partial u}{\partial z}
\end{bmatrix},
\]

\[
\text{curl}^{n*}_{rz} \begin{bmatrix}
u_r \\
u_\theta \\
u_z
\end{bmatrix} = \begin{bmatrix}
\frac{n}{r} u_z - \frac{\partial u_\theta}{\partial z} \\
\frac{\partial u_r}{\partial z} - \frac{\partial u_z}{\partial r} \\
-\frac{nu_r + u_\theta}{r} + \frac{\partial u_\theta}{\partial r}
\end{bmatrix},
\]

\[
\text{div}^{n*}_{rz} \begin{bmatrix}
u_r \\
u_\theta \\
u_z
\end{bmatrix} = \partial_r u_r + \frac{u_r + nu_\theta}{r} + \partial_z u_z.
\]

Note that when \(k = 0\), \(d^{k-1}\) does not exist and similarly, when \(k = 3\), \(\delta^{k+1}\) does not exist. Therefore, the affected terms are set equal to zero.

We denote the domain of \(L_k\) as \(D^k_L\), which is defined by

\[
D^k_L = \{ u \in V^k \cap V^*_k \mid d^k u \in V^*_k, \quad \delta_k u \in V^{k-1} \}.
\]

The spaces \(V^k\) and \(V^*_k\) are Hilbert spaces associated with their respective operators,

\[
V^0 = H_r(\text{grad}^n, \Omega),
V^1 = H_r(\text{curl}^n, \Omega),
V^2 = H_r(\text{div}^n, \Omega),
V^3 = L^2_r(\Omega),
\]

\[
V^*_0 = L^2_r(\Omega),
V^*_1 = H_r,0(\text{div}^{n*}, \Omega)
= \{ (u_r, u_\theta, u_z) \in L^2_r(\Omega) \times L^2_r(\Omega) \times L^2_r(\Omega) : \text{div}^{n*}_{rz} u \in L^2_r(\Omega), (u_r, u_z) \cdot n = 0 \text{ on } \Gamma_1 \},
V^*_2 = H_r,0(\text{curl}^{n*}, \Omega)
= \{ (u_r, u_\theta, u_z) \in L^2_r(\Omega) \times L^2_r(\Omega) \times L^2_r(\Omega) : \text{curl}^{n*}_{rz} u \in L^2_r(\Omega) \times L^2_r(\Omega) \times L^2_r(\Omega),
(u_r, u_z) \cdot t = 0 \text{ and } u_\theta = 0 \text{ on } \Gamma_1 \},
V^*_3 = H_r,0(\text{grad}^{n*}, \Omega)
= \{ (u_r, u_\theta, u_z) \in L^2_r(\Omega) : \text{grad}^{n*}_{rz} u \in L^2_r(\Omega) \times L^2_r(\Omega) \times L^2_r(\Omega), u = 0 \text{ on } \Gamma_1 \}.
\]

From [19], a solution \(u \in D^k_L\) that satisfies the Hodge Laplacian problem \(L_k u = f\), must satisfy the following

\[
(d^k u, d^k v)_{L^2_r(\Omega)} + (\delta_k u, \delta_k v)_{L^2_r(\Omega)} = (f, v)_{L^2_r(\Omega)} \quad \forall \ v \in V^k \cap V^*_k.
\]

Let us use \((\cdot, \cdot)_{L^2_r(\Omega)}\) to denote the inner-product associated with \(L^2_r(\Omega)\), i.e.,

\[
(u, v)_{L^2_r(\Omega)} = \int \int_{\Omega} u v r dr dz.
\]
Finally, we can define the weighted mixed formulation of the abstract Hodge Laplacian with $k = 0, 1, 2, 3$. Let $\sigma = \delta_k u$, then $\sigma$ and $u$ satisfy the following mixed formulation:

Find $(\sigma, u) \in V^{k-1} \times V^k$ such that,

$$
(\sigma, \tau)_{L^2_\delta(\Omega)} - (d^{k-1} \tau, u)_{L^2_\delta(\Omega)} = 0, \quad \forall \, \tau \in V^{k-1},
$$

$$
(d^{k-1} \sigma, v)_{L^2_\delta(\Omega)} + (d^k u, d^k v)_{L^2_\delta(\Omega)} = (f, v)_{L^2_\delta(\Omega)}, \quad \forall \, v \in V^k.
$$

(3.2)

The discretization of each of the four Hodge Laplacian problems with $k = 0, 1, 2, 3$ are given in detail in the next section. The problem as written in Eq. (3.2) corresponds to the following four problems as shown in [19].

