MULTILEVEL GRADIENT UZAWA ALGORITHMS FOR SYMMETRIC SADDLE POINT PROBLEMS

CONSTANTIN BACUTA AND LU SHU

ABSTRACT. In this paper, we introduce a general multilevel gradient Uzawa algorithm for symmetric saddle point systems. We compare its performance with the performance of the standard Uzawa multilevel algorithm. The main idea of the approach is to combine a double inexact Uzawa algorithm at the continuous level with a gradient type algorithm at the discrete level. The algorithm is based on the existence of a priori multilevel sequences of nested approximation pairs of spaces, but the family does not have to be stable. To ensure convergence, the process has to maintain an accurate representation of the residuals at each step of the inexact Uzawa algorithm at the continuous level. The residual representations at each step are approximated by projections or representation operators. Sufficient conditions for ending the iteration on a current pair of discrete spaces are determined by computing simple indicators that involve consecutive iterations. When compared with the standard Uzawa multilevel algorithm, our proposed algorithm has the advantages of automatically selecting the relaxation parameter, lowering the number of iterations on each level, and improving on running time. By carefully choosing the discrete spaces and the projection operators, the error for the second component of the solution can be significantly improved even when comparison is made with the discretization on standard families of stable pairs.

1. INTRODUCTION

A survey about numerical solutions of discrete saddle point problems (SPP) was done by Benzi, Golub and Liesen, in [14]. Recent state of the art solvers for the Stokes and Navier Stokes systems are presented in the work of Silvester, Wathen and Elman, [31]. In this paper, we propose to discretize Saddle Point Problems (SPPs) that have symmetric and coercive form $a(\cdot,\cdot)$ by using inexact Uzawa type algorithms at the continuous level. The inexact Uzawa algorithm was introduced for the finite dimensional case in [23, 30], and for the infinite dimensional case (or continuous level) in [4, 11, 13, 28]. Our approach is based on a double Inexact Uzawa (IU) type algorithm at the continuous level, as introduced and analyzed in [11]. The idea of using an inexact Uzawa algorithm at the continuous level is not new

2000 *Mathematics Subject Classification.* 74S05, 74B05, 65N22, 65N55.

Key words and phrases. inexact Uzawa algorithms, symmetric saddle point system, multilevel methods, gradient methods.

This work was partially supported by NSF, DMS-0713125.

and is presented in the adaptive context in [13, 28, 36, 37, 41]. In the present work we interpret the double IU algorithm as a *multilevel* algorithm and do not use *a posteriori error estimators*. The main advantage of the resulting multilevel algorithms is that, in spite of the absence of a discrete (LBB) condition, we could get better rates of convergence for the second variable. In the case of Stokes discretization, e.g., the convergence order for the second variable improves from $O(h)$ for the standard stable discretization to almost $O(h^2)$ for our *multilevel and residual smoothing* discretization method (see Section 6). We observed similar behavior for the div-curl systems in [11]. In the case of the multilevel gradient algorithm, the relaxation parameters are computed automatically by the algorithm, and the number of iterations on each level is reduced, when compared to the similar Uzawa algorithms. The proposed approach is based on the presence of a multilevel structure, as presented in [6, 22, 24] and it is also related to ideas used in the cascadic multilevel (multigrid), [15, 17, 18, 19]. For the cascadic multilevel methods the level change criteria is based on keeping the iteration error close to the expected discretization error and a discrete stability condition is needed. In our approach, the level change criteria is based on the convergence of the inexact Uzawa algorithm at the continuous level, and a discrete stability condition is not required.

We test the performance of our algorithms on solving standard Stokes equations, but the algorithms can be applied to a large class of problems that can be reformulated at the continuous level as symmetric SPP, including e.g, div-curl systems or Maxwell equations (see [20, 21]).

