A novel iterative method for discrete Helmholtz decomposition

JaEun Ku

Department of Mathematics, Oklahoma State University, 401 Mathematical Sciences, Stillwater, OK 74078, USA.

Lothar Reichel

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

Abstract

A new iterative method for the computation of the discrete Helmholtz decomposition of a vector is presented. We are particularly interested in computing the discrete Helmholtz decomposition when the given vector is discretized by a mixed finite element method defined by Raviart-Thomas (RT) or Brezzi-Douglas-Marini (BDM) elements. The decomposition is computed by solving a system of linear equations by an iterative method, that splits a given vector into a divergence-free component and a curl-free component. Each iteration cycle uses a well-developed solver based on the algebraic multigrid method for computing a projection onto $H(\text{div})$ or $H(\text{curl})$. Only a few iteration cycles are required to compute an accurate approximate solution. As a by-product, we obtain an iterative method for the solution of linear systems of equations with a nearly singular matrix.

Keywords: finite element method, nearly singular system, variational problem

AMS classification: Primary 65F10; Secondary 65N22

1. Introduction

The Helmholtz decomposition splits a vector field into a curl-free component and a divergence-free component. This decomposition is useful, for

Email addresses: `jku@okstate.edu` (JaEun Ku), `reichel@math.kent.edu` (Lothar Reichel)

instance, when modeling, analyzing, and manipulating fluids, and is applied in visualization, computer graphics, astrophysics, and imaging; see, e.g., [3] and references therein.

The finite element method (FEM) is a widely used technique for approximating the solution of boundary and initial value problems for partial differential equations (PDEs). Mixed finite element spaces, such as spaces made up of Raviart-Thomas (RT) or Brezzi-Douglas-Marini (BDM) elements, see [7, 8, 22], are commonly used as approximation spaces when solving incompressible fluid and electromagnetic problems. To the best of our knowledge, there is no efficient algorithm for computing the discrete Helmholtz decomposition for mixed element spaces. This decomposition is described in, e.g., [1, 9].

The aim of this paper is to present a new iterative method for computing the discrete Helmholtz decomposition of a given vector in a mixed finite element space. Our iterative method has a simple structure, which makes it easy to implement. In each iteration step, the method uses an algebraic multigrid method (AMG) to determine the projection of a given vector onto divergence-free or curl-free components. Typically only few iteration steps are required. The computed components are essentially orthogonal. This is an important property of the discrete Helmholtz decomposition. Our iterative method is based on a scheme for computing the solution of a linear system of equations with a nearly singular matrix. This scheme is an extension of the method described in [17].

This paper is organized as follows. Section 2 introduces necessary notation and describes the background of our decomposition method. The iterative method is presented in Section 3, where also its convergence properties are studied. Section 4 discusses iterative solution of nearly singular systems. Numerical examples are presented in Section 5, and concluding remarks can be found in Section 6.

2. Problem formulation

This section discusses the Helmholtz decomposition and introduces notation to be used subsequently.

2.1. The Helmholtz decomposition

Let $\mathbf{V}_h$ be a discrete subspace of $(L_2(\Omega))^n$, where $\Omega \in \mathbb{R}^n$ is a bounded simply connected polygonal domain with boundary $\partial\Omega$. Our discussion is for $n = 2$ space-dimensions; however, the method generalizes in a straightforward manner to $n = 3$ space-dimensions. We are interested in the situation when $\mathbf{V}_h$ is a mixed finite element space, such as a Raviart-Thomas (RT) space or a Brezzi-Douglas-Marini (BDM) space; see [22] and [7, 8]. Our analysis is presented for RT spaces, but can easily be extended to BDM spaces. The Helmholtz decomposition is based on the fact that a vector $\mathbf{f} \in \mathbf{V}_h$ can be split into divergence-free and curl-free components; see [12].

2.2. Discrete spaces

Let $\mathcal{T}_h$ be a quasi-uniform family of triangulations of Ω, where $h > 0$ is a parameter representative of the diameter of the triangles; see [4]. We denote the triangles of $\mathcal{T}_h$ by T. For each nonnegative integer r, the Raviart-Thomas space of index r is given by

$$\mathbf{V}_h = \{\mathbf{v} \in H(\text{div}) : \mathbf{v}|_T \in P_r(T) + (x,y)P_r(T) \text{ for all } T \in \mathcal{T}_h\}. \tag{2.1}$$

Here $P_r(T)$ denotes the set of polynomials of degree at most r on T, and $H(\text{div}) = \{\boldsymbol{\sigma} \in (L_2(\Omega))^2 : \nabla \cdot \boldsymbol{\sigma} \in L_2(\Omega)\}$. We refer to [2] and references therein for more detailed discussions on properties and implementations of RT spaces.

2.3. The Helmholtz decomposition of $\mathbf{V}_h$

Introduce the space

$$W_h = \{s \in H^1 : s|_T \in P_{r+1}(T)\}$$

of continuous piece-wise polynomials of degree at most $r+1$ on T with a derivative, as well as the space

$$S_h = \{q \in L_2 : q|_T \in P_r(T)\}$$

of (possibly discontinuous) piece-wise polynomials of degree at most r on T. Define the discrete gradient operator $\mathbf{grad}_h : S_h \to \mathbf{V}_h$ by

$$(\mathbf{grad}_h q, \mathbf{v}) = -(q, \nabla \cdot \mathbf{v}) \quad \forall\, \mathbf{v} \in \mathbf{V}_h.$$

The discrete Helmholtz decomposition of $\mathbf{V}_h$ is given by

$$\mathbf{V}_h = \mathbf{grad}_h S_h \oplus \mathbf{curl}\, W_h,$$

where $\mathbf{curl} = (-\frac{\partial}{\partial y}, \frac{\partial}{\partial x})^T$; cf. [1, 9]. Hence, a vector $\mathbf{f} \in \mathbf{V}_h$ can be decomposed according to

$$\mathbf{f} = \mathbf{f}_{\text{div}} + \mathbf{f}_{\text{curl}}, \tag{2.2}$$

where $\mathbf{f}_{\text{div}} \in \mathbf{grad}_h S_h$ and $\mathbf{f}_{\text{curl}} \in \mathbf{curl}\, W_h$. This decomposition is orthogonal with respect to both standard L_2 and $H(\text{div})$ inner products.

