8-2023

Null Space Removal in Finite Element Discretizations

Pengfei Jia
pengfej@clemson.edu

Null Space Removal in Finite Element Discretizations

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Applied Mathematics

---

by
Pengfei Jia
August 2023

Accepted by:
Dr. Timo Heister, Committee Chair
Dr. Xue Fei
Dr. Leo Rebholz

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Partial differential equations are frequently utilized in the mathematical formulation of physical problems. Boundary conditions need to be applied in order to obtain the unique solution to such problems. However, some type of boundary conditions do not lead to unique solutions because the continuous problem has a nullspace. This non-uniqueness manifests when the problem is discretized. For instance, Neumann boundary condition only fixes the normal derivative of the solution on the boundary, which only determines the solution up to a constant. Here, we present several methods to remove such null space for different problems: Laplace problem, Linear Elasticity problem and Stokes problem.

The outline is as follows: We first review the foundation of all three problems and prove that Laplace problem, linear elasticity problem and Stokes problem can be well posed if we restrict the test and trail space in the continuous and discrete finite element setting.

Next, we introduce methods to solve the linear system and obtain a numerical solution. We first explain how to compute a basis of the null space numerically. We will examine two types of null space: translation and rotation. Then, we show how to remove such a null space from the preconditioner, the right hand side and the discrete solution or to condense it into the linear system. We will also utilize the method of fixing DoFs, which guarantees uniqueness of the linear system. Moreover, we discuss the capability of CG and GMRES in solving resulting linear system of equations.

Finally, we will present the numerical experiments conducted in the finite element library deal.II. We compare the numerical and graphical output to evaluate the performance of each method,

providing a more comprehensive understanding of the strengths and limitations.

Chapter 2

Literature Review

In this section, we will discuss the null space and well posedness of Laplace problem, linear elasticity problem and Stokes problem with Neumann boundary condition in continuous level. Discrete form and construction of linear system will also be mentioned as well as the corresponding null space.

2.1 Laplace Problem

We start with Laplace problem with mixed boundary conditions [12, 34]:

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \Gamma_D \\ \partial_n u &= g \quad \text{on} \quad \Gamma_N. \end{aligned}$$

Let $\Omega \subseteq \mathbb{R}$, where $\Gamma_D \cup \Gamma_N = \partial\Omega$ is the boundary of Ω and f, g are real valued function that represent external force. $\partial^n u$ is the normal derivative of u and $g \in H^1(\partial\Omega)$. To derive the weak formulation, if we pick the test space and solution space to be $H_0^1(\Omega) = \{u \in H^1(\Omega) | u = 0 \text{ on } \Gamma_D\}$, then for any $v \in H_0^1(\Omega)$:

$$\int_\Omega -\Delta u v dx = \int_\Omega -\nabla \cdot (\nabla u) v dx$$
$$= \int_\Omega \nabla u \cdot \nabla v dx - \int_{\Gamma_D} (n \cdot \nabla u) v ds - \int_{\Gamma_N} g v ds \quad \text{by green's formula.}$$

We define the bilinear form as $a(u,v) = \int_\Omega \nabla u \cdot \nabla v dx$ and $L(v) = \int_{\Gamma_N} g v dx ds$.

Now consider Laplace problem but with Neumann boundary conditions on $H^1(\Omega)$, the problem becomes:

$$-\Delta u = f \quad \text{in} \quad \Omega$$
$$\partial^n u = g \quad \text{on} \quad \partial\Omega.$$

So the bilinear form is

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v$$

and the right hand side is

$$L(v) = \int_\Omega fv + \int_{\partial\Omega} (n \cdot \nabla u) v = \int_\Omega fv + \int_{\partial\Omega} gv$$

by Neumann boundary conditions. The compatibility condition for Neumann boundary condition is:

$$\int_\Omega f + \int_{\partial\Omega} g = 0$$

The solution of the problem is determined up to a constant, which creates a null space. The formulation of Laplace problem with Neumann boundary conditions only involves second derivative and normal derivative. If a solution is found, add any constant to this solution will give us another solution and both will satisify the formulation, see [14].

We can solve this issue by fixing the integral of solution over the domain. We impose $\int_\Omega u = 0$ onto $H^1(\Omega)$. Denote $V = \{u \in H^1(\Omega) : \int_\Omega u dx = 0\}$ with the same norm of $H^1(\Omega)$ which is $||\cdot||_{1,\Omega}$. The weak formulation we consider is:

$$\text{find } u \in V \quad \text{such that}$$
$$a(u,v) = L(v) \quad \forall v \in V.$$

To show uniqueness, we will apply Lax-Milgram, Banach-Necas-Babusk and Poincare-Friedrichs inequality, by [1]:

Theorem 2.1.1. *(Lax-Milgram): Assume V is a Hilbert space and $a \in L(V \times V, \mathbb{R})$ and let $f \in L(V)'$. If the bilinear form is coercive, i.e. $\exists \alpha > 0, \quad \forall u \in V, \quad a(u,u) \geq \alpha ||u||_V^2$, then the problem is well-posedwith a priori bound of the solution:*

$$\forall f \in V' \quad ||u||_V \leq \frac{1}{\alpha} ||f||_{V'}.$$

Theorem 2.1.2. *(Banach-Necas-Babuska): Let W be a Banach space and V be a reflective Banach space. Let $a \in L(W \times V, \mathbb{R}), f \in V'$, then the problem is well-posediff*

$$\exists \alpha > 0 \quad \inf_{w \in W} \sup_{v \in V} \frac{a(w,v)}{||w||_W ||v||_V} \geq \alpha \quad \textit{(BNB1)}$$
$$\forall v \in V \quad (\forall w \in W, a(w,v) = 0) \Rightarrow (v = 0). \quad \textit{(BNB2)}$$

If the problem is well-posed, the following error estimate holds:

$$\forall f \in V\prime \quad ||u||_W \leq \frac{1}{\alpha} ||f||_{V\prime}.$$

Remark 1. *Assume $W = V$ then Lax-Milgram implies BNB1 and BND2.*

Theorem 2.1.3. *(Poincare-Friedrichs) Let $1 \leq p < \infty$ and Ω be a bounded connected open set having extension property. Let M be a linear form on $W^{1,p}(\Omega)$ whose restriction on constant function is not zero, let $W = \{v \in W^{1,p}(\Omega) \quad M(v) = 0\}$ then W is a closed subspace of $W^{1,p}(\Omega)$ and $\forall v \in W, c||v||_{W^{1,p}(\Omega)} \leq ||\nabla v||_{L^p(\Omega)}$.*

Theorem 2.1.4. *(Well Posedness) The problem above is well-posed with the following error bound:*

$$\forall f \in L^2(\Omega) \quad \forall g \in L^2(\partial\Omega) \quad ||u||_{1,\Omega} \leq c(||f||_{0,\Omega} + ||g||_{0,\partial\Omega}).$$

Proof. We will try to take advantage of Poincare-Friedrichs inequality by applying it to $W = V$. Define a linear form $M(u) = \int_{\partial\Omega} u$, then by construction of V, $M(v) = 0$ for all v in V, then Poincare-Friedrichs inequality implies the following:

$$\forall u \in W, \quad c||u||_W \leq ||\nabla u||_{L^2(\Omega)}.$$

Let $v = u$, then

$$a(u,u) = \int_\Omega \nabla u \cdot \nabla u = \int_\Omega (\nabla u)^2 = |\nabla u|^2_{0,\Omega} = ||\nabla u||^2_{L^2(\Omega)} \geq c^2||u||^2_W \text{ for some } c.$$

Therefore, the bilinear form $a(u,v)$ is coercive on $V = \{u = H^1(\Omega) | \int_\Omega u = 0\}$. Following Lax-Milgram, we may claim the problem is well posed [1]. □

Remark 2. *We fix the integral over the domain by mean value constraint, i.e, impose $\int_\Omega u = 0$ onto $H^1(\Omega)$. An alternative way to do so is, instead of imposing mean value constraint, we impose mean value over the boundary. It can be shown that this constraint will make the problem well-posed as well, see [1] for more details. The proof is similar to the one with mean value constraint.*

2.1.1 Discrete Problems

Consider the following discrete form of Laplace problem with Neumann boundary conditions.

$$\text{Seek } u_h \in V_h \text{ such that :}$$
$$a(u_h, v_h) = L(v_h) \quad \forall v_h \in V_h$$

Let V_h be a finite dimensional subspace of V with basis $\{\phi_i\}$. Since a is coercive in V, then a is also coercive in V_h. It has shown in [1] that with Q_1 finite element, the constraint will be satisified and the discrete form is well-posed.