$k = 0$: The Neumann Problem for the Axisymmetric Poisson Equation

$$
-\text{div}_{rz}^n \text{grad}_{rz}^n u = f \quad \text{in} \, \Omega,
\text{grad}_{rz}^n u \cdot \vec{n} = 0 \quad \text{on} \, \Gamma_1.
$$

(3.3)

$k = 1$: The Axisymmetric Vector Laplacian $\text{curl} \, \text{curl} + \text{grad} \, \text{div}$

$$
-\text{grad}_{rz}^n \text{div}_{rz}^n u + \text{curl}_{rz}^n \text{curl}_{rz}^n u = f,
(\text{curl}_{rz}^n u)_{rz} \cdot \vec{r} = 0, \, (\text{curl}_{rz}^n u)_a = 0, \, u_{rz} \cdot \vec{n} = 0 \quad \text{on} \, \Gamma_1.
$$

(3.4)

$k = 2$: The Axisymmetric Vector Laplacian $\text{curl} \, \text{curl} + \text{grad} \, \text{div}$

$$
\text{curl}_{rz}^n \text{curl}_{rz}^n u - \text{grad}_{rz}^n \text{div}_{rz}^n u = f,
\quad u_{rz} \cdot \vec{r} = 0, \, u_\theta = 0, \, \text{div}_{rz}^n u = 0 \quad \text{on} \, \Gamma_1.
$$

(3.5)

$k = 3$: The Dirichlet Problem for the Axisymmetric Poisson Equation

$$
-\text{div}_{rz}^n \text{grad}_{rz}^n u = f \quad \text{in} \, \Omega,
\quad u = 0 \quad \text{on} \, \Gamma_1.
$$

(3.6)

4 Implementation of Finite Element Methods for Hodge Laplacian Problems

We will present the discrete mixed formulation of each problem in this section and describe in detail how matrices and vectors are constructed and used in the FEM program for the $k = 1$ case.

4.1 Discrete Mixed Formulations

$k = 0$ Discrete Mixed Formulation:

Find $u_h \in A_{h_1}$ such that

$$
(\text{grad}_{rz}^n u_h, \text{grad}_{rz}^n v_h)_{L^2_\delta(\Omega)} = (F, v_h)_{L^2_\delta(\Omega)}, \quad \forall \, v_h \in A_{h_1}.
$$

(4.1)

$k = 1$ Discrete Mixed Formulation:
Find \((\sigma_h, u_h) \in A_{h_1} \times B_{h_1}\) such that
\[
(\sigma_h, \tau_h)_r - (\text{grad}^n r \sigma_h, u_h)_r = 0 \quad \forall \tau_h \in A_{h_1},
\]
\[
(\text{grad}^n r \sigma_h, v_h)_r + (\text{curl}^n r \sigma_h, v_h)_r = (F, v_h)_r \quad \forall v_h \in B_{h_1}.
\] (4.2)

\(k = 2\) Discrete Mixed Formulation:
Find \((\sigma_h, u_h) \in B_{h_1} \times C_{h_1}\) such that
\[
(\sigma_h, \tau_h)_r - (\text{curl}^n r \sigma_h, u_h)_r = 0 \quad \forall \tau_h \in B_{h_1},
\]
\[
(\text{curl}^n r \sigma_h, v_h)_r + (\text{div}^n r \sigma_h, v_h)_r = (F, v_h)_r \quad \forall v_h \in C_{h_1}.
\] (4.3)

\(k = 3\) Discrete Mixed Formulation:
Find \((\sigma_h, u_h) \in C_{h_1} \times D_{h_1}\) such that
\[
(\sigma_h, \tau_h)_r - (u_h, \text{div}^n r \tau_h)_r = 0 \quad \forall \tau_h \in C_{h_1},
\]
\[
(\text{div}^n r \sigma_h, v_h)_r = (F, v_h)_r \quad \forall v_h \in D_{h_1}.
\] (4.4)

4.2 Implementation of the \(k = 1\) Case

In this subsection, we focus on the \(k = 1\) case to provide the details of implementation. The other cases are programmed in a similar way.