2.4. A variational problem

Our approach to computing the Helmholtz decomposition of a vector is based on the equation

$$(\nabla \cdot \boldsymbol{\sigma}, \nabla \cdot \boldsymbol{\tau}) + \delta(\boldsymbol{\sigma}, \boldsymbol{\tau}) = (\mathbf{f}, \boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathbf{V}_h. \tag{2.3}$$

The above variational problem may be thought of as a realization of a partial differential equation with the natural boundary condition, i.e., $\nabla \cdot \boldsymbol{\sigma} = 0$ on $\partial\Omega$; see [1]. Note that the second term in the left-hand side of (2.3) with the parameter $\delta > 0$ makes the operator in the left-hand side positive definite. As a result, the existence and uniqueness of the solution $\boldsymbol{\sigma} \in \mathbf{V}_h$ of (2.3) is guaranteed. Also, this term allows us to compute the components $\mathbf{f}_{\text{div}}$ and $\mathbf{f}_{\text{curl}}$ of $\mathbf{f}$, satisfying $\mathbf{f}_{\text{div}} \in \mathbf{grad}_h S_h$ and $\mathbf{f}_{\text{curl}} \in \mathbf{curl}\, W_h$, by using an iterative method and taking advantage of the different convergence behaviors of these components. We can observe in computed examples reported in Section 5 that a smaller $\delta > 0$ gives slightly more accurate approximations of $\mathbf{f}_{\text{div}}$ and $\mathbf{f}_{\text{curl}}$, while also slightly increasing the number of iterations.

Remark 2.1. *One can develop a method for computing the Helmholtz decomposition based on the variational problem*

$$(\mathbf{curl}\, \boldsymbol{\sigma}, \mathbf{curl}\, \boldsymbol{\tau}) + \delta(\boldsymbol{\sigma}, \boldsymbol{\tau}) = (\mathbf{f}, \boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathbf{V}_h.$$

The analysis of such a method is almost identical to the analysis of our method, which is based on (2.3).

2.5. Two linear operators

Define the linear operators A and A_δ by

$$\begin{aligned} (A\boldsymbol{\sigma}, \boldsymbol{\tau}) &= (\nabla \cdot \boldsymbol{\sigma}, \nabla \cdot \boldsymbol{\tau}) + (\boldsymbol{\sigma}, \boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathbf{V}_h, \\ (A_\delta\boldsymbol{\sigma}, \boldsymbol{\tau}) &= (\nabla \cdot \boldsymbol{\sigma}, \nabla \cdot \boldsymbol{\tau}) + \delta(\boldsymbol{\sigma}, \boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathbf{V}_h. \end{aligned}$$

The operator A is symmetric and positive definite. Its smallest eigenvalue is larger than or equal to 1. Hence, all eigenvalues of A^{-1} lie in the semi-open interval $(0, 1]$. The operator A_δ can be expressed as

$$A_\delta = A + (\delta - 1)I. \tag{2.4}$$

This shows that A_δ also is symmetric and positive definite for $\delta > 0$, with its eigenvalues bounded below by δ.

Using orthogonality, one can show that the two summand spaces $\mathbf{grad}_h S_h$ and $\mathbf{curl}\, W_h$ are invariant under A and A_δ. Moreover, we have

$$A\boldsymbol{\psi} = \boldsymbol{\psi} \;\text{ and }\; A_\delta\boldsymbol{\psi} = \delta\boldsymbol{\psi} \quad \forall \boldsymbol{\psi} \in \mathbf{curl}\, W_h. \tag{2.5}$$

2.6. A linear equation

We would like to compute an accurate approximation of the Helmholtz decomposition of a vector $\mathbf{f} \in \mathbf{V}_h$. To achieve this, we solve

$$A_\delta \boldsymbol{\sigma}_h = \mathbf{f} \tag{2.6}$$

by an iterative method. The method splits $\mathbf{f}$ into a divergence-free component $\mathbf{f}_{\text{curl}}$ and a curl-free component $\mathbf{f}_{\text{div}}$; cf. (2.2). To understand the splitting procedure, we consider (2.6) as two problems with data vectors $\mathbf{f}_{\text{curl}}$ and $\mathbf{f}_{\text{div}}$,

$$A_\delta(\boldsymbol{\sigma}_{\text{curl}}) = \mathbf{f}_{\text{curl}} \quad \text{and} \quad A_\delta(\boldsymbol{\sigma}_{\text{div}}) = \mathbf{f}_{\text{div}}. \tag{2.7}$$

These two subproblems cannot be solved separately, because we do not know the components $\mathbf{f}_{\text{curl}}$ and $\mathbf{f}_{\text{div}}$ of $\mathbf{f}$. Instead, we will apply our iterative method to the solution of (2.6) and use the fact that the components of the computed iterates associated with the solutions of the subproblems (2.7) converge with different rates to separate $\mathbf{f}$ into $\mathbf{f}_{\text{curl}}$ and $\mathbf{f}_{\text{div}}$.

Thus, we apply our iterative method to (2.6). The iterates generated are made up of a linear combination of approximations of $\boldsymbol{\sigma}_{\text{curl}}$ and $\boldsymbol{\sigma}_{\text{div}}$. The rates of convergence of these approximations towards $\boldsymbol{\sigma}_{\text{curl}}$ and $\boldsymbol{\sigma}_{\text{div}}$ differ. This makes it possible to separate them and in this manner determine the components $\mathbf{f}_{\text{curl}}$ and $\mathbf{f}_{\text{div}}$ of $\mathbf{f}$. In detail, using (2.5), we obtain

$$\boldsymbol{\sigma}_{\text{curl}} = \frac{\mathbf{f}_{\text{curl}}}{\delta} \quad \text{since} \quad A_\delta\left(\frac{\mathbf{f}_{\text{curl}}}{\delta}\right) = \mathbf{f}_{\text{curl}}.$$

Let $\boldsymbol{\sigma}_{\text{div}}$ satisfy

$$A_\delta(\boldsymbol{\sigma}_{\text{div}}) = \mathbf{f}_{\text{div}}.$$

Then

$$A_\delta\left(\boldsymbol{\sigma}_{\text{div}} + \frac{\mathbf{f}_{\text{curl}}}{\delta}\right) = \mathbf{f}_{\text{div}} + \mathbf{f}_{\text{curl}} = \mathbf{f}. \tag{2.8}$$

In our iterative method for the approximation of the solution of (2.8), the rates of convergence towards the components $\boldsymbol{\sigma}_{\text{div}}$ and $\frac{\mathbf{f}_{\text{curl}}}{\delta}$ of the solution differ and depend on the choice of the parameter $\delta > 0$. This is analyzed in Section 3, and it allows us to split $\mathbf{f}$ into $\mathbf{f}_{\text{curl}}$ and $\mathbf{f}_{\text{div}}$. To simplify our notation, we will in the remainder of this paper denote $\mathbf{f}_{\text{div}}$ and $\mathbf{f}_{\text{curl}}$ by $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively.