2.2 Linear Elasticity Problem

Linear elasticity problem is formulated as follows: Consider a deformable object and some external load f, where if we apply the load to the object, it will start to deform. Our interest is the displacement field u when the object reaches equilibrium again. Denote $\Omega \subseteq \mathbb{R}^3$ to be the initial stage, then define stress tensor σ so that at equilibrium stage, $\nabla \cdot \sigma + f = 0$ and define the strain tensor $\epsilon = \frac{1}{2}(\nabla u + \nabla u^T)$. By linear isotropic elasticity, the stress strain tensor can be related as follows:$\sigma(u) = \lambda(tr(\epsilon(u)))I + 2\mu\epsilon(u)$. Where λ, μ are called Lame coefficient and I is identity matrix. Combining the two definitions, we have $\sigma(u) = \lambda(\nabla \cdot u)I + \mu(\nabla u + \nabla u^T)$, see [13, 19].

For boundary conditions, Mixed boundary conditions and Neumann boundary conditions are mostly considered. For Mixed boundary conditions, we split the boundary condition into two: $u = 0$ for $u \in \Gamma_1$ and $\partial_n u = g$ for $u \in \Gamma_2$ where Γ_1, Γ_2 together form the whole boundary. For Neumann boundary conditions, we only have $\partial_n u = g$ for $u \in \Gamma$. Such problem is also referred to the pure traction problem. Later we will refer to the mixed boundary condition linear elasticity problem as mixed boundary problem and linear elasticity problem with Neumann boundary condition as pure traction problem.

The mixed boundary problem is defined as:

$$\begin{aligned} -\nabla \cdot (\sigma(u)) &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \Gamma_1 \\ \partial_n u &= g \quad \text{on} \quad \Gamma_2. \end{aligned}$$

The pure traction problem is defined as:

$$\begin{aligned} -\nabla \cdot (\sigma(u)) &= f \quad \text{in} \quad \Omega \\ \partial_n u &= g \quad u \quad \text{on} \quad \Gamma. \end{aligned}$$

To derive the bilinear form, we will start with:

$$-\nabla \cdot (\sigma(u)) = f.$$

If we multiply the test function v on both side and take integral, we will have:

$$\int_\Omega -\nabla \cdot (\sigma(u)) \cdot v = \int_\Omega f \cdot v.$$

If we do integration by parts, we will have:

$$\int_\Omega \sigma(u) : \nabla v = \int_\Omega f \cdot v + \int_\Gamma n \cdot \sigma(u) \cdot v.$$

Moreover, plug in the relationship between two tensors into the bilinear form, we have

$$\begin{aligned} a(u,v) &= \int_\Omega \sigma(u) : \nabla(v) = \int_\Omega (\lambda(\nabla \cdot u)I + \mu(\nabla u + \nabla u^T)) : \nabla(v)dx \\ &= \int_\Omega \lambda \nabla \cdot u(I : \nabla(v)) + 2\mu\epsilon(u) : \nabla(v)dx \\ &= \int_\Omega \lambda(\nabla \cdot u)(\nabla \cdot v) + 2\mu(\epsilon(u) : \nabla(v))dx. \end{aligned}$$

We are more interested in the pure traction problem. Choosing $[H^1(\Omega)]^3 = \{(u_1, u_2, u_3) : u_1, u_2, u_3 \in H^1(\Omega)\}$ as test and solution space, we may derive the weak formula as follows:

$$\begin{aligned} &\textit{Seek } u \in [H^1(\Omega)]^3 \textit{such that} \\ &a(u,v) = \int_\Omega f \cdot v + \int_\Gamma g \cdot v, \forall v \in [H^1(\Omega)]^3 \\ \text{With } &a(u,v) = \int \sigma(u) : \nabla(v) = \int_\Omega \lambda(\nabla \cdot u)(\nabla \cdot v) + \int_\Omega 2\mu\epsilon(u) : \nabla(v). \end{aligned}$$

However, this weak form is not well-posed. The idea is as following: The physical law described by pure traction problem does not require a specific location of object in three-dimensional space, therefore, without further restriction, we will have infintely many solutions, i.e., there's a null space generated by rigid body movement. To remove this null space, we need to first understand the behavior of rigid body movement. In a three-dimensional environment, the translation of an object has a three-dimensional basis, namely, translate along x-axis, y-axis and z-axis. Moreover, rotation of an object also has a three-dimensional basis, rotation around x-axis, y-axis and z-axis.

One way to take care of the null space generated by rigid body motion is to choose $H^1_\perp(\Omega)$

as test and solution space.

$$H^1_\perp(\Omega) = \{u \in H^1(\Omega) \quad | \quad \int_\Omega u = 0 \quad \int_\Omega \nabla \times u = 0\}$$

To show that the pure traction problem is well-posed under $H^1_\perp(\Omega)$, we will need the following Lemma:

Theorem 2.2.1. *(Petree-Tartar Lemma) Let X, Y, Z be banach spaces, let $A \in L(X,Y)$ be an injective operator and $T \in L(X,Z)$ be a compact operator. If there exists a constant c such that $c||x||_X \leq ||Ax||_Y + ||Tx||_Z$, then there exist an $\alpha > 0$ such that $\alpha||x||_X \leq ||Ax||_Y$.*

Now we present the proof of uniqueness, let $RM = \{\alpha + \beta \times x\}$, where $\alpha, \beta \in \mathbb{R}^d, x = \{x_1, x_2, ..., x_d\}$, d is the dimension of the problem:

Theorem 2.2.2. *Let $\Omega \subseteq \mathbb{R}^3$, assume $f \in [L^2(\Omega)]^3, g \in [L^2(\partial\Omega)]^3$ such that*

$$\int_\Omega f \cdot v + \int_{\partial\Omega} g \cdot v = 0 \quad \forall v \in RM.$$

Then the pure traction problem is well-posed in $H^1_\perp(\Omega)$ and there exists a constant c such that:

$$||u||_{1,\Omega} \leq c(||f||_{0,\Omega} + ||g||_{0,\partial\Omega}).$$

Proof. Let $X = H^1_\perp(\Omega)$ and $Y = [L^2(\Omega)]^{3,3}$ then we define $A : X \to Y$ by $Au = \epsilon(u)$. Let $u_1, u_2 \in X$ and assume $A(u_1) = A(u_2)$, then $Au_1 - Au_2 = 0$, which mean $A(u_1 - u_2) = 0$ by gradient operator is a linear operator and linear combination of linear operator is again a linear operator. Then we will have $\epsilon(u_1 - u_2) = 0$. By Korn's inequality, we show that $\epsilon(u) = 0$ if and only if $u \in RM$ but if $u \in H^1_\perp(\Omega)$ then $u \notin RM$ by orthogonality in the definition, so $\epsilon(u_1 - u_2) = 0 \Rightarrow u_1 = u_2$. Therefore, A is injective.

Set $Z = [L^2(\Omega)]^3$ and let $T : X \to Z$ be a compact embedding operator from H^1 to L^2 guaranteed to exist by Rellich Kondrachov theorem, then we may apply Korn's second lemma to show that

$$\forall u \in X, \quad ||u||_X \leq c(||Ax||_Y + ||Tx||_z).$$

This satisify the assumption of Petree-Tartar's lemma, therefore, applying Petree-Tartar lemma we

may further restrict our inequality:

$$\forall u \in X \quad ||u||_X \leq c||\epsilon(u)||_Y \quad \Rightarrow \quad ||u||_{1,\Omega} \leq c||\epsilon(u)||_{0,\Omega}.$$

Therefore, we have $a(u, v)$ is coercive on $H^1_\perp(\Omega)$. It can be shown that $H^1_\perp(\Omega)$ is again Hilbert since it's a subset of $H^1(\Omega)$ and it's closed. Applying Lax-Milgram will then give us the conclusion that the problem is well-posed, see [1]. □

2.2.1 Discrete Problems

Let $\hat{H}^1_\perp$ be a finite dimensional subspace of $H^1_\perp(\Omega)$, we obtain $\hat{H}^1_\perp$ as introduced in [1, 33], basically we use continuous Lagrange finite element to approximate the continuous bilinear form of degree $k \geq 1$. Then it can be shown that the finite element approximation is well-posed.