Let \(N_1 = N + N_m\) and \(N_2 = 2N_1 + 2N_t + N + N_m\). Then, let us use \(\{\psi^n_i\}_{i=1}^{N_1}\) to denote the basis of \(A_{h_1}\) as defined in Eq. (2.10) and \(\{\zeta^n_m\}_{m=1}^{N_2}\) to denote the basis of \(B_{h_1}\) as defined in Eq. (2.12). Then,
\[
\sigma_h = \sum_{i=1}^{N_1} a_i \psi^n_i,
\]
\[
u_h = \sum_{j=1}^{N_2} b_j \zeta^n_j,
\]
and Eq. (4.2) can be rewritten as,
\[
(\sum_{i=1}^{N_1} a_i \psi^n_i, \psi^n_i)_r = (\text{grad}^n r \psi^n_i, \sum_{j=1}^{N_2} b_j \zeta^n_j)_r \quad \forall \psi^n_i \in \{\psi^n_i\}_{i=1}^{N_1},
\]
\[
(\sum_{i=1}^{N_1} a_i \text{grad}^n r \psi^n_i, \zeta^n_m)_r + (\sum_{j=1}^{N_2} b_j \text{curl}^n r \zeta^n_m, \text{curl}^n r \zeta^n_m)_r = (F, \zeta^n_m)_r \quad \forall \zeta^n_m \in \{\zeta^n_m\}_{m=1}^{N_2}.
\]
The above system of equations can again be rewritten as,
\[
\sum_{i=1}^{N_1} (\psi^n_i, \psi^n_i)_r, a_i = \sum_{j=1}^{N_2} (\text{grad}^n r \psi^n_i, \zeta^n_j)_r, b_j \quad \forall \psi^n_i \in \{\psi^n_i\}_{i=1}^{N_1},
\]
\[
\sum_{i=1}^{N_1} (\text{grad}^n r \psi^n_i, \zeta^n_m)_r, a_i + \sum_{j=1}^{N_2} (\text{curl}^n r \zeta^n_j, \text{curl}^n r \zeta^n_m)_r, b_j = (F, \zeta^n_m)_r \quad \forall \zeta^n_m \in \{\zeta^n_m\}_{m=1}^{N_2}.
\]
Let \(\bar{\sigma} = \sum_{i=1}^{N_1} a_i\) and \(\bar{u} = \sum_{j=1}^{N_2} b_j\). Then, we define the following vectors and matrices
\[
M(i, l) = (\psi^n_i, \psi^n_l)_r, \quad M \in \mathbb{R}^{(N_1 \times N_1)},
\]
\[
B(l, j) = (\text{grad}^n r \psi^n_l, \zeta^n_j)_r, \quad B \in \mathbb{R}^{(N_1 \times N_2)},
\]
\[
S(j, m) = (\text{curl}^n r \zeta^n_j, \text{curl}^n r \zeta^n_m)_r, \quad S \in \mathbb{R}^{(N_2 \times N_2)},
\]
\[
\bar{F}(m) = (\bar{F}, \zeta^n_m)_r, \quad F \in \mathbb{R}^{(N_2 \times 1)}.
\] (4.5)

Then, the system of equations reduces to
\[
M \bar{\sigma} = B \bar{u},
\]
\[
B^T \bar{\sigma} + S \bar{u} = \bar{F}.
\] (4.6)
By solving, we get
\[ \vec{u} = (B^T M^{-1} B + S)^{-1} \vec{F}, \]
\[ \vec{\sigma} = M^{-1} B \vec{u}. \]

Note that since M is symmetric positive definite, it is an invertible matrix.

5 Numerical Results

We will present numerical results of the weighted Hodge Laplacian on axisymmetric domains in this section. We will compare the results for the lowest and first order Fourier FEMs for each of \( k = 0, 1, 2, 3 \).

For each problem with a known exact solution, we define our domain as the unit square in \( \mathbb{R}^2 \) such that the vertices of \( \Omega \) are \((0, 0), (1, 0), (1, 1), (0, 1)\).