3. The iterative method

We first describe our iterative method, then analyze its convergence behavior, and finally discuss its application to the Helmholtz decomposition of a vector.

3.1. Derivation of the iterative method

The splitting (2.4) yields

$$A_\delta \boldsymbol{\sigma}_h = A\boldsymbol{\sigma}_h + (\delta - 1)\boldsymbol{\sigma}_h = \mathbf{f},$$

which suggests that equation (2.8) be solved with the iterative method

$$\boldsymbol{\sigma}_{n+1} = (1-\delta)A^{-1}\boldsymbol{\sigma}_n + A^{-1}\mathbf{f}. \tag{3.1}$$

Here and below the iteration number is indicated by an integer subscript of $\boldsymbol{\sigma}$.

Remark 3.1. *The convergence rate of the iterates (3.1) is very slow for small $\delta > 0$. In fact, the error reduction rate is $1-\delta$ due to the divergence-free component of $\mathbf{f}$ in $\mathbf{curl}\, W_h$. A convergence analysis is presented in the next subsection. Because of the slow convergence of the divergence-free component of the solution, a standard stopping criterion based on measuring $\|\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}\|$ is not appropriate. Here the norm $\|\cdot\|$ is induced by the standard inner product. Instead, we will use the quantities F_n defined in line 6 of Algorithm 3.1 to determine when to terminate the iterations.*

Algorithm 3.1 Iterative method

1: **Input:** tolerance TOL> 0 and given vector $\mathbf{f}$
2: **Output:** Helmholtz decomposition of $\mathbf{f}$ and approximate solution $\boldsymbol{\sigma}^*$
3: $\boldsymbol{\sigma}_0 := \mathbf{0}$
4: **for** $n = 0, 1, 2, \dots$
5:     $\boldsymbol{\sigma}_{n+1} := (1-\delta)A^{-1}\boldsymbol{\sigma}_n + A^{-1}\mathbf{f}$
6:     **if** $F_n := \|\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1} - \frac{1}{(1-\delta)}(\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n)\| < \text{TOL}$
7:         $\boldsymbol{\sigma}^* := \boldsymbol{\sigma}_n + \frac{(1-\delta)}{\delta}(\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1})$
8:         exit
9:     **end if**
10: **end for**

At exit from Algorithm 3.1, $\boldsymbol{\sigma}^*$ is an approximate solution of (2.6), and $\frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^n}$ is an approximation of $\mathbf{f}_2$. The Helmholtz decomposition of $\mathbf{f}$ is

$$\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2 = \left(\mathbf{f} - \frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^n}\right) + \frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^n}.$$

Remark 3.2. *The stopping criterion in Algorithm 3.1 based on the size of F_n is meaningful since it removes the slowly convergent component; see Lemma 3.2 below, and it is effective for computing the Helmholtz decomposition. However, if one instead is interested in determining an accurate approximate solution of (2.6), then a different stopping criterion should be used. For instance, one might terminate the iterations when $\|\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}\|$ is sufficiently small.*

Remark 3.3. *The application of A^{-1} in Algorithm 3.1 is carried out with an algebraic multigrad method. We use the public domain code provided by Notay [19]; see Section 5 for details. The algorithm therefore is simple to implement. Computed examples reported in Section 5 illustrate that only fairly few iteration steps of the algorithm are required. This makes the algorithm quite fast.*

3.2. Convergence analysis

Recall that

$$\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2, \quad \mathbf{f}_1 \in \mathbf{grad}_h S_h, \quad \mathbf{f}_2 \in \mathbf{curl}\, W_h, \tag{3.2}$$

and let $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_2$ denote the solutions of

$$A_\delta \boldsymbol{\sigma}_1 = \mathbf{f}_1 \quad \text{and} \quad A_\delta \boldsymbol{\sigma}_2 = \mathbf{f}_2.$$

Then, clearly, $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_2$, and it follows that

$$\begin{aligned} \boldsymbol{\sigma}_1 &= (1-\delta)A^{-1}\boldsymbol{\sigma}_1 + A^{-1}\mathbf{f}_1, \\ \boldsymbol{\sigma}_2 &= (1-\delta)A^{-1}\boldsymbol{\sigma}_2 + A^{-1}\mathbf{f}_2. \end{aligned}$$

These equations suggest that we decompose the iterates in (3.1) as

$$\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_n^1 + \boldsymbol{\sigma}_n^2, \tag{3.3}$$

where

$$\begin{aligned} \boldsymbol{\sigma}_{n+1}^1 &= (1-\delta)A^{-1}\boldsymbol{\sigma}_n^1 + A^{-1}\mathbf{f}_1, \\ \boldsymbol{\sigma}_{n+1}^2 &= (1-\delta)A^{-1}\boldsymbol{\sigma}_n^2 + A^{-1}\mathbf{f}_2. \end{aligned} \tag{3.4}$$

We have the following identities for $\boldsymbol{\sigma}_n^1$ and $\boldsymbol{\sigma}_n^2$.

Lemma 3.1. *Let the sequences $\{\boldsymbol{\sigma}_n^1\}_{n=0}^\infty$ and $\{\boldsymbol{\sigma}_n^2\}_{n=0}^\infty$ be defined by (3.4) with $\boldsymbol{\sigma}_0^1 = \mathbf{0}$ and $\boldsymbol{\sigma}_0^2 = \mathbf{0}$. Then*

$$\boldsymbol{\sigma}_n^1 = \sum_{k=1}^n (1-\delta)^{k-1} A^{-k}\mathbf{f}_1 \in \mathbf{grad}_h S_h \tag{3.5}$$

and

$$\boldsymbol{\sigma}_n^2 = \sum_{k=1}^{n} (1-\delta)^{k-1} \mathbf{f}_2 = \frac{1-(1-\delta)^n}{\delta} \mathbf{f}_2 \in \mathbf{curl}\, W_h. \tag{3.6}$$

Proof. Equation (3.5) is obtained from the first equation of (3.4) with $\boldsymbol{\sigma}_0^1 = 0$. Turning to (3.6), we use that $\mathbf{f}_2 \in \mathbf{curl}\, W_h$ and the relations (2.5) to obtain

$$\begin{aligned} \boldsymbol{\sigma}_n^2 &= (1-\delta)^{n-1}\mathbf{f}_2 + (1-\delta)^{n-2}\mathbf{f}_2 + \cdots + (1-\delta)\mathbf{f}_2 + \mathbf{f}_2 \\ &= \frac{1-(1-\delta)^n}{1-(1-\delta)} \mathbf{f}_2 = \frac{1-(1-\delta)^n}{\delta} \mathbf{f}_2. \end{aligned}$$