We would like to write the approximated solution u_h as a linear combination of basis function, so we can use the same method as before to derive the linear system. Let $\Phi_i(x)$ be the basis function of $\hat{H}^1_\perp(\Omega)$. Since linear elasticity problem is a vector-valued problem, the basis function $\Phi_i(x)$ is also a vector-valued function. The dimension of $\Phi_i(x)$ is equal to the dimension of the solution. To number the basis function, denote $\Phi_i(x) = \phi_i(x)e_{comp(i)}$, where $comp(i)$ is the component of i that is nonzero and e_i is the ith unit vector. So we approximate each component with $\phi_i(x)$ and combine them with $e_{comp(i)}$. Hence, we can still represent the approximated solution u_h as a linear combination of basis function $\Phi_i(x)$. Therefore, we can derive the linear system as usual:

Recall the bilinear form is:

$$a(u, v) = \int_\Omega \lambda(\nabla \cdot u)(\nabla \cdot v) + \int_\Omega 2\mu\epsilon(u) : \nabla(v).$$

If we substitute $u_h = \sum_{i=1}^{N} U_i \Phi_i$ and $v_h = \sum_{j=1}^{N} V_j \Phi_j$, we will have:

$$
\begin{aligned}
a(u_h, v_h) &= \lambda \int_\Omega (\nabla \cdot \sum_{i=1}^{N} U_i \Phi_i)(\nabla \cdot \sum_{j=1}^{N} V_j \Phi_j) + 2\mu \int_\Omega \epsilon(\sum_{i=1}^{N} U_i \Phi_i) : \nabla(\sum_{j=1}^{N} V_j \Phi_j) \\
&= \lambda \int_\Omega (\sum_{i=1}^{N} U_i \sum_k (\Phi_i)_k)(\sum_{j=1}^{N} V_j \sum_l (\Phi_i)_l) + 2\mu \int_\Omega (\nabla U : \nabla V) + (\nabla U^T : \nabla V) \\
&= \lambda \int_\Omega (\sum_{i=1}^{N} U_i \sum_k (\Phi_i)_k)(\sum_{j=1}^{N} V_j \sum_l (\Phi_i)_l) + 2\mu \int_\Omega \sum_{i=1}^{N} U_i \sum_{j=1}^{N} V_j (\nabla \Phi_i : \nabla \Phi_j) + \sum_{i=1}^{N} U_i \sum_{j=1}^{N} V_j (\nabla \Phi_i^T : \nabla \Phi_j) \\
&= \lambda \int_\Omega (\sum_{i=1}^{N} U_i \sum_k (\Phi_i)_k)(\sum_{j=1}^{N} V_j \sum_l (\Phi_i)_l) + 2\mu \int_\Omega \sum_{i=1}^{N} U_i \sum_{j=1}^{N} V_j (\sum_k \sum_l \partial_k (\Phi_i)_l \partial_l (\Phi_j)_k) + (\partial_l (\Phi_i)_k \partial_l (\Phi_j)_k)
\end{aligned}
$$

Here, $(\cdot)_k$, $(\cdot)_l$ are the partial derivative with respect to x_k, x_l, k, l range from 1 to d, which is the dimension of problem, we can compute each entry in linear system based on the last formula, see [7] for more details. It has been shown in [27] that if we apply Q_1 finite element to this weak form, we will obtain well posedness.

2.3 Stokes Problem

The Stokes equation is defined as follows, see [15]:

$$
\begin{aligned}
-2\nabla \cdot (\epsilon(u)) + \nabla p &= f \quad \text{in} \quad \Omega \\
-\nabla \cdot u &= 0 \quad \text{in} \quad \Omega \\
u &= g \quad \text{on} \quad \partial\Omega.
\end{aligned}
$$

Where $u : \Omega \to \mathbb{R}^d$ is the velocity of the flow, p is the pressure and f is an external force. By definition, u and f are vector-valued function and p is a scalar-valued function.

To derive the weak form of Stokes problem, let V, Q be the test space for u and p. Then for $v \in V, q \in Q$:

$$\int_\Omega -2\nabla \cdot (\epsilon(u))v + \nabla p v dx = \int_\Omega f v dx$$
$$-\int_\Omega q\nabla \cdot u = 0.$$

Consider the first equation, by property of divergence, matrix-vector product and matrix-matrix contractions:

$$\nabla \cdot (v\epsilon(u)) = \nabla \cdot (\epsilon(u))v + \nabla v\epsilon(u) \Rightarrow \nabla \cdot (\epsilon(u))v = \nabla \cdot (v\epsilon(u)) - \nabla v\epsilon(u).$$

Then

$$\int_\Omega -2\nabla \cdot (v\epsilon(u)) + \int_\Omega 2\nabla v\epsilon(u) + \int_\Omega \nabla p v = \int_\Omega f v.$$

By divergence theorem,

$$\int_\Omega -2\nabla \cdot (v\epsilon(u)) = \int_{\partial\Omega} -2n \cdot v\epsilon(u).$$

So the weak form is

$$\int_{\partial\Omega} -2n \cdot v\epsilon(u) + \int_\Omega 2\nabla v\epsilon(u) + \int_\Omega \nabla p v = \int_\Omega f v.$$

Applying divergence theorem and property of divergence on second and third term:

$$\int_{\partial\Omega} -2n \cdot v\epsilon(u) + \int_{\Omega} 2\nabla v\epsilon(u) - \int_{\Omega} p\nabla \cdot v + \int_{\partial\Omega} n \cdot vp = \int_{\Omega} fv.$$

If we add first and second equation,

$$\int_{\partial\Omega} -2n \cdot v\epsilon(u) + \int_{\Omega} 2\nabla v\epsilon(u) - \int_{\Omega} p\nabla \cdot v + \int_{\partial\Omega} n \cdot vp - \int_{\Omega} q\nabla \cdot u = \int_{\Omega} fv.$$

Since we have $u \in H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, we can remove the first and fourth term. Therfore,the weak form is: Find $u \in H_0^1(\Omega), p \in L^2(\Omega)$ such that $a((u,p),(v,q)) = L(v)$ for all $v \in H_0^1(\Omega), q \in L^2(\Omega)$ where

$$\begin{aligned} a((u,p),(v,q)) &= \int_{\Omega} 2\nabla v\epsilon(u) - \int_{\Omega} p\nabla \cdot v - \int_{\Omega} q\nabla \cdot u \\ L(v) &= \int_{\Omega} fv. \end{aligned}$$

However, the Stokes problem runs into the same issue as Laplace problem with Neumann boundary condition, which is pressure is fixed up to a constant. We may use the same method to take care of constant of pressure. The test space for u is $[H_0^1(\Omega)]^3$ and test space for p is $\{q \in L^2(\Omega), \quad \int_{\Omega} q = 0\}$. Well posedness is shown in [1].

2.3.1 Discrete Problems

The weak form is: let $X_h \subset [H_0^1(\Omega)]^d, M_h \subset L^2_{\int=0}(\Omega)$ be the finite element spaces. We seek $u_h \in X_h, p_h \in M_h$ such that

$$\begin{aligned} a(u_h, v_h) + b(v_h, p_h) &= f(v_h) \quad \forall v_h \in X_h \\ b(u_h, q_h) &= g(q_h) \quad \forall q_h \in M_h \end{aligned}$$

Based on [1, 22], this weak form is well-posed if $Q_2 \times Q_1$ finite element is applied. We will examine the well posedness property of such finite element in numerical experiment section. We will be applying the same method as laplace problem to solve a solution with mean value constraint.

Chapter 3

Methods for Remove Null Space

In this section, we will discuss how to implement exact null space for all three problems, then discuss how to remove such null space in the linear system. We will introduce three methods in this section, all of them are specifically about solving the problem in linear system level: Apply affine constraints, solve singular system and null space removal operator. We will analyze each method and present corresponding numerical experiments in the next section.

Other than these three methods, there is another method that could also solve the problem with a null space, it is the Lagrange multiplier, see [5, 10]. This method requires different weak formulations and therefore is not discussed in this thesis.

3.1 Exact Null Space

For Laplace and Stokes problem, the solution is determined up to a constant. Therefore, the null space consists of all constant functions. Thanks to the partition of unity property, this continuous null space is exactly approximated by the one-dimensional subspace spanned by $e = \{1, 1, ..., 1\}$.

For linear elasticity problem, the null space is $RM = \{u \in [H^1(\Omega)]^d | u(x) = \alpha + \beta \times x\}$ $\alpha, \beta \in \mathbb{R}^d$, where d is the dimension of u, $x = \{x_1, x_2, ..., x_d\}$ and $\times$ is cross product, we can show RM is the null space of $a(u, v)$ based on u.