Note that in the error charts throughout this section, the order of convergence, also referred to as the rate in this paper, is calculated by taking \( \log_2(\frac{\text{error}_k}{\text{error}_{k+1}}) \) where \( k \) is the mesh level. Generally, as can be observed in the error charts presented, the rate is a positive number since the error reduces to zero as the mesh level increases. However, if the mesh level \( k + 1 \) error is larger than the mesh level \( k \) error, then the rate is a negative number. On lower level meshes, like mesh levels 1 and 2, the efficiency of the method may not yet be demonstrated and negative rates may be observed in these coarse meshes.

5.1 The \( k = 0 \) Case

The problem written below has the same left-hand side as the weighted Hodge Laplacian with \( k = 0 \), but has a modified right-hand side where \( Q_h : H_r(\text{grad}^n, \Omega) \to A_{h_1} \) is a projection that satisfies
\[ (\text{grad}^n_{rz} Q_h u, \text{grad}^n_{rz} v_h)_{L^2_r(\Omega)} = (\text{grad}^n_{rz} u, \text{grad}^n_{rz} v_h)_{L^2_r(\Omega)} \quad \forall \ v_h \in A_{h_1}. \] (5.1)

We define \( u \) as \( u = r \sin(z) \).

<table>
<thead>
<tr>
<th>mesh level</th>
<th>Lowest Order</th>
<th>Rate</th>
<th>First Order</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 )</td>
<td>8.47e-03</td>
<td>1.64e-03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 2 )</td>
<td>2.76e-03</td>
<td>1.62</td>
<td>2.54e-04</td>
<td>2.69</td>
</tr>
<tr>
<td>( 3 )</td>
<td>7.52e-04</td>
<td>1.88</td>
<td>3.51e-05</td>
<td>2.86</td>
</tr>
<tr>
<td>( 4 )</td>
<td>1.99e-04</td>
<td>1.92</td>
<td>4.59e-06</td>
<td>2.93</td>
</tr>
<tr>
<td>( 5 )</td>
<td>5.10e-05</td>
<td>1.96</td>
<td>5.86e-07</td>
<td>2.97</td>
</tr>
<tr>
<td>( 6 )</td>
<td>1.29e-05</td>
<td>1.99</td>
<td>7.41e-08</td>
<td>2.98</td>
</tr>
<tr>
<td>( 7 )</td>
<td>3.23e-06</td>
<td>1.99</td>
<td>9.31e-09</td>
<td>2.99</td>
</tr>
<tr>
<td>( 8 )</td>
<td>8.09e-07</td>
<td>2.00</td>
<td>1.17e-09</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Table 5.1 shows the \( L^2_r(\Omega) \)-norm of the errors between the exact solution \( u \) and the approximated solution \( Q_h u \) computed by both the lowest and first order programs.
Table 5.2: $k = 0$ case approximate solution vector lengths

<table>
<thead>
<tr>
<th>mesh level</th>
<th>Lowest Order</th>
<th>First Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>289</td>
</tr>
<tr>
<td>5</td>
<td>289</td>
<td>1089</td>
</tr>
<tr>
<td>6</td>
<td>1089</td>
<td>4225</td>
</tr>
<tr>
<td>7</td>
<td>4225</td>
<td>16641</td>
</tr>
<tr>
<td>8</td>
<td>16641</td>
<td>66049</td>
</tr>
</tbody>
</table>

Table 5.2 shows the length of the solution vector $\mathbf{x}$ representing the approximation for $Q_h u$ in each mesh level. Note that in this problem, the length of the solution vector $\mathbf{x}$ is the same as the length of the vector $\mathbf{b}$ and dimensions of the FEM matrix, $S$.

First of all, the order of convergence for the first order program is 3, while that of the lowest order program is 2. As a result, the 5th mesh of the lowest order program and 3rd mesh of the first order program have similar $L^2(\Omega)$-norm error measurements of $5.10 \times 10^{-05}$ and $3.51 \times 10^{-05}$. However, the first order program has an FEM matrix of dimension $81 \times 81$ in the 3rd mesh, whereas the lowest order program has an FEM matrix of dimension $289 \times 289$ in the 5th mesh. This shows that the first order program approximates $u$ with more accuracy than the lowest order program and while using a much smaller matrix. Furthermore, comparing the $L^2(\Omega)$-norm error measurements of the 5th mesh for each program reveals that the first order program yields an approximation with two digits of accuracy more than the lowest order program.