□

Let $\{(\lambda_i, \boldsymbol{\psi}_i)\}_{i=1}^m$ be the eigenpairs of A^{-1} on $\mathbf{grad}_h S_h$, where $\{\boldsymbol{\psi}_i\}_{i=1}^m$ is an orthonormal basis for $\mathbf{grad}_h S_h$. Without loss of generality, we may assume that λ_1 is the largest eigenvalue, where we recall that $0 < \lambda_i \leq 1$ for all $i = 1, 2, \ldots, m$. Then

$$\mathbf{f}_1 = \sum_{i=1}^{m} c_i \boldsymbol{\psi}_i \tag{3.7}$$

for certain coefficients c_i. Using (3.5), we obtain

$$\boldsymbol{\sigma}_n^1 = \sum_{k=1}^{n} \sum_{i=1}^{m} (1-\delta)^{k-1} \lambda_i^k c_i \boldsymbol{\psi}_i.$$

Convergence of the sequence $\boldsymbol{\sigma}_1^1, \boldsymbol{\sigma}_2^1, \ldots$ is fast due to the factors A^{-k} in (3.5), while convergence of the sequence $\boldsymbol{\sigma}_1^2, \boldsymbol{\sigma}_2^2, \ldots$ defined by (3.6) is slow when $\delta > 0$ is small. On the other hand, $\boldsymbol{\sigma}_n^2$ is a multiple of $\mathbf{f}_2$, and we can take advantage of this fact to terminate the iterative process based on the convergence of the sequence $\boldsymbol{\sigma}_1^1, \boldsymbol{\sigma}_2^1, \ldots$.

Lemma 3.2. *Let*

$$\mathbf{E}_n := \boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1} = (1-\delta)^{n-1} A^{-n} \mathbf{f}_1 + (1-\delta)^{n-1} \mathbf{f}_2$$

and let F_n be defined as in line 6 of Algorithm 3.1, i.e.,

$$F_n = \|\mathbf{E}_n - \frac{1}{(1-\delta)} \mathbf{E}_{n+1}\|.$$

Then

$$(1-\lambda_1)\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\| \leq F_n \leq \|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|, \tag{3.8}$$

where $0 < \lambda_1 \leq 1$ *is the largest eigenvalue of* A^{-1} *on* $\mathbf{grad}_h S_h$*. Moreover,*

$$\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\| = (1-\delta)^{n-1} \left(\sum_{i=1}^m \lambda_i^{2n} c_i^2 \right)^{1/2}. \tag{3.9}$$

Proof. Using (3.5), (3.6) with (3.3), we obtain

$$\mathbf{E}_n - \frac{1}{(1-\delta)} \mathbf{E}_{n+1} = (1-\delta)^{n-1} A^{-n} \mathbf{f}_1 - (1-\delta)^{n-1} A^{-n-1} \mathbf{f}_1.$$

Now, (3.7) and the fact that $\{(\lambda_i, \boldsymbol{\psi}_i)\}_{i=1}^m$ are eigenpairs of A^{-1} give

$$\mathbf{E}_n - \frac{1}{(1-\delta)} \mathbf{E}_{n+1} = (1-\delta)^{n-1} \sum_{i=1}^m (\lambda_i^n - \lambda_i^{n+1}) c_i \boldsymbol{\psi}_i = (1-\delta)^{n-1} \sum_{i=1}^m \lambda_i^n (1-\lambda_i) c_i \boldsymbol{\psi}_i.$$

Since λ_1 is the largest eigenvalue and the eigenfunctions $\{\boldsymbol{\psi}_i\}_{i=1}^m$ are orthonormal, we have

$$(1-\lambda_1)^2 (1-\delta)^{2(n-1)} \sum_{i=1}^m \lambda_i^{2n} c_i^2 \leq F_n^2 \leq (1-\delta)^{2(n-1)} \sum_{i=1}^m \lambda_i^{2n} c_i^2. \tag{3.10}$$

Using (3.5), the fact that $\{(\lambda_i, \boldsymbol{\psi}_i)\}_{i=1}^m$ are eigenpairs of A^{-1}, and (3.7) yield

$$\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1 = (1-\delta)^{n-1} \sum_{i=1}^m \lambda_i^n c_i \boldsymbol{\psi}_i.$$

Thus

$$\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|^2 = (1-\delta)^{2(n-1)} \sum_{i=1}^m \lambda_i^{2n} c_i^2. \tag{3.11}$$

Taking square roots on both sides, we obtain (3.9). Now, combining (3.11) with (3.10) gives (3.8). This completes the proof. □

If F_n is small, then $\boldsymbol{\sigma}_n^1$ is an accurate approximation of $\boldsymbol{\sigma}^1$. This was already shown in [17]. However, even when $\boldsymbol{\sigma}_n^1$ is an accurate approximation of $\boldsymbol{\sigma}^1$,

$$\boldsymbol{\sigma}_n^2 = \frac{1-(1-\delta)^n}{\delta} \mathbf{f}_2$$

may be a poor approximation of $\boldsymbol{\sigma}^2 = \frac{1}{\delta}\mathbf{f}_2$ when $\frac{(1-\delta)^n}{\delta}\mathbf{f}_2$ is large due to that $\delta > 0$ is small. We therefore define an approximate solution of (2.6) as in line 7 of Algorithm 3.1, i.e.,

$$\boldsymbol{\sigma}^* = \boldsymbol{\sigma}_n + \frac{(1-\delta)}{\delta} (\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}).$$

Define the projection of $\mathbf{f}$ onto the space $\mathbf{curl}\, W_h$ by

$$P_h\mathbf{f} := \frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^{n-1}}. \tag{3.12}$$

Theorem 3.3. *Let $\mathbf{f} \in \mathbf{V}_h$ have the Helmholtz decomposition $\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$, where $\mathbf{f}_1 \in \mathbf{grad}_h S_h$ and $\mathbf{f}_2 \in \mathbf{curl}\, W_h$. Let $\boldsymbol{\sigma} = \boldsymbol{\sigma}_h^1 + \frac{1}{\delta}\mathbf{f}_2$ be the solution of (2.6). Then*

$$\|\mathbf{f}_2 - P_h\mathbf{f}\| = \frac{\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|}{(1-\delta)^{n-1}} \leq \frac{\mathrm{TOL}}{(1-\lambda_1)(1-\delta)^{n-1}},$$

where the $\boldsymbol{\sigma}_n^1$ and TOL are defined by Algorithm 3.1. Also, let $\boldsymbol{\sigma}^*$ denote the approximation of $\boldsymbol{\sigma}$ defined in line 7 of Algorithm 3.1. Then

$$\|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\| \leq \|\boldsymbol{\sigma}_h^1 - \boldsymbol{\sigma}_n^1\| + \frac{1-\delta}{\delta}\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|.$$