Theorem 3.1.1. *Let $a(u, v)$ be the bilinear form of linear elasticity problem with Neumann boundary condition, then $(u \in RM) \Leftrightarrow a(u, v) = 0, \quad \forall v \in [H^1(\Omega)]^3$.*

Proof. ($\Rightarrow$) If $u \in RM$ then we have $u = \begin{pmatrix} \alpha_1 + \beta_2 x_3 - \beta_3 x_2 \\ \alpha_2 + \beta_3 x_1 - \beta_1 x_3 \\ \alpha_3 + \beta_1 x_2 - \beta_2 x_1 \end{pmatrix}$. Note that the strain of u is zero by skew symmetry, we will have zero matrix. Therefore, $\epsilon(u) = 0$. Similarly $\nabla \cdot u = 0$ which directly imply that $a(u, v) = 0$.

($\Leftarrow$) If $a(u, v) = 0$ for all $v \in [H^1(\Omega)]^3$, we take $v = u$, then by definition, $a(u, u) = \int \sigma(u) : \epsilon(u)dx = \int_\Omega \lambda(\nabla \cdot u)(\nabla \cdot u) + \int_\Omega 2\mu\epsilon(u) : \epsilon(u)dx = 0$ Since integral is non-negative, we can further claim $\int_\Omega \lambda \nabla u \cdot \nabla u = \int_\Omega 2\mu\epsilon(u) : \epsilon(u)dx = 0$.

On one hand, $\int_\Omega \lambda \nabla \cdot u \nabla \cdot u dx = 0 \Rightarrow \int_\Omega (\nabla \cdot u)(\nabla \cdot u)dx = 0 \Rightarrow \lambda \int_\Omega (\nabla \cdot u)^2 dx = 0 \Rightarrow \nabla \cdot u = 0$. On the other hand, the second part of integral implies $\epsilon(u) = 0$.

Now consider the following

$$\frac{\partial^2 u_i}{\partial x_j \partial x_k} = \partial_{jk} u_i = \partial_k(\partial_j u_i) = \partial_k(\partial_j u_i + \partial_i u_j - \partial_i u_j) = \partial_k(2\epsilon_{ij}) - \partial_k \partial_i u_j$$

$$\frac{\partial^2 u_i}{\partial x_j \partial x_k} = \partial_{jk} u_i = \partial_j(\partial_k u_i) = \partial_j(2\epsilon_{ik}) - \partial_j \partial_i u_k.$$

Adding two equations, by $\epsilon(u) = 0$, we have:

$$\partial_k(2\epsilon_{ij}) - \partial_k \partial_i u_j + \partial_j(2\epsilon_{ik}) - \partial_j \partial_i u_k = 2\partial_k(\epsilon_{ij}) + 2\partial_j(\epsilon_{ik}) - \partial_i(\partial_k u_j + \partial_j u_k)$$

$$2\partial_k(\epsilon_{ij}) + 2\partial_j(\epsilon_{ik}) - \partial_i(2\epsilon_{jk}) = 0.$$

In short, $\partial_{jk} u_i = \partial_k(\partial_j u_i) = 0$ for all $1 \leq i, j, k \leq 3$, therefore we may claim that u is a linear functional of x, i.e. $u = \alpha + Bx$. Moreover, $\epsilon(u) = \nabla u + \nabla u^T = B + B^T = 0 \Rightarrow B = -B^T$ By property of skew symmetric matrix, there exist another vector β such that $Bx = \beta \times x$. Hence proved. □

Skew symmetric matrix has a general form, for any scalar $b, b_1, b_2, b_3 \in \mathbb{R}$:

$$B = \begin{pmatrix} 0 & -b \\ b & 0 \end{pmatrix} \quad \text{or} \quad B = \begin{pmatrix} 0 & b_1 & b_2 \\ -b_1 & 0 & b_3 \\ -b_2 & -b_3 & 0 \end{pmatrix}.$$

For simplicity purpose, we will use (x, y) and (x, y, z) as variables in two dimension and

three dimension in this section. The discrete null space basis come from RM. Recall by definition of RM, each function in RM can be represented by $\alpha + Bx$ and B is a skew symmetric matrix. Therefore, Bx can be written explicitly as a vector. In two dimension,

$$Bx = \begin{pmatrix} 0 & -b \\ b & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = b \begin{pmatrix} -y \\ x \end{pmatrix}$$

Hence, the basis of RM for the two-dimensional case is:

$$\text{Translation: } \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{and Rotation: } \begin{pmatrix} -y \\ x \end{pmatrix}.$$

In order to discretize x and y, we need to locate support points in the mesh as they are the discretization of x and y. The discrete basis is then: $e_x = \{1, 0, 1, 0, ...\}, e_y = \{0, 1, 0, 1, ...\}$ and $r = \{-y_1, x_1, -y_2, x_2...\}$, where x_i and y_i are the x, y coefficient of support points in the mesh. Similarly, for three-dimensional problem, the basis is then: $\{1, 0, 0, 1, 0, 0, ...\}$, $\{0, 1, 0, 0, 1, 0, ...\}$, $\{0, 0, 1, 0, 0, 1, ...\}$ for translation and $\{y_1, -x_1, 0, y_2, ...\}, \{z_1, 0, x_1, z_2, ...\}$ and $\{0, z_1, y_1, 0, ...\}$.

We will discuss several approaches for working with this null space on the discrete level in next section. One common procedure among all methods is the orthogonal projection onto the null space. The process of projecting out the null space from the solution involves several steps. First, The Gram-Schmidt method will be utilized to ensure that the basis of the null space is orthonormal. Then the null space can be removed by applying orthogonal projection. Specifically, if $N = \{n_1, n_2, ..., n_k\}$ represents the set of orthonormal basis and we want to project N out of any vector b, we will apply projection operator P_N, defined as:

$$P_N(b) = b - (\sum_{i=1}^{k} \langle b, n_i \rangle n_i) \quad \forall n_i \in N.$$

Note, $P_N(b)$ is orthogonal to RM under all finite element functions.

3.2 Constraints

In this section, we will discuss how to interpolate constraints numerically and how to condense such constraints into a linear system, see [17, 23]. Then we will discuss why condensing such

constraints is not good idea and an alternative way, fix DoFs.

3.2.1 Affine Constraints

Two constraints are considered in this paper: Translational constraint: $\int_\Omega u = 0$ and rotational constraint: $\int_\Omega \nabla \times u = 0$ We have shown that for Laplace problem and Stokes problem, imposing translational constraint will give us a unique solution and for linear elasticity problem, imposing translational and rotational null space will give us a unique solution.

In order to apply these constraints in linear system, we need to convert them into vector. In discrete level, we interpolate u_h with basis in finite dimensional space $\phi_i(x)$. Therefore,

$$u_h = \sum_{i=1}^{N} u_i \phi_i(x)$$

$$\int_\Omega u_h = \int_\Omega \sum_{i=1}^{N} u_i \phi_i(x) = \sum_{i=1}^{N} u_i \int_\Omega \phi_i(x).$$

Similarly, to interpolate $\int_\Omega \nabla \times u$, we will interpolate each component and put them together, in two dimension:

$$\begin{aligned}\int_\Omega \nabla \times u = \int_\Omega \nabla \times \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} &= \int_\Omega \partial_2 u_1 - \partial_1 u_2 \\ &= \int_\Omega \left(\partial_2 \textstyle\sum_{i=1}^{N} U_i (\Phi_i)_1 - \partial_1 \sum_{i=1}^{N} U_i (\Phi_i)_2 \right) \\ &= \sum_{i=1}^{N} U_i \int_\Omega \partial_2 (\Phi_i)_1 - \partial_1 (\Phi_i)_2.\end{aligned}$$

Since each shape function is explicitly know, we can compute the integral of shape function. Hence, we can convert continuous constraints into affine constraints and condense them into linear system.

3.2.2 Condensing Constraints into Linear System

Condensing a constraint into a linear system is one way to increase the rank of a matrix and hence obtain a unique solution. There are two major goals we wish to accomplish after condensing

extra vectors: maintain the symmetry of the matrix, keep all existing information of the matrix and adding information from the constraint to the matrix or solution.

We will take advantage of Gauss elimination. Recall in Gauss elimination, we take one row each time then do elimination based on the pivoting point in each step. The solution before and after elimination will not change since elimination is nothing but linear combination of each row. Hence, if we manipulate the matrix based on Gauss elimination method then the solution will not change.