5.2 The $k = 1$ Case

For the $k = 1$ case we define $f$ such that the exact solution $(\sigma, u)$ is

\[
\sigma = -5r^3 + 4r^2 - (2z - 1)\left(\frac{1}{3}r^3 - \frac{1}{2}r^2\right),
\]

\[
u = \left[r^4 - r^3, 0, (z^2 - z)\left(\frac{1}{3}r^3 - \frac{1}{2}r^2\right)\right]^T.
\]

Recall from Eq. (3.4) that $f = -\text{grad}_r \text{div}_r u + \text{curl}_r u.$

Table 5.3: $k = 1$ case with Fourier mode $n = 2$

| mesh level | $||\sigma - \sigma_h||_{L^2(\Omega)}$ rate | $||u - u_h||_{L^2(\Omega)}$ rate | $||\sigma - \sigma_h||_{L^2(\Omega)}$ rate | $||u - u_h||_{L^2(\Omega)}$ rate |
|------------|----------------------------------------|---------------------------------|----------------------------------------|---------------------------------|
| 1          | 1.15e-01                               | 3.42e-02                        | 2.26e-03                               | 1.64e-02                        |
| 2          | 6.51e-02                               | 0.82                            | 6.49e-04                               | 9.96e-03                        | 0.72                             |
| 3          | 1.95e-02                               | 1.74                            | 9.47e-05                               | 3.19e-03                        | 1.64                             |
| 4          | 5.16e-03                               | 1.92                            | 1.24e-05                               | 8.53e-04                        | 1.90                             |
| 5          | 1.31e-03                               | 1.97                            | 1.59e-06                               | 2.18e-04                        | 1.97                             |
| 6          | 3.30e-04                               | 1.99                            | 2.01e-07                               | 5.50e-05                        | 1.99                             |

Table 5.3 depicts the $L^2(\Omega)$-norm of the errors between the exact solution $(\sigma, u)$ and the approximated solution $(\sigma_h, u_h)$ for the $k = 1$ case with $n = 2$. The $\sigma$- and $u$-errors
are bounded by a constant multiple of the best approximation error in the corresponding finite element spaces. The difference in the order of convergences for $\sigma$ and $u$ also follows by the best approximation error properties of each finite element space.

Table 5.4: $k = 1$ case approximate solution vector lengths

<table>
<thead>
<tr>
<th>mesh level</th>
<th>length of solution vector $\vec{\mathbf{\sigma}}$</th>
<th>length of solution vector $\vec{\mathbf{u}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lowest Order: 4</td>
<td>First Order: 9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>289</td>
</tr>
<tr>
<td>5</td>
<td>289</td>
<td>1089</td>
</tr>
<tr>
<td>6</td>
<td>1089</td>
<td>4225</td>
</tr>
</tbody>
</table>

Table 5.4 shows the lengths of the solution vectors $\vec{\mathbf{\sigma}}$ and $\vec{\mathbf{u}}$ representing $\sigma_h$ and $u_h$ in each mesh level.

By looking at table 5.3, we can easily see that the first order program has a higher order of convergence for both parts of the solution. As a result, the $L^2_2(\Omega)$-norm of the error for $u_h$ in the 2nd mesh of the first order program and the 4th mesh of the lowest order program are the both $9.96e-03$. Additionally, the $L^2_2(\Omega)$-norm of the error for $\sigma_h$ in the 3rd mesh for the first program, $9.47e-05$, is smaller than that of the 6th mesh in the lowest order program, $3.30e-04$.