Proof. We obtain from (3.6) that

$$\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_n^1 + \boldsymbol{\sigma}_n^2 = \boldsymbol{\sigma}_n^1 + \frac{1-(1-\delta)^n}{\delta}\mathbf{f}_2. \tag{3.13}$$

Again using (3.6), we have

$$\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1} = \boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1 + (1-\delta)^{n-1}\mathbf{f}_2. \tag{3.14}$$

Thus,

$$\mathbf{f}_2 - \frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^{n-1}} = -\frac{\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1}{(1-\delta)^{n-1}}. \tag{3.15}$$

Combining (3.15) with (3.9), (3.8) and $F_n < \mathrm{TOL}$ gives

$$\left\|\mathbf{f}_2 - \frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^{n-1}}\right\| = \left(\sum_{i=1}^m \lambda_i^{2n} c_i^2\right)^{1/2} \leq \frac{\mathrm{TOL}}{(1-\lambda_1)(1-\delta)^{n-1}}.$$

Using (3.13) and (3.14), we obtain

$$\frac{(1-\delta)}{\delta}(\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}) = \frac{(1-\delta)}{\delta}(\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1) + \frac{(1-\delta)^n}{\delta}\mathbf{f}_2. \tag{3.16}$$

Equations (3.13) and (3.16) show that

$$\boldsymbol{\sigma}^* = \boldsymbol{\sigma}_n + \frac{(1-\delta)}{\delta}(\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}) = \boldsymbol{\sigma}_n^1 + \frac{1}{\delta}\mathbf{f}_2 + \frac{(1-\delta)}{\delta}(\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1).$$

Thus

$$\|\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\| \leq \|\boldsymbol{\sigma}^1 - \boldsymbol{\sigma}_n^1\| + \frac{1-\delta}{\delta}\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|.$$

This completes the proof. □

Remark 3.4. *It follows from (3.16) that*

$$\frac{\boldsymbol{\sigma}_n - \boldsymbol{\sigma}_{n-1}}{(1-\delta)^{n-1}} = \mathbf{f}_2 + \frac{\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1}{(1-\delta)^{n-1}}. \tag{3.17}$$

Due to (3.11) and Lemma 3.2, the second term in the right-hand side is small since F_n in our stopping criterion is of about the same size as $\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\|$. The equality (3.17) therefore suggests the application of the projector (3.12) onto $\mathbf{curl}\, W_h$.

4. The solution of nearly singular systems of equations

We consider the solution of nearly singular linear systems of equations determined by the variational problem (2.3) with $0 < \delta \ll 1$ very small, and use Raviart-Thomas finite element spaces defined in Section 3 to approximate $\boldsymbol{\sigma} \in H(\text{div})$. Solution methods for linear systems of equations with a general singular or nearly singular matrix have received considerable attention in the literature; see, e.g., [10, 11, 13, 14, 15, 23]. The problem that we are considering has a structure that makes it possible to compute a useful approximate solution in a simple manner.

4.1. Approximation properties

Let $\mathbf{V} = H(\text{div})$. Define the subspace

$$Q_h^r = \{q \in L^2(\Omega) \,:\, q|_K \in P_r(K) \;\text{ for each }\; K \in \mathcal{T}_h\},$$

and let $LP_h : L_2(\Omega) \to Q_h^r$ be the local L_2 projection, i.e., for $K \in \mathcal{T}_h$,

$$(g - LP_h g, v_h)_K = \int_K (g - LP_h) \cdot v_h dx = 0 \quad \forall\, v_h \in Q_h^r.$$

The projector LP_h onto Q_h^r satisfies

$$\|v - LP_h(v)\| \le Ch^{r+1}\|v\|_{H^{r+1}(\Omega)}$$

for all $v \in H^{r+1}(\Omega)$; see [24].

Let the finite element space $Q_h^r \times \mathbf{V}_h$ be defined with respect to $\mathcal{T}_h$. We let $\mathbf{V}_h$ be the Raviart-Thomas space of index r; see (2.1) and [22]. The Raviart-Thomas projection operator $\Pi_h : \mathbf{V} \to \mathbf{V}_h$, discussed in [8], satisfies

$$\nabla \cdot \Pi_h \boldsymbol{\tau} = LP_h(\nabla \cdot \boldsymbol{\tau}) \quad \forall\, \boldsymbol{\tau} \in \mathbf{V}_h. \tag{4.1}$$

We have the following approximation property, see [8],

$$\|\boldsymbol{\tau} - \Pi_h \boldsymbol{\tau}\| \le Ch^{r+1}\|\boldsymbol{\tau}\|_{H^{r+1}(\Omega)} \quad \forall\, \boldsymbol{\tau} \in (H^{r+1}(\Omega))^n. \tag{4.2}$$

4.2. An error estimate

Define the approximate solution $\boldsymbol{\sigma}_h \in \mathbf{V}_h$ of (2.3) by

$$(\nabla \cdot \boldsymbol{\sigma}_h, \nabla \cdot \boldsymbol{\tau}_h) + \delta(\boldsymbol{\sigma}_h, \boldsymbol{\tau}_h) = (\mathbf{f}, \boldsymbol{\tau}_h) \quad \forall \boldsymbol{\tau}_h \in \mathbf{V}_h. \tag{4.3}$$

We have the following basic error estimate.

Theorem 4.1. *Let $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_h$ satisfy (2.3) and (4.3), respectively. Then*

$$\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\| \le \|\boldsymbol{\sigma} - \Pi_h \boldsymbol{\sigma}\|,$$

where $\Pi_h : \mathbf{V} \to \mathbf{V}_h$ is the RT projection operator satisfying (4.1) and (4.2).