On the other hand, symmetry of a matrix is also critical if we wish to apply iterative solvers like the conjugate gradient method. If we just condense the vector into a symmetric matrix, then symmetry is unlikely to remain. To prevent this, some preprocessing is required. Denote the linear system as $Ax = b$, $A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^n$ the augmented matrix $Q = [A \quad b]$, the algorithm is as follows:

1. Pick a pivoting column: Without loss of generality, we pick column L.
2. Transform the constraint $\sum_{i=1}^{n} k_i u_i = 0 \Rightarrow u_L = \sum_{i\neq L} \frac{k_i}{k_L} u_i$, denote $C = \{c_i\} = \{\frac{k_i}{k_L}\}$ as the constraint vector.
3. Denote each row in A as A_i, do $A_i = A_i - c_i A_L$ for $i \neq L$.
4. Eliminate Lth column of A using the constraint and replace Lth row with zero. So after elimination, Lth column should have all zero entries as well as Lth row.
5. Replace $A_{L,L}$ with with the mean of diagonal to improve condition number.

Theorem 3.2.1. *Condensing constraint with the above method will not break the symmetry of A*

Proof. We take L as constrainted degree of freedom, i.e. $c_L = 1$, constraint is then $C = \{\frac{k_1}{k_L}, \frac{k_2}{k_L}, ..., \frac{k_n}{k_L}\} = \{c_i\}$

Based on the method, let A_i be the row of A. Do

$$\hat{A}_i = A_i - c_i A_L = A_i - c_i A_L$$

Elementwise, we have

$$\hat{A_{ij}} = A_{ij} - c_i A_{Lj} = A_{ij} - c_i A_{Lj}$$

Then eliminate column L of $\hat{A}$:

$$A_i^* = \hat{A}_i - \frac{\hat{A_{iL}}}{c_L} C \quad i \neq L$$

Elementwise, we have

$$\begin{aligned} A_{ij}^* &= \hat{A_{ij}} - \frac{\hat{A_{iL}}}{c_L} c_i = A_{ij} - c_i A_{Lj} - (A_{iL} - c_i A_{LL}) c_j \\ &= A_{ij} - c_i A_{Lj} - A_{iL} c_j + c_i c_j A_{LL} \end{aligned}$$

If we interchange i and j, we will have

$$A_{ji}^* = A_{ji} - c_j A_{Li} - A_{jL} c_i + c_j c_i A_{LL}$$

Therefore,

$$A_{ij}^* - A_{ji}^* = (A_{ij} - A_{ji}) + (-c_i A_{Lj} - A_{iL} c_j) - (-c_j A_{Li} - A_{jL} c_i) + (c_i c_j A_{LL} - c_j c_i A_{LL})$$

By symmetry of A, it's clear that the difference is zero, hence $A_{ij}^* = A_{ji}^*$, therefore, condensing will maintain symmetry. □

Note, there is no restriction about which DoF to isolate and condense. But one needs to choose one, do the elimination and process forward.

The linear system of finite element often involves sparse matrix. However, the interpolation we have are likely to be full vectors. Based on the algorithm, using Gauss elimination with a full vector onto a sparse system will turn a sparse matrix into a full matrix, which is not what we wish to see. On the other hand, Computing $\int_\Omega u = \int_\Omega \nabla \times u = 0$ in each cell and iteration is expensive and not necessary, we may just remove these null space after obtaining a solution. Therefore, we will consider an alternative way, the method of fixing DoFs.

3.2.3 Fixing DoFs method

As we discussed before, the null space exists because of translation and rotation of the object and we can impose $\int_\Omega u = 0$ and $\int_\Omega \nabla \times u = 0$ to prevent such null space. However, a more direct

approach to prevent the object from translating and rotating is fixing specific points of the object. Since points doesn't exist in linear system, we need to locate the DoFs associated with the points we wish to fix, then set those DoFs to specific values in terms of constraint. Therefore, the constraint is sparse, it only contains one nonzero coefficient. Condensing such constraint will be much cheaper compared to condensing full constraints.

The number of DoFs we fix depends on the dimension of null space. For Laplace problem, since the null space is one-dimensional, fixing exactly one DoF restores the uniqueness of the problem. For two-dimensional linear elasticity problem, the null space is three-dimensional as there are translation in the x and y directions and rotation. One possible method for selecting these DoFs involves selecting a point p_1, identifying the corresponding DoFs associated with its x and y values, and fix them to zero. At this point, translation is prevented, but rotation is still possible. Therefore, another point p_2 with the same x-value as p_1 is selected, and the degree of freedom associated with the y-value of p_2 will be fixed to zero. By adopting this approach, the null space is completely removed. For Stokes problem, we will apply the same method as Laplace problem. We fix one pressure DoFs to remove the null space in pressure.

After picking points and finding the corresponding DoFs in linear system, we will fix the DoFs to zero. Therefore, each affine constraint only involves one nonzero DoF. We will condense these constraints into the linear system. The sparsity pattern will not break in this case since these constraint are sparse.

As discussed previously, fixing point will grant a solution, but it's not necessaryly the correct solution we need. Therefore, after obtaining a solution by fixing DoFs, we will remove exact null space from solution by orthogonal projection as defined previously.

The DoFs we pick to fix depends on the topology to mesh. Normally we fix the DoFs associated with corner point of the mesh so that when multilevel method or adaptive refinement are applied, these points will always exist.

There are certain advantage of this method. If we use iterative solver, less computation will be done in each iteration since we have less non-zero entries in matrix A because of sparse constraint. There are also some disadvantages. The major disadvantage is that it takes more iterations for iterative solver to converge. One possibility is the eigenvalue distribution is affected when we condense the fixing DoF constraint.

3.3 Preconditioned Singular System

Through this section, we denote the matrix obtained from finite element approximation is A, A is positive semidefinite and right hand side is b, the linear system we need to solve is $Au = b$. We have the information of null space of A as explained in previous sections.

3.3.1 Convergence of Singular system

While analyzing behaviors of iterative solver based on [32, 4, 18], we normally consider a positive definite matrix or a non-singular matrix. However, matrix with a null space may converge as well, see [20, 30]. Based on [26, 21, 25, 29], if CG is appled to solve a symmetric singular system $Au = b$ and b is in the range of A, then CG will converge. GMRES, by [31], does not require stiffness matrix to be symmetric. It has been shown in [24] that GMRES will converge if $b \in R(A)$ and $R(A) = R(A^T)$.

We have to keep the right hand side in the range of A for both iterative solvers. We have previously discussed the vector form of exact null space, therefore, we can first compute the orthonormal basis and use projection operator to remove the null space from right hand side.

A common approach to enhance the performance of iterative solver is to apply a preconditioner. We have shown that the iterative solver will converge, but with preconditioner, the situation is more complicated. If the range of preconditioner is equal to or is a subset of range of A, then the proconditioner will not map any vector into the null space of A. Therefore, convergence can be achieved. However, that's not always the case. If a preconditioner maps a vector into the null space of A because of the preconditioner has a larger range or round-off errors, then iterative solver may diverge. In the next section, we will introduce an operator to prevent this divergence from happening.

3.4 Null Space Removal Operator

Other than utilizing constraints to solve the linear system and solving singular system directly, an alternative approach is to remove the null space in the process of solving linear system. The approach is denoted as null space remove operator.

3.4.1 Null space removal operator

In order to remove the null space from each matrix vector product, we need one additional step in each iteration: project out the null space. Basically, operator P_N with orthonormal null space basis is applied to override matrix vector multiplication. Before applying CG solver, we remove nullspace from right hand side and apply the removal operator to the preconditioner. Then, every time we did matrix vector multiplication with preconditioner, we remove the null space from the product by P_N to ensure the product lies within the range of A. Hence, the assumption is satisified and CG will converge. Note, we do not apply such operator around A simply because A can only map a vector to its range. If CG failed to converge when the above implementation is utilized, then the basis of null space is not accurately implemented.

The major advantage of null space removal operator is its accuracy. If the null space is implementated correctly, it will take fewer iteration to converge compared to fix DoFs method as we will see in the next section and it's more stable than solving the singular system directly.

Chapter 4

Numerical Experiments

The numerical experiment of this thesis is based on deal.II [2, 3, 9, 8], a C++ software library supporting the creation of finite element codes and an open community of users and developers. The code of numerical experiments can be found at: `https://github.com/pengfej/dealii_cg_for_null`.

Compute mean value constraint

We integrate the solution over the whole domain, i.e., we do

$$\text{Mean value } = \frac{1}{|\Omega|}\int_{\Omega} u_h(x)dx$$

for each component of the solution, where Ω is the domain, $u_h(x)$ is the function representation of the nodal vector. The value is evaluated numerically based on the given quadrature formula, see [16] for more details. If mean value is not explicitly shown, then it's zero.