In addition to comparing the errors, we can compare the size of the vectors in each mesh level relative to the error produced. In the 4th mesh of the first order program the $\vec{\mathbf{\sigma}}$ vector is $289\times 1$ and the $\vec{\mathbf{u}}$ vector is $961\times 1$, while in the 5th mesh of the lowest order program, they are $289\times 1$ and $1089\times 1$, respectively. The $L^2_2(\Omega)$-norm of the errors for $\sigma_h$ for these meshes are $1.24e-05$ for the first order program and $1.31e-03$ for the lowest order program. These numbers indicate that the approximated solution $\sigma_h$ from the first order program is accurate to the 5th decimal place, while the lowest order program only produces accuracy to the 3rd decimal place. This comparison indicates that the first order program produces far more accurate results, specifically two digits of accuracy more, with the same $\vec{\mathbf{\sigma}}$ vector length. In fact, if we look at the 2nd mesh of the first order program, where the $\vec{\mathbf{\sigma}}$ vector is $25\times 1$ and the $\vec{\mathbf{u}}$ vector is $73\times 1$, the $L^2_2(\Omega)$-norm of the error for $\sigma_h$ is $6.49e-04$. With both vectors more than a tenth of the length smaller than in the 5th mesh of the lowest order program, the first order program still produces an approximated solution for $\sigma$ that is accurate to an additional decimal point than the lowest order program. These results show that the first order program produces more accurate approximations with less computation and a faster rate of convergence.

5.3 The $k = 2$ Case

For the $k = 2$ case we define $f$ such that the exact solution $(\sigma, u)$ is

$$\sigma = \begin{bmatrix} nr(r - 1), & -3r^2 + 2r, & 0 \end{bmatrix}^T,$$
$$u = \begin{bmatrix} 0, & 0, & r^2(r - 1) \end{bmatrix}^T.$$
As defined in Eq. (3.5), \( f = \text{curl}_r \text{curl}_r^* u - \text{grad}_r^* \text{div}_r u. \)

### Table 5.5: \( k = 2 \) case with Fourier mode \( n = 10 \)

<table>
<thead>
<tr>
<th>mesh level</th>
<th>Lowest Order</th>
<th>First Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9.46e-01</td>
<td>4.72e-02</td>
</tr>
<tr>
<td>2</td>
<td>6.13e-01</td>
<td>3.06e-02</td>
</tr>
<tr>
<td>3</td>
<td>3.32e-01</td>
<td>2.02e-02</td>
</tr>
<tr>
<td>4</td>
<td>1.72e-01</td>
<td>1.10e-02</td>
</tr>
<tr>
<td>5</td>
<td>8.70e-02</td>
<td>5.64e-03</td>
</tr>
<tr>
<td>6</td>
<td>4.37e-02</td>
<td>2.84e-03</td>
</tr>
</tbody>
</table>

Table 5.5 shows the \( L^2_r(\Omega) \)-norm of the errors for the approximated solution \((\sigma_h, u_h)\) in comparison to the exact solution \((\sigma, u)\) for the \( k = 2 \) case with Fourier mode \( n = 10 \).

### Table 5.6: \( k = 2 \) case approximate solution vector lengths

<table>
<thead>
<tr>
<th>mesh level</th>
<th>length of solution vector ( \vec{\sigma} )</th>
<th>length of solution vector ( \vec{u} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest Order</td>
<td>First Order</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>73</td>
</tr>
<tr>
<td>3</td>
<td>81</td>
<td>257</td>
</tr>
<tr>
<td>4</td>
<td>289</td>
<td>961</td>
</tr>
<tr>
<td>5</td>
<td>1089</td>
<td>3713</td>
</tr>
<tr>
<td>6</td>
<td>4225</td>
<td>14593</td>
</tr>
</tbody>
</table>

In table 5.6, we show the lengths of the solution vectors \( \vec{\sigma} \) and \( \vec{u} \) representing \( \sigma_h \) and \( u_h \) in each mesh level.

By comparing the \( L^2_r(\Omega) \)-norm of the errors for \( u_h \) between the lowest and first order, we see that the 3\textsuperscript{rd} mesh of the first order program produces a smaller error of 3.70e–03 when compared to 5.64e–03 in 5\textsuperscript{th} mesh of the lowest order program. Additionally, the \( L^2_r(\Omega) \)-norm of the error for \( \sigma_h \) is 1.37e–02 for the first order program and 8.70e–02 for the lowest order program. These both indicate that for lower mesh levels, the first order program produces a smaller error for both parts of the solution. Furthermore, by comparing the order of convergence for the two parts of the solution in each program, we see that the first order program has a higher order of convergence, indicating that it decreases the error between mesh levels at a faster rate than the lowest order program.