Proof. Subtracting (4.3) from (2.3) gives

$$(\nabla \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h), \nabla \cdot \boldsymbol{\tau}_h) + \delta(\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \boldsymbol{\tau}_h) = 0. \tag{4.4}$$

Using (4.1), (4.4) and the Cauchy-Schwarz inequality, we get

$$\begin{aligned}
\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|^2 &= (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \boldsymbol{\sigma} - \boldsymbol{\sigma}_h) \\
&\le (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \boldsymbol{\sigma} - \boldsymbol{\sigma}_h) + \frac{1}{\delta}(\nabla \cdot (\Pi_h\boldsymbol{\sigma} - \boldsymbol{\sigma}_h), \nabla \cdot (\Pi_h\boldsymbol{\sigma} - \boldsymbol{\sigma}_h)) \\
&= (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \Pi_h\boldsymbol{\sigma} - \boldsymbol{\sigma}_h) + \frac{1}{\delta}(\nabla \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h), \nabla \cdot (\Pi_h\boldsymbol{\sigma} - \boldsymbol{\sigma}_h)) \\
&\qquad + (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \boldsymbol{\sigma} - \Pi_h\boldsymbol{\sigma}) \\
&= (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h, \boldsymbol{\sigma} - \Pi_h\boldsymbol{\sigma}) \\
&\le \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\| \|\boldsymbol{\sigma} - \Pi_h\boldsymbol{\sigma}\|,
\end{aligned}$$

which yields the desired inequality. □

4.3. An iterative method for nearly singular systems

When $\delta > 0$ is tiny, equation (2.6), or equivalently equation (2.3), are nearly singular. The vector defined in line 7 of Algorithm 3.1 typically is not an accurate approximation of the solution of (2.6) due to a large factor $\frac{1}{\delta}$. To overcome this difficulty, we combine Algorithm 3.1 with the iterative method for solving nearly singular linear systems developed in [17]. The iterative method described in [17] is applicable when $\mathbf{f} \in \mathbf{grad}_h S_h$. Here, we extend the method to be applicable when $\mathbf{f} \in \mathbf{V}_h \subset (L_2(\Omega))^2$.

Consider the Helmholtz decomposition (3.2). We apply Algorithm 3.1 to obtain an approximation $P_h\mathbf{f}$ of $\mathbf{f}_2$, and use $\mathbf{f}_1 = \mathbf{f} - P_h\mathbf{f}$ as an approximation of $\boldsymbol{\sigma}_1$. Note that the solution $\boldsymbol{\sigma}_2^h$ of (4.3) corresponding to $\mathbf{f}_2$ is $\frac{\mathbf{f}_2}{\delta}$, and the

Algorithm 4.1 Iterative method for nearly singular system

Input: tolerance TOL>0 and $\mathbf{f}_1$
Output: approximate solution $\boldsymbol{\sigma}_h^1$
$\boldsymbol{\sigma}_0^1 := \mathbf{0}$
for $n = 0, 1, 2, \ldots$
  $\boldsymbol{\sigma}_{n+1}^1 := (1-\delta)A^{-1}\boldsymbol{\sigma}_n^1 + A^{-1}\mathbf{f}_1$
  if $\|\boldsymbol{\sigma}_n^1 - \boldsymbol{\sigma}_{n-1}^1\| < \text{TOL}$
    $\boldsymbol{\sigma}_h^1 := \boldsymbol{\sigma}_n^1$
    exit
  end if
end for

solution $\boldsymbol{\sigma}_h^1$ of (4.3) corresponding to $\mathbf{f}_1$ can be determined by the iterative method described in [17]. The computations are summarized by Algorithm 4.1.

We define the solution of (4.3) as $\boldsymbol{\sigma}_1^h + \boldsymbol{\sigma}_2^h$. Computed examples reported in [17] show that the number of iterations with Algorithm 4.1 to achieve a desired accuracy can be reduced by vector extrapolation. We will not dwell on extrapolation in the present paper, and instead refer to [17] as well as to the references [5, 6, 16] for discussions on extrapolations methods.

5. Numerical examples

This section presents examples that illustrate the convergence behavior of the iterative scheme defined by Algorithms 3.1 and 4.1. We refer to [2] for details on the implementation of the FEM, such as the enumeration of edges and nodes.

Let $\Omega = [0,1]^2$ and discretize on a uniform mesh with mesh size h. Unless stated otherwise, $h = \frac{1}{32}$. We use the lowest order Raviart-Thomas element space, denoted by $\mathbf{V}_h$, to approximate the space $H(\text{div})$ in our experiments. Let $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N$ be an edge basis for $\mathbf{V}_h$, i.e.,

$$\mathbf{V}_h = \text{span}\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N\}.$$

For $h = \frac{1}{32}$, the total number of edges (which is the number of unknowns) is $N = 3136$. For a function $\mathbf{f} = \sum_{i=1}^N f_i \boldsymbol{\phi}_i \in H(\text{div})$, we let $\vec{\mathbf{f}} = [f_1, f_2, \ldots, f_N]^T$ denote its vector representation with respect to the basis $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N$.

The algebraic equation corresponding to line 7 of Algorithm 3.1 can be written as

$$\vec{\boldsymbol{\sigma}}_{n+1} = (1-\delta)S^{-1}B\vec{\boldsymbol{\sigma}}_n + S^{-1}B\vec{\mathbf{f}},$$

where

$$D = [d_{ij}], \quad d_{ij} = (\nabla \cdot \boldsymbol{\phi}_j, \nabla \cdot \boldsymbol{\phi}_i), \qquad B = [b_{ij}], \quad b_{ij} = (\boldsymbol{\phi}_j, \boldsymbol{\phi}_i),$$

and $S = D + B$; see [17] for more details. We use the AMG solver developed and made available by Notay [18, 19, 20, 21] for the computation of $S^{-1}\vec{\mathbf{y}}$ for a vector $\vec{\mathbf{y}} \in \mathbb{R}^N$.

Example 1: Helmholtz decomposition. We would like to compute the Helmholtz decomposition of the vector $\mathbf{f} \in (L_2(\Omega))^2$ defined by $\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$, where

$$\mathbf{f}_1 = -\mathbf{grad}_h(2(x-x^2)+(y-y^2)+\delta(x-x^2)(y-y^2)), \quad \mathbf{f}_2 = \boldsymbol{\phi}_1 - \boldsymbol{\phi}_{33} + \frac{1}{\sqrt{2}}\boldsymbol{\phi}_{34}.$$

Note that $\mathbf{f}_2 = \mathbf{curl}\,\zeta \in \mathbf{curl}\,W_h$, where ζ is the piece-wise linear polynomial with the value 1 at the vertex $(0,0)$ and the value 0 at the other vertices.