Error Estimate

Throughout this section, we will use L_2 norm to estimate errors. So within cell K, the cellwise error estimate is

$$E_K = \sqrt{\int_K \sum (u_h - u)^2 dx}$$

and the overall error estimate is then

$$E = \sqrt{\sum_K E_K^2} = \sqrt{\int_\Omega (u - u_h)^2} = \|u - u_h\|_{L^2(\Omega)}.$$

The integral on each cell is estimated by quadrature. In this thesis, we will use the same numerical quadrature as implementing matrix A and right hand side vector b.

Linear Operator Implementations

To implement null space removal operator in deal.II, we will take advantage of the class "LinearOperator" from [28].

A linear operator is applied to an object that contains the following four functions: matrix vector multiplication, matrix vector multiplication with addition, transpose matrix vector multiplication and transpose matrix vector multiplication with addition. Within the operator, we add extra functionality to these functions. In our case, after doing matrix vector multiplication, we also do projection with the result vector to keep the vector inside the range of A. Note, in order to make this happen, the null space basis need to be explicitly known in the operator, this can be done by adding an extra argument in the operator constructor.

Time Estimation

A timer will be initiated prior to the execution of the CG/GMRES solver and stopped after the solver converges. In our analysis, fix DoFs results in reduced computation within each iteration and require more iterations to achieve convergence. On the other hand, the removal operator entails increased computation per iteration, but takes fewer iterations to converge. Therefore, it would be interesting to compare their performance.

Rate of Convergence

The rate of convergence will also be calculated and subjected to comparison. Given that the L_2 error is estimated and global refinement is applied after each cycle, we expect quadratic convergence for each case by [6]. In other words, when dividing the error in the current cycle by the error in the previous cycle, we should expect a ratio value of 4 approximately. This specific column

will be denoted as "RoC" in the table of outputs in each numerical experiment.

4.1 Q1 Finite Element for Laplace Problem

The manufactured solution of Laplace problem is $u(x, y) = \sin \pi x \cos \pi y$ and the domain is the square centered at origin with length 2. The gradient of u is

$$\{\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\} = \{\pi \cos \pi x \cos \pi y, -\pi \sin \pi x \sin \pi y\}.$$

To find the right hand side function f, we take the negative divergence of gradient of u:

$$f = -\nabla \cdot \nabla u = -(-\pi^2 \sin \pi x \cos \pi y - \pi^2 \sin \pi x \cos \pi y) = 2\pi^2 \sin \pi x \cos \pi y.$$

To compute the Neumann boundary conditions, we need to take the normal derivative of u on the boundary of mesh. If we define the right boundary as boundary 1, the bottom boundary as boundary 2, the left boundary as boundary three and the top boundary as boundary 4, the corresponding normal vector is $(1, 0), (0, -1), (-1, 0), (0, 1)$.

Multiplying the graident of u with corresponding normal vector, we will get Neumann boundary condition:

$$\frac{\partial u}{\partial n} = \begin{cases} \pi \cos \pi x \cos \pi y & u \in \text{ Boundary 1} \\ \pi \sin \pi x \sin \pi y & u \in \text{ Boundary 2} \\ -\pi \cos \pi x \cos \pi y & u \in \text{ Boundary 3} \\ -\pi \sin \pi x \sin \pi y & u \in \text{ Boundary 4} \end{cases}$$

Here's a graphical output:

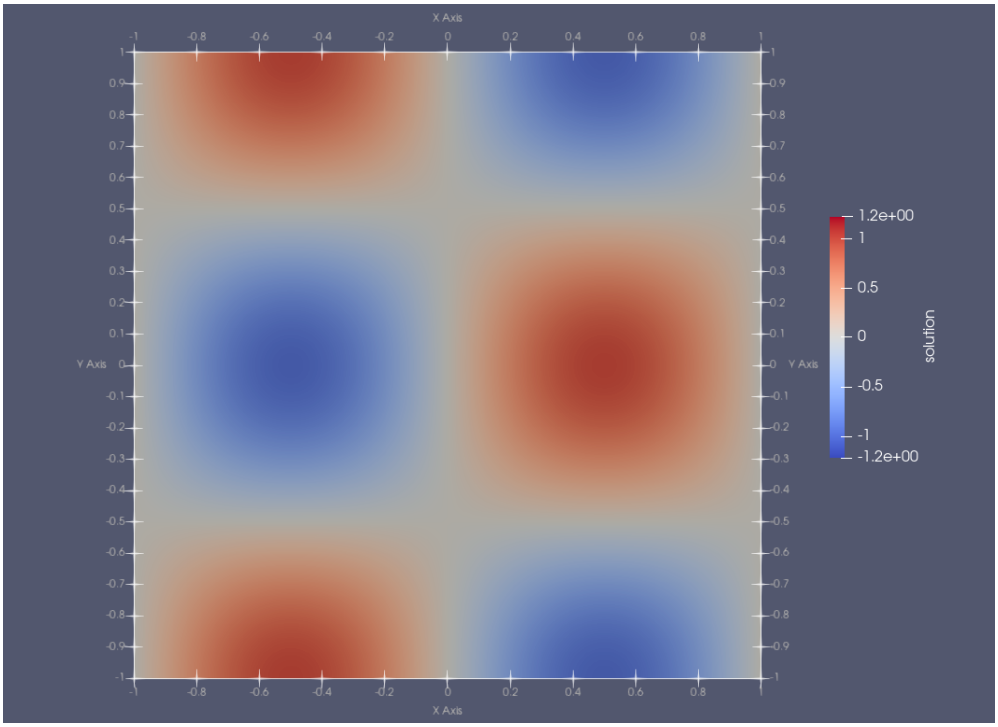

Figure 4.1: Laplace Problem with Global Refinemet

We will use Q_1 finite element to approximate the solution and after each iteration we will do a global refinement. The initial mesh is refined twice so it contains 16 cells. We will test among three cases: Solve the singular system with SSOR preconditioner, solve the same system with removal operator applied on SSOR preconditioner and establish a constraint by fixing the first DoF and condense this constraint. We will denote the result of each case by "Singular System", "Removal Operator" and "Fix dof" accordingly.

SSOR	Singular System				Removal Operator				Fix DoF			
cells	error	RoC	CPU time	Iterations	error	RoC	CPU time	Iterations	error	RoC	CPU time	Iterations
16	0.197	N/A	0.00	13	0.196166	N/A	0.00	23	0.196166	N/A	0.00	15
64	0.050889	3.87	0.00	19	0.050848	3.86	0.00	23	0.050848	3.86	0.00	21
256	0.012822	3.97	0.04	30	0.012816	3.97	0.02	35	0.012816	3.97	0.03	34
1024	0.003211	3.99	0.14	55	0.003211	3.99	0.14	64	0.003211	3.99	0.11	62
4096	0.000803	4.00	0.69	103	0.000803	4.00	0.78	109	0.000803	4.00	0.78	116
16384	0.000201	4.00	5.13	194	0.000201	4.00	5.39	208	0.000201	4.00	5.72	226
65536	0.00005	4.02	36.56	371	0.00005	4.02	36.77	371	0.00005	4.02	44.19	445

Table 4.1: L_2 Error for Laplace problem with three approaches

The interpolated null space basis, $\int_\Omega u = 0$, is removed after solving with each method and error is computed afterward. The mean value is not included since mean value for all solutions are zero. Based on the output, all three method has similar errors. The number of iterations of fixing DoFs are relatively larger. The convergence is quadratic, which is what we expected. The time spent by solving singular system and removal operator are very close, but the time spent by fixing DoFs are significantly higher, which mean the projections in each iteration didn't take too long compared to extra iterations.