We also compare the size of vectors in each mesh level of each program which has a direct correlation to the number of computations required. In the 4\textsuperscript{th} mesh of the first order program, the \( \vec{\sigma} \) vector is 961×1 and the \( \vec{u} \) vector is 753×1 and in the 5\textsuperscript{th} mesh of the lowest order program they are 1089×1 and 1312×1. By examining the table, we can see that the smaller vectors in the first order program produce errors with one more digit of accuracy than those in the lowest order program for both parts of the solution. Furthermore, with even smaller vectors of lengths 257 and 201 in the 3\textsuperscript{rd} mesh, the first order program still produces errors that are smaller for both parts of the solution when compared to the 5\textsuperscript{th} mesh of the lowest order program. These results indicate that the first
order program approximates solutions with more accuracy and efficiency when compared to the lowest order program.

Table 5.7: $k = 2$ case with Fourier modes $n = 1, 2, 3, 4, 5$

| mesh level | \(|\sigma - \sigma_h|_{L^2(\Omega)}\) | rate | \(|\sigma - \sigma_h|_{L^2(\Omega)}\) | rate | \(|\sigma - \sigma_h|_{L^2(\Omega)}\) | rate | \(|\sigma - \sigma_h|_{L^2(\Omega)}\) | rate |
|------------|---------------------------------|------|---------------------------------|------|---------------------------------|------|---------------------------------|------|
| 1          | 3.66e-02                        |      | 2.40e-02                        |      | 1.75e-02                        |      | 1.45e-02                        |      |
| 2          | 2.90e-02                        | 0.31 | 2.08e-02                        | 0.11 | 1.83e-02                        | -0.07| 1.90e-02                        | -0.39|
| 3          | 1.21e-02                        | 0.25 | 8.88e-03                        | 0.13 | 7.11e-03                        | 0.25 | 7.94e-03                        | 0.26 |
| 4          | 4.57e-03                        | 0.41 | 3.33e-03                        | 0.42 | 2.80e-03                        | 1.46 | 2.76e-03                        | 1.52 |
| 5          | 1.66e-03                        | 0.46 | 1.98e-03                        | 1.48 | 9.66e-04                        | 1.53 | 9.12e-04                        | 1.60 |
| 6          | 5.92e-04                        | 0.49 | 4.20e-04                        | 1.50 | 3.30e-04                        | 1.55 | 2.98e-04                        | 1.62 |

Table 5.7 shows the order of convergence for $\sigma_h$ gradually increasing as $n$ increases from 1 to 5.

![Figure 5.1: Fourier mode n vs. $\sigma_h$ order of convergence for the $k = 2$ case](image)

Figure 5.1 shows the relationship between the $n^{th}$ Fourier mode and the order of convergence for $\sigma_h$ from mesh level 6 in the given problem. We observe that the order of convergence approaches 1.8 as $n$ increases from 1 to 20, then begins to fall as $n$ continues to increase. The convergence analysis related to the Fourier mode $n$ remains as future work.

5.4 The $k = 3$ Case

For the $k = 3$ case we define $f$ such that the exact solution $(\sigma, u)$ is

$$\sigma = \left[ -\sin(\pi z)(2r - 1), -n \sin(\pi z)(r - 1), -\pi \cos(\pi z)(r^2 - r) \right]^T,$$

$$u = \sin(\pi z)(r^2 - r),$$

where $f = -\text{div}_r \text{grad}_{r^2} u$ as defined in Eq. (3.6).
Table 5.8: $k = 3$ case with Fourier mode $n = 5$

<table>
<thead>
<tr>
<th>mesh level</th>
<th>Lowest Order</th>
<th>First Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rate</td>
<td>$</td>
</tr>
<tr>
<td>1</td>
<td>$1.16\times10^0$</td>
<td>$9.16\times10^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>$7.54\times10^{-1}$</td>
<td>$6.39\times10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$4.57\times10^{-1}$</td>
<td>$2.47\times10^{-2}$</td>
</tr>
<tr>
<td>4</td>
<td>$2.37\times10^{-1}$</td>
<td>$1.21\times10^{-2}$</td>
</tr>
<tr>
<td>5</td>
<td>$1.26\times10^{-1}$</td>
<td>$6.01\times10^{-3}$</td>
</tr>
<tr>
<td>6</td>
<td>$6.65\times10^{-2}$</td>
<td>$3.00\times10^{-3}$</td>
</tr>
<tr>
<td>7</td>
<td>$3.48\times10^{-2}$</td>
<td>$1.50\times10^{-3}$</td>
</tr>
</tbody>
</table>

The $L^2_r(\Omega)$-norm of the errors of the approximated solution $(\sigma_h, u_h)$ compared to the exact solution $(\sigma, u)$ for the $k = 3$ case with Fourier mode $n = 5$ are shown in table 5.8.