We compute the iterates $\boldsymbol{\sigma}_n$ by Algorithm 3.1 with TOL$= 10^{-20}$. The approximation $P_h\mathbf{f}$ of $\mathbf{f}_2$ is defined as in (3.12). Table 5.1 shows the convergence behavior of $P_h\mathbf{f}$. We see that $P_h\mathbf{f}$ provides an accurate approximation of $\mathbf{f}_2$. Moreover, $\mathbf{f} - P_h\mathbf{f}$ furnishes an approximation of $\mathbf{f}_1$ of the same accuracy since

$$\mathbf{f}_1 - (\mathbf{f} - P_h\mathbf{f}) = \mathbf{f}_1 - (\mathbf{f}_1 + \mathbf{f}_2 - P_h\mathbf{f}) = -(\mathbf{f}_2 - P_h\mathbf{f}).$$

The vectors $\mathbf{f}_1$ and $P_h\mathbf{f}$ are nearly orthogonal. Table 5.1 shows the inner product $(\mathbf{f}_1, P_h\mathbf{f})$. Also note that

$$\cos^{-1}\left(\frac{(\mathbf{f}_1, P_h\mathbf{f})}{\|\mathbf{f}_1\|\|P_h\mathbf{f}\|}\right) \approx 1.5708,$$

which is about $\frac{\pi}{2}$.

Example 2: Nearly singular system. We solve the nearly singular system defined by (4.3) with $\delta = 10^{-k}$, $k = 6, 8, 10, 12$. The mesh sizes are $h = \frac{1}{32}$ and $h = \frac{1}{64}$. The problem data $\mathbf{f}$ is defined by $\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$, where

$$\begin{aligned} \mathbf{f}_1 &= -\mathbf{grad}_h(2(x-x^2) + 2(y-y^2) + \delta(x-x^2)(y-y^2)), \\ \mathbf{f}_2 &= \delta\,\mathbf{curl}((x-x^2)(y-y^2)). \end{aligned}$$

As described in Subsection 4.3, we first compute the approximation $P_h\mathbf{f}$ of $\mathbf{f}_2$ with Algorithm 3.1. We use TOL $= 10^{-20}$ in the algorithm, which then requires 9 iterations for all values of δ; see Table 5.2 for details.

Table 5.1: Performance for different δ values with $h = \frac{1}{32}$.

δ	0.1	0.3	0.5	0.7	0.9
$\|\mathbf{f}_2 - P_h\mathbf{f}\|$	$3.0 \cdot 10^{-13}$	$4.2 \cdot 10^{-12}$	$4.2 \cdot 10^{-12}$	$8.8 \cdot 10^{-11}$	$1.8 \cdot 10^{-9}$
$(\mathbf{f}_1, P_h\mathbf{f})$	$5.8 \cdot 10^{-11}$	$1.2 \cdot 10^{-11}$	$1.3 \cdot 10^{-11}$	$2.7 \cdot 10^{-10}$	$5.6 \cdot 10^{-9}$
$\|\nabla \cdot P_h\mathbf{f}\|$	$6.8 \cdot 10^{-9}$	$8.6 \cdot 10^{-9}$	$1.8 \cdot 10^{-8}$	$1.7 \cdot 10^{-8}$	$1.9 \cdot 10^{-8}$
# of iterations	9	8	8	7	6

Then we use $\mathbf{f} - P_h\mathbf{f}$ as input data for Algorithm 4.1 to determine an approximation of $\boldsymbol{\sigma}_1^h$ corresponding to the data $\mathbf{f}_1$. When computing the approximation of $\boldsymbol{\sigma}_1^h$, we use TOL $= 10^{-10}$ in Algorithm 4.1, which requires 4 iterations for all values of δ. Results are collected in Table 5.3. Note that $\frac{P_h\mathbf{f}}{\delta}$ is our approximation of $\boldsymbol{\sigma}_h^2$.

Table 5.2: Performance for different δ values with $h = \frac{1}{32}$.

δ	$(0.1)^6$	$(0.1)^8$	$(0.1)^{10}$	$(0.1)^{12}$
$\|\mathbf{f}_2 - P_h\mathbf{f}\|$	$6.6 \cdot 10^{-13}$	$6.6 \cdot 10^{-15}$	$1.6 \cdot 10^{-16}$	$1.5 \cdot 10^{-16}$
$\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_1^h\|$	$4.7 \cdot 10^{-3}$	$4.7 \cdot 10^{-3}$	$4.7 \cdot 10^{-3}$	$4.7 \cdot 10^{-3}$
$\|\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\|$	$7.6 \cdot 10^{-3}$	$7.6 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	$2.1 \cdot 10^{-1}$
$\frac{\|\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\|}{\|\boldsymbol{\sigma}_2\|}$	$5.0 \cdot 10^{-8}$	$5.1 \cdot 10^{-10}$	$5.3 \cdot 10^{-12}$	$1.4 \cdot 10^{-12}$

Table 5.3: Performance for different δ values with $h = \frac{1}{64}$.

δ	$(0.1)^6$	$(0.1)^8$	$(0.1)^{10}$	$(0.1)^{12}$
$\|\mathbf{f}_2 - P_h\mathbf{f}\|$	$5.8 \cdot 10^{-14}$	$5.9 \cdot 10^{-16}$	$8.4 \cdot 10^{-17}$	$8.3 \cdot 10^{-17}$
$\|\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_1^h\|$	$2.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$
$\|\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\|$	$3.8 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$6.4 \cdot 10^{-3}$	$5.1 \cdot 10^{-1}$
$\frac{\|\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\|}{\|\boldsymbol{\sigma}_2\|}$	$2.5 \cdot 10^{-8}$	$2.5 \cdot 10^{-10}$	$4.3 \cdot 10^{-12}$	$3.4 \cdot 10^{-12}$

Example 3: Our last two tables show the quality of the computed solutions, $\boldsymbol{\sigma}_h^P$, determined by our proposed iterative method (Algorithm 4.1), and compares them to the quality of the solutions, $\boldsymbol{\sigma}_h^D$, computed with the MATLAB direct solver \. Table 5.4 shows results when $\mathbf{f} = \mathbf{f}_1 + \delta\mathbf{curl}(x - x^2)(y - y^2)$ and Table 5.5 for $\mathbf{f} = \mathbf{f}_1 + \sqrt{\delta}\mathbf{curl}(x - x^2)(y - y^2)$. Note that $\boldsymbol{\sigma}_h^P = \boldsymbol{\sigma}_h^1 + \boldsymbol{\sigma}_h^2$, where $\boldsymbol{\sigma}_h^1$ and $\boldsymbol{\sigma}_h^2$ are defined in Example 2. Al-

gorithm 4.1 can be seen to yield higher accuracy when $\delta > 0$ is small. The entries $-$ in Tables 5.4 and 5.5 indicate that the MATLAB direct solver was not able to compute a solution. Further illustrations of the iterative method can be found in [17].

Table 5.4: Performance of proposed solution $\boldsymbol{\sigma}_h^P$ and $\boldsymbol{\sigma}_h^D$ for different δ values with $h = \frac{1}{128}$.