4.2 Q1 Finite Element for Linear Elasticity Problem

The problem setup is as follows [11]:

$$f_1(x,y) = -\pi^2 \sin \pi x \sin \pi y + 2\pi^2(\frac{1}{\lambda} + 1)\cos \pi x \sin \pi y,$$
$$f_2(x,y) = -\pi^2 \cos \pi x \cos \pi y + 2\pi^2(\frac{1}{\lambda} + 1)\sin \pi x \cos \pi y.$$

The Neumann boundary conditions are:

$$g_1(x,y) = (-\frac{\pi}{\lambda}\cos(\pi x), 0), \quad g_2(x,y) = (\pi \sin \pi y, -\frac{\pi}{\lambda}\cos(\pi y)),$$
$$g_3(x,y) = (-\frac{\pi}{\lambda}\cos(\pi x), 0), \quad g_4(x,y) = (\pi \sin \pi y, -\frac{\pi}{\lambda}\cos(\pi y)).$$

The exact solution of the problem is:

$$u_1(x,y) = (-\sin \pi x + \frac{1}{\lambda}\cos \pi x)\sin \pi y + \frac{1}{4\pi^2},$$
$$u_2(x,y) = (-\cos \pi x + \frac{1}{\lambda}\sin \pi x)\cos \pi y.$$

If $\lambda = 2$ and $\mu = 0.5$, then the graphical output is:

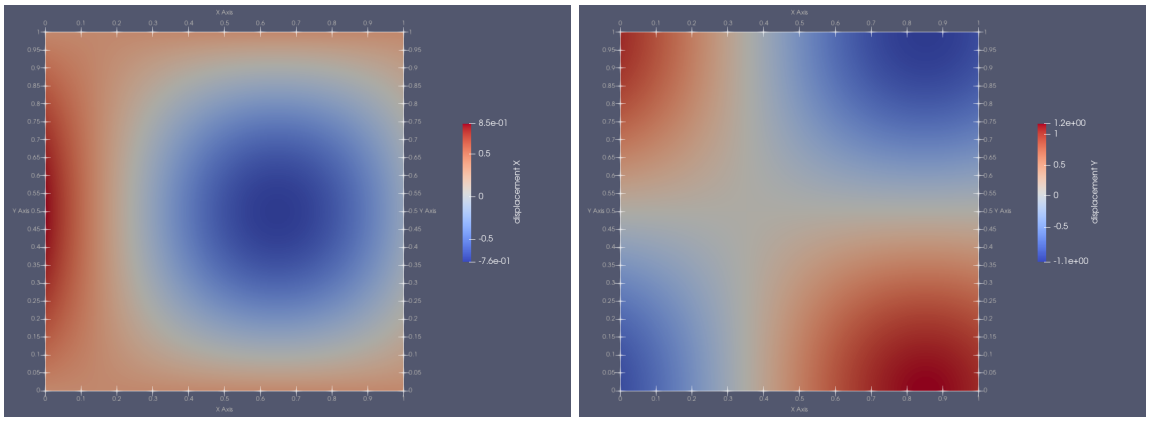

Figure 4.2: $u_1(x, y)$ on the left and $u_2(x, y)$ on the right

Q_1 finite element and global refinement is considered similar to Laplace problem. ILU preconditioner is applied for comparision purpose. Note, SSOR and other preconditioner can be applied as well. We will again compare between fixing DoF, solving singular system and and applying null space removal operator. For fixing dofs, we will fix dof corresponding x and y of $(0, 0)$ and dof corresponding to y of $(1, 0)$.

SSOR	Singular System				Removal Operator				Fix DoFs			
cells	error	RoC	CPU time	Iterations	error	RoC	CPU time	Iterations	error	RoC	CPU time	Iterations
4	0.263883	N/A	0.00	11	0.266663	N/A	0.00	11	0.266282	N/A	0.00	13
16	0.104667	2.52116713	0.00	19	0.107744	2.474968444	0.00	19	0.107724	2.471891129	0.00	26
64	0.036268	2.885932502	0.01	29	0.034054	3.163916133	0.02	29	0.034053	3.163421725	0.02	43
256	0.019274	1.881705925	0.09	54	0.010208	3.336010972	0.12	54	0.010208	3.335913009	0.15	75
1024	0.011842	1.62759669	0.67	98	0.003149	3.24166402	0.62	98	0.003149	3.24166402	0.98	140
4096	0.006766	1.750221697	4.49	186	0.001018	3.093320236	4.62	186	0.001018	3.093320236	6.52	270

Table 4.2: L_2 Error for Linear Elasticity problem with three approaches

We were unable to achieve the optimal rate for all three cases in our study. The primary reason for this outcome is that the manufactured solution is not exactly orthogonal to $\int_\Omega u_1 = 0$. More specifically, the inner product between the manufactured solution and the interpolation of $\int_\Omega u_1 = 0$ approaches zero with each cycle of global refinement. Consequently, even after projecting out $\int_\Omega u = 0$ and $\int_\Omega \nabla \times u = 0$ from our solution, non-zero inner products still introduce some error.

Furthermore, it is worth noting that the rate of convergence when solving the singular system is worse than when applying the removal operator. This observation suggests that solving the singular system is less robust compared to applying the removal operator. We will discuss this

idea in greater detail in the next subsection. Additionally, the time required for each method is similar to the numerical results obtained in the Laplace problem, where fixing DoFs takes longer due to the increased number of iterations required for the iterative solver to converge.

4.3 Q2-Q1 Finite Elements for Stokes Problem

The linearized Kovasznay problem is the follows: If Ψ is a stream function, such that

$$u = \frac{\partial \Psi}{\partial y}, \quad v = \frac{\partial \Psi}{\partial x}, \quad \omega = -\nabla^2 \Psi.$$

Let $\Psi_0 = y$, we can describe the two-dimensional flow as below:

$$\begin{aligned}
\Psi + \Psi_0 &= y - \frac{1}{2\pi} e^{\lambda x} \sin(2\pi y), \\
1 + u &= 1 - e^{\lambda x} \cos(2\pi y), \\
v &= -\frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y), \\
\omega &= \lambda R e^{\lambda x} \sin(2\pi y), \\
p &= \frac{e^{3\lambda} - e^{-\lambda}}{8\lambda} - \frac{e^{2\lambda x}}{2}.
\end{aligned}$$

Where R is the Reynolds number, $\lambda = R/2 - \sqrt{R^2/4 + 4\pi^2}$ and p is the pressure term. In this problem, we take $R = 10$. The domain we consider is the square: $[-0.5, 1.5] \times [-0.5, 1.5]$. GMRES is utilized to solve the linear system with block Schur complement preconditioner.

The exact solution is shown in the following figure:

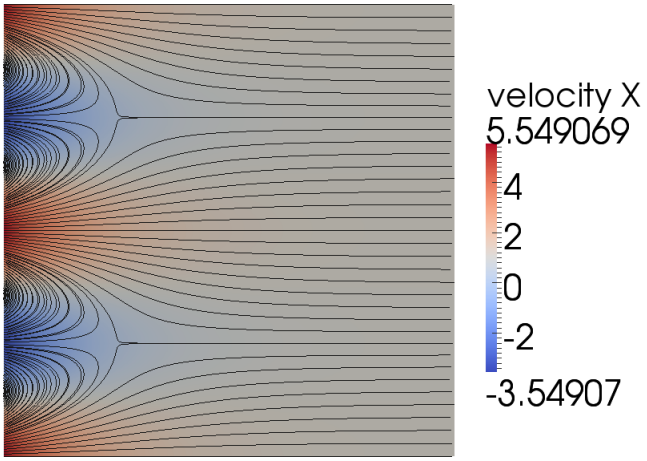

Figure 4.3: Numerical Result of Stokes Problem.

The red and blue color defines the velocity and the black lines are the streamline.

To define the preconditioner, recall the matrix form of Stokes problem is:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}.$$

Let $S = -BA^{-1}B^T$, then Schur complement preconditioner P_{Schur}^{-1} is:

$$P_{Schur}^{-1} = \begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix}.$$

The finite element we pick for Stokes problem is $Q_2 \times Q_1$ and global refinement is applied. We will use a CG solver with the mass matrix in the pressure space for approximating the action of S^{-1}. For the approximation of the velocity block A we will perform a single AMG V-cycle. We will test for three cases: solve preconditioned singular system, solve preconditoned singular system with null space removal operator and fix DoF. We will fix the first pressure DoF in the experiment.

The numerical output is the following:

	Singular System				
Cycle	Error of U	Error of P	RoC - P	Time	Iter
0	0.080650	0.348625	N/A	0.0158	56
1	0.007945	0.090797	3.84	0.0414	63
2	0.000806	0.022607	4.02	0.186	66
3	0.000093	0.005629	4.02	1.26	70
4	0.000011	0.001405	4.01	5.38	72
	Removal Operator				
Cycle	Error of U	Error of P	RoC - P	Time	Iter
0	0.080486	0.348565	N/A	0.0265	56
1	0.008031	0.090805	3.84	0.0458	63
2	0.000895	0.022612	4.02	0.173	66
3	0.000138	0.005631	4.02	1.28	70
4	0.000028	0.001406	4.00	5.73	72
	Fix DoFs				
Cycle	Error of U	Error of P	RoC - P	Time	Iter
0	0.080651	0.348625	N/A	0.045	112
1	0.007945	0.090797	3.84	0.0831	139
2	0.000806	0.022607	4.02	0.344	172
3	0.000093	0.005629	4.02	3.7	233
4	0.000011	0.001405	4.01	59.9	982

Table 4.3: L_2 Error for Stokes problem with three approaches

The mean value of pressure is computed after solving. We then removed the mean value of pressure from solution and have the table above. The number of iterations for fixing DoFs is slightly larger than all other methods. Time spent for fixing DoFs follows the pattern of iterations, which is also much higher compared to the other two methods. For each methods, we are expecting quadratic convergence and the results meet our expection.