Table 5.9: $k = 3$ case approximate solution vector lengths

<table>
<thead>
<tr>
<th>mesh level</th>
<th>length of solution vector $\vec{\sigma}$</th>
<th>length of solution vector $\vec{u}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest Order</td>
<td>First Order</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>57</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>201</td>
</tr>
<tr>
<td>4</td>
<td>336</td>
<td>753</td>
</tr>
<tr>
<td>5</td>
<td>1312</td>
<td>2913</td>
</tr>
<tr>
<td>6</td>
<td>5184</td>
<td>11457</td>
</tr>
<tr>
<td>7</td>
<td>20608</td>
<td>45441</td>
</tr>
</tbody>
</table>

Table 5.9 depicts the lengths of the solution vectors $\vec{\sigma}$ and $\vec{u}$ representing $\sigma_h$ and $u_h$ for the approximated solutions in each mesh.

For this particular example, the order of convergence for the $L^2_r(\Omega)$-norm of the errors for $\sigma_h$ is 1 for both the lowest and first order program. However, the order of convergence for the $L^2_r(\Omega)$-norm of the errors for $u_h$ is 1 for the lowest order program and 2 for the first order program.

If we compare the $L^2_r(\Omega)$-norm of the errors for $u_h$ in the 5th mesh of the first order program and the 7th mesh of the lowest order program, we see that the error in the first order program, $5.13e-04$, is smaller than the error in the lowest order program, $1.50e-03$. In the 7th mesh of the lowest order program, the $\vec{\sigma}$ vector is 20608×1 and the $\vec{u}$ vector is 8192×1. In comparison, in the 5th mesh of the first order program the vectors are 2913×1 and 289×1, respectively. The much smaller vectors in the first order program are more efficient to build and produce smaller errors for the approximation of $u$. These results show that the first order program is more efficient and produces better approximations than the lowest order program.

Finally, we present the results of the $k = 3$ case with an unknown exact solution and $f = 2$.

For this problem, we define an L-shaped domain in $\mathbb{R}^2$ such that the vertices of $\Omega$ are $(0, 0), (1, 0), (1, 0.5), (0.5, 0.5), (0.5, 1), (0, 1)$. 529
Table 5.10: $k = 3$ case with no known exact solution and $n = 3$

| mesh level $m$ | $||u_{m-1} - u_m||_{L^2(\Omega)}$ rate | $||u_{m-1} - u_m||_{H^1(\Omega)}$ rate |
|---------------|----------------------------------------|----------------------------------------|
| 2             | 3.81e-03                               | 4.97e-02                               |
| 3             | 1.63e-03                               | 4.22e-02                               | 0.24 |
| 4             | 5.20e-04                               | 2.62e-02                               | 0.69 |
| 5             | 1.46e-04                               | 1.44e-02                               | 0.86 |
| 6             | 3.93e-05                               | 7.68e-03                               | 0.91 |
| 7             | 1.06e-05                               | 4.09e-03                               | 0.91 |

Table 5.10 contains the $L^2(\Omega)$-norm and the $H^1(\Omega)$-norm of the errors of the approximated solution $u_{m-1}$ compared to the approximated solution $u_m$ in the next mesh level, where $u_m$ denotes $u_h$ in the $m^{th}$ mesh level.

Figure 5.2: Fourier mode $n$ vs. $u_h$ order of convergence for the $k = 3$ case

Figure 5.2 shows the relationship between the $n^{th}$ Fourier mode and the order of convergence for $\sigma_h$ from mesh level 7 in the given problem.

6 Concluding Remarks

This research studied higher order Fourier FEMs for Hodge Laplacian problems on axisymmetric domains using a new family of Fourier finite element spaces. The new family of Fourier finite element spaces is extended from [17] and is constructed with higher order spaces. Through numerical examples, we have shown that higher order programs yield better results with less computation and more efficiency when compared to the lowest order programs.
Bibliography