δ	$(0.1)^6$	$(0.1)^8$	$(0.1)^{10}$	$(0.1)^{12}$
$\lVert\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_1^h\rVert$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$
$\lVert\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\rVert$	$1.9 \cdot 10^{-3}$	$1.9 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$	$1.4 \cdot 10^{0}$
$\frac{\lVert\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\rVert}{\lVert\boldsymbol{\sigma}_2\rVert}$	$1.3 \cdot 10^{-8}$	$1.3 \cdot 10^{-10}$	$9.8 \cdot 10^{-12}$	$9.7 \cdot 10^{-12}$
$\lVert\boldsymbol{\sigma} - \boldsymbol{\sigma}_h^P\rVert$	$2.2 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	$1.4 \cdot 10^{0}$
$\lVert\boldsymbol{\sigma} - \boldsymbol{\sigma}_h^D\rVert$	$2.2 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$2.3 \cdot 10^{-2}$	$-$

Table 5.5: Performance of proposed solution $\boldsymbol{\sigma}_h^P$ and $\boldsymbol{\sigma}_h^D$ for different δ values with $h = \frac{1}{128}$.

δ	$(0.1)^6$	$(0.1)^8$	$(0.1)^{10}$	$(0.1)^{12}$
$\lVert\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_1^h\rVert$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$
$\lVert\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\rVert$	$1.9 \cdot 10^{0}$	$1.9 \cdot 10^{1}$	$1.9 \cdot 10^{2}$	$1.9 \cdot 10^{3}$
$\frac{\lVert\boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_2^h\rVert}{\lVert\boldsymbol{\sigma}_2\rVert}$	$7.8 \cdot 10^{-6}$	$1.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$
$\lVert\boldsymbol{\sigma} - \boldsymbol{\sigma}_h^P\rVert$	$1.9 \cdot 10^{0}$	$1.9 \cdot 10^{1}$	$1.9 \cdot 10^{2}$	$1.9 \cdot 10^{3}$
$\lVert\boldsymbol{\sigma} - \boldsymbol{\sigma}_h^D\rVert$	$1.9 \cdot 10^{0}$	$1.9 \cdot 10^{1}$	$2.1 \cdot 10^{3}$	$-$

6. Conclusion

The paper describes a new iterative method for accurate approximation of the Helmholtz decomposition of a given vector. The method can be applied to a finite element solution for the simulation of incompressible fluids to satisfy a divergence-free condition.

Acknowledgment

The research was supported in part by NSF grants DMS-1720259 and DMS-1729509.

References

[1] D. N. Arnold, R. S. Falk, and R. Winther, *Preconditioning in $H(\mathrm{div})$ and applications*, Math. Comp., 66 (1997), pp. 957–984.

[2] C. Bahriawati and C. Carstensen, *Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori control*, Comput. Methods Appl. Math., 5 (2005), pp. 333–361.

[3] H. Bhatia, G. Norgard, V. Pascucci and P.-T. Bremer, *The Helmholtz-Hodge decomposition – a survey*, IEEE Trans. Vis. Comput. Graph., 19 (2013), pp. 1386–1404.

[4] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, New York, 2008.

[5] C. Brezinski and M. Redivo Zaglia, *The genesis and early developments of Aitken's process, Shanks' transformation, the epsilon-algorithm, and related fixed point methods*, Numer. Algorithms, 80 (2019), pp. 11–133.

[6] C. Brezinski, M. Redivo Zaglia, and Y. Saad, *Shanks sequence transformations and Anderson acceleration*, SIAM Rev., 60 (2018), pp. 646–669.

[7] F. Brezzi, J. Douglas, Jr., and L. D. Marini, *Two families of mixed finite elements for second order elliptic problems*, Numer. Math., 47 (1985), pp. 217–235.

[8] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, Berlin, 1991.

[9] F. Brezzi, M. Fortin, and R. Stenberg, *Error analysis of mixed-interpolated elements for Reissner-Mindlin plates*, Math. Models Methods Appl. Sci., 1 (1991), pp. 125–151.

[10] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

[11] M. Eiermann and L. Reichel, *On the application of orthogonal polynomials to the iterative solution of singular linear systems of equations*, eds. J. Dongarra, I. Duff, P. Gaffney, and S. McKee, Vector and Parallel Computing, Prentice-Hall, Englewood Cliffs, NJ, 1989, pp. 285–297.

[12] V. Girault and P. A. Raviart, *Finite Element Methods for Navier-Stokes equations*, Springer, Berlin, 1986.

[13] W. W. Hager, *Iterative methods for nearly singular linear systems*, SIAM J. Sci. Comput., 22 (2000), pp. 747–766.

[14] M. Hanke and M. Hochbruck, *A Chebyshev-like semiiteration for inconsistent linear systems*, Electron. Trans. Numer. Anal., 1 (1993), pp. 89–103.

[15] K. Hayami and M. Sugihara, *A geometric view of Krylov subspace methods on singular systems*, Numer. Linear Algebra Appl., 18 (2011), pp. 449–460, and 21 (2014), pp. 701–702.

[16] K. Jbilou and H. Sadok, *Vector extrapolation methods. Applications and numerical comparison*, J. Comput. Appl. Math., 122 (2000), pp. 149–165.

[17] J. Ku and L. Reichel, *Simple efficient solvers for certain ill-conditioned systems of linear equations, including $H(\mathrm{div})$ problems*, J. Comput. Appl. Math, 343 (2018), pp. 240–249.

[18] A. Napov and Y. Notay, *An algebraic multigrid method with guaranteed convergence rate*, SIAM J. Sci. Comput., 34 (2012), pp. A1079–A1109.

[19] Y. Notay, *AGMG software and documentation*, available at `http://homepages.ulb.ac.be/~ynotay/AGMG`

[20] Y. Notay, *An aggregation-based algebraic multigrid method*, Electron. Trans. Numer. Anal., 37 (2010), pp. 123–146.

[21] Y. Notay, *Aggregation-based algebraic multigrid for convection-diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2288–A2316.

[22] P. A. Raviart and J. M. Thomas, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of the Finite Element Method, eds L. I. Galligani and E. Magenes, Lecture Notes in Mathematics, vol. 606, Springer, New York, 1977, pp. 292–315.

[23] L. Reichel and Q. Ye, *Breakdown-free GMRES for singular systems*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1001–1021.

[24] L. Wahlbin, *Superconvergence in Galerkin Finite Element Methods*, Lecture Notes in Mathematics, vol. 1605, Springer, Berlin, 1995