4.4 Errors and Perturbations in Preconditioner

The singular system with preconditioner exhibit less robustness compared to the operators. However, this aspect has not been demonstrated in the numerical experiments section yet, as the singular system converges in each case. The preconditioners chosen for the experiments are well-defined, the range of the preconditioner is either identical to the range of matrix A or a subset of it.

In either case, convergence is guaranteed.

To emphasize the effect of round-off error and explore cases where the preconditioner maps onto the null space of A, we will manually add some error into the preconditioner. In Laplace problem, we used the SSOR preconditioner. To introduce error, we will utilize the SSOR preconditioner within a LinearOperator and add some errors to each entry in the product of matrix-vector multiplication (vmult) by preconditioner matrix. We then apply the removal operator to assess whether the perturbations continue to impact the solver.

The result can be summarized as follows:

Errors	Without Operator	Operator
1.00E-14	Convergence	Convergence
1.00E-13	Convergence	Convergence
1.00E-12	Convergence	Convergence
1.00E-11	Divergence	Convergence
1.00E-10	Divergence	Convergence
1.00E-09	Divergence	Convergence

Table 4.4: Effect of Pertubation in Laplace Problem

The perturbations on the right hand side has been tested in [26] and the result is similar: manually removing the null space in each iteration will remove the effect of perturbations. Consequently, if the singular system fails to converge, the removal operator can effectively resolve the problem. Therefore, it is reasonable to claim that null space removal operator should always be implemented even if the singular system can be solved, because it is more robust.

Chapter 5

Conclusions and Discussion

We discussed three problems in this thesis: Laplace problem, linear elasticity problem and Stokes Problem. We proved how all three problems can be well posed given such setup.

Then, we discussed how to solve these problems numerically. There are three major options: apply the method of fixing DoFs, solve the singular system or remove null space while solving. We then examined the computational properties for all methods.

Lastly, numerical experiments are presented. All three methods of removing null space for Laplace problem and linear elasticity problem are considered. We applied CG solver for Laplace problem and linear elasticity problem, then inserted operator into manually constructed preconditioner for Stokes problem and used GMRES solver.

To briefly summarize our findings in numerical experiment.

Condensing full constraints is very expensive in terms of computational costs.Therefore, it will not work for problems at large scale. Fix DoFs may require more iterations to converge but it takes less cost to setup. Solve singular system directly will work but not preferred since round off error or a preconditioner with larger range compared to A will result in divergence, see 4.4. Null space removal operator will prevent divergence in solving the singular system. But it will take more computation in each iteration and it requires the knowledge of the discrete basis of exact null space.

The future work of this thesis can be expanded in the following directions:

Instead of applying global refinement, an alternative approach worth exploring is the implementation of null space with adaptive refinement. Since the distribution of the grid is no longer uniform, accurate implementation poses additional challenges.

Another area for extension in this thesis is the behavior of multigrid or multilevel preconditioners. Predicting the performance of each method when a multigrid preconditioner is applied is difficult and requires further examination.

Bibliography

[1] J. Guermond A. Ern. *Theory and Practice of Finite Elements*. Springer, 2004.

[2] Daniel Arndt, Wolfgang Bangerth, Denis Davydov, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Jean-Paul Pelteret, Bruno Turcksin, and David Wells. The deal.II finite element library: Design, features, and insights. *Computers & Mathematics with Applications*, 81:407–422, 2021.

[3] Daniel Arndt, Wolfgang Bangerth, Marco Feder, Marc Fehling, Rene Gassmöller, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Peter Munch, Jean-Paul Pelteret, Simon Sticko, Bruno Turcksin, and David Wells. The `deal.II` library, version 9.4. *Journal of Numerical Mathematics*, 30(3):231–246, 2022.

[4] Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29, 1951.

[5] Ivo Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[6] Ivo Babuška. The rate of convergence for the finite element method. *SIAM Journal on Numerical Analysis*, 8(2):304–315, 1971.

[7] W. Bangerth. Deal.ii: The step-8 tutorial program. `https://www.dealii.org/current/doxygen/deal.II/step_8.html`, 2000.

[8] W. Bangerth and O. Kayser-Herold. Data structures and requirements for hp finite element software. *ACM Trans. Math. Softw.*, 36(1), mar 2009.

[9] Wolfgang Bangerth, Carsten Burstedde, Timo Heister, and Martin Kronbichler. Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Trans. Math. Softw.*, 38(2), jan 2012.

[10] Pavel Bochev and R. B. Lehoucq. On the finite element solution of the pure neumann problem. *SIAM Review*, 47(1):50–66, 2005.

[11] S. Brenner. A nonconforming mixed multigrid method for the pure traction problem in planar linear elasticity. *Mathematics of Computation*, 63:435–460, 1994.

[12] Philippe G Ciarlet. The finite element method for elliptic problems, vol. 40 of classics in applied mathematics, society for industrial and applied mathematics (siam), philadelphia, pa, 2002. reprint of the 1978 original, 1978.

[13] Philippe G Ciarlet. Mathematical elasticity. theory of plates, vol. ii. *Studies in Mathematics and its Applications*, 27, 1997.

[14] Martin Costabel and Monique Dauge. Crack singularities for general elliptic systems. *Mathematische Nachrichten*, 235(1):29–49, 2002.

[15] Michel Crouzeix and P-A Raviart. Conforming and nonconforming finite element methods for solving the stationary stokes equations i. *Revue française d'automatique informatique recherche opérationnelle. Mathématique*, 7(R3):33–75, 1973.

[16] deal.II Community. Compute mean value. `https://www.dealii.org/current/doxygen/deal.II/namespaceVectorTools.html#ad086eb08b8424fd7c853e389a3978a9a`.

[17] deal.II Community. Affine constraint. `https://www.dealii.org/current/doxygen/deal.II/classAffineConstraints.html`, 2016.

[18] Howard C Elman. *Iterative methods for large, sparse, nonsymmetric systems of linear equations*. Yale University, 1982.

[19] Richard S Falk. Nonconforming finite element methods for the equations of linear elasticity. *mathematics of computation*, 57(196):529–550, 1991.

[20] Roger Fletcher and Michael JD Powell. A rapidly convergent descent method for minimization. *The computer journal*, 6(2):163–168, 1963.

[21] Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

[22] Matthias Hieber and Jürgen Saal. *The Stokes Equation in the Lp-Setting: Well-Posedness and Regularity Properties*, pages 117–206. Springer International Publishing, Cham, 2018.

[23] Jason S. Howell. Prestructuring sparse matrices with dense rows and columns via null space methods. *Numerical Linear Algebra with Applications*, 25(2):e2133, 2018. e2133 nla.2133.

[24] K. Sugihara K. Hayami. Gmres on singular systems revisited. *arXiv*, 2009.

[25] E. Kaasschieter. A practical termination criterion for the conjugate gradient method. *BIT Numerical Mathematics*, 28:308–322, 1988.

[26] E. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *Journal of Computational and Applied Mathematics*, 24:265–275, 1988.

[27] Chang-Ock Lee. A conforming mixed finite element method for the pure traction problem of linear elasticity. *Applied Mathematics and Computation*, 93(1):11–29, 1998.

[28] Matthias Maier, Mauro Bardelloni, and Luca Heltai. Linearoperator—a generic, high-level expression syntax for linear algebra. *Computers & Mathematics with Applications*, 72(1):1–24, 2016.

[29] A Perry. A modified cg algorithm. Technical report, Discussion Paper 229, Center for Mathematical Studies in Economics, 1976.

[30] John Ker Reid. *Large Sparse Sets of Linear Equations: proceedings of the Oxford conference of the Institute of Mathematics and Its Applications held in April, 1970*. Academic Press, 1971.

[31] Youcef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. ScI. STAT. COMPUT.*, 7(3):856–869, 1986.

[32] Yousef Saad. Variations on arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear algebra and its applications*, 34:269–295, 1980.

[33] Ridgway Scott. Interpolated boundary conditions in the finite element method. *SIAM Journal on Numerical Analysis*, 12(3):404–427, 1975.

[34] Gilbert Strang, George J Fix, et al. An analysis of the finite element method. 1969.