The rest of the paper is organized as follows. In Section 2, we introduce the notation and review some properties of the Schur operators. Based on the Schur complement properties, in Section 3 we review the main convergence results about Uzawa and gradient algorithms and prove a sharp error reduction estimate for the Uzawa gradient algorithm in the general infinite dimensional case. In Section 4, we present a new approach in building discrete spaces for discretizing SPPs using Uzawa or gradient algorithms. The Multilevel Inexact Uzawa (MIU) or gradient algorithms are discussed in Section 5. In Section 6 we present numerical results for the Stokes system and compare the performance of Uzawa and gradient algorithms for four different choices of discrete spaces for the pressure. We summarize our conclusions in Section 7.

2. Properties of Schur complements

In this section, we start with a review of the notation of the classical SPP theory and introduce the spaces, the operators and the norms for the general abstract case. We let $\mathbf{V}$ and Q be two Hilbert spaces with inner products $a_0(\cdot,\cdot)$ and $(\cdot,\cdot)$ respectively, with the corresponding induced norms $|\cdot|_{\mathbf{V}} = |\cdot| = a_0(\cdot,\cdot)^{1/2}$ and $\|\cdot\|_Q = \|\cdot\| = (\cdot,\cdot)^{1/2}$. The dual pairings on $\mathbf{V}^* \times \mathbf{V}$ and $Q^* \times Q$ are denoted by $\langle\cdot,\cdot\rangle$. Here, $\mathbf{V}^*$ and Q^* denote the

duals of $\mathbf{V}$ and Q, respectively. With the inner products $a_0(\cdot,\cdot)$ and $(\cdot,\cdot)$, we associate the operators
$A_0 : V \to V^*$ and $\mathcal{C} : Q \to Q^*$ defined by

$$\langle A_0\mathbf{u}, \mathbf{v}\rangle = a_0(\mathbf{u}, \mathbf{v}) \quad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}$$

and

$$\langle \mathcal{C}p, q\rangle = (p, q) \quad \text{for all } \ p, q \in Q.$$

The operators ${A_0}^{-1} : V^* \to V$ and $\mathcal{C}^{-1} : Q^* \to Q$ are called the Riesz-canonical isometries and satisfy

$$a_0({A_0}^{-1}\mathbf{u}^*, \mathbf{v}) = \langle \mathbf{u}^*, \mathbf{v}\rangle, \ \ |{A_0}^{-1}\mathbf{u}^*|_{\mathbf{V}} = \|\mathbf{u}^*\|_{\mathbf{V}^*}, \mathbf{u}^* \in \mathbf{V}^*, \mathbf{v} \in \mathbf{V}, \tag{2.1}$$

$$(\mathcal{C}^{-1}p^*, q) = \langle p^*, q\rangle, \ \ \|\mathcal{C}^{-1}p^*\| = \|p^*\|_{Q^*}, \ p^* \in Q^*, q \in Q. \tag{2.2}$$

Next, we suppose that $b(\cdot,\cdot)$ is a continuous bilinear form on $\mathbf{V} \times Q$, satisfying the inf-sup condition. More precisely, we assume that

$$\inf_{p\in Q} \sup_{\mathbf{v}\in\mathbf{V}} \frac{b(\mathbf{v}, p)}{\|p\| \ |\mathbf{v}|} = m > 0 \tag{2.3}$$

and

$$\sup_{p\in Q} \sup_{\mathbf{v}\in\mathbf{V}} \frac{b(\mathbf{v}, p)}{\|p\| \ |\mathbf{v}|} = M < \infty. \tag{2.4}$$

Here, and throughout this paper, the "inf" and "sup" are taken over nonzero vectors. With the form b, we associate the linear operators $B : V \to Q^*$ and $B^* : Q \to V^*$ defined by

$$\langle B\mathbf{v}, q\rangle = b(\mathbf{v}, q) = \langle B^*q, \mathbf{v}\rangle \quad \text{for all } \mathbf{v} \in \mathbf{V}, \ q \in Q.$$

Let $\mathbf{V}_0$ be the kernel of B or $\mathcal{C}^{-1}B$, i.e.,

$$\mathbf{V}_0 = Ker(B) = \{\mathbf{v} \in \mathbf{V}| \ B\mathbf{v} = 0\} = \{\mathbf{v} \in \mathbf{V}| \ \mathcal{C}^{-1}B\mathbf{v} = 0\}.$$

Due to (2.4), $\mathbf{V}_0$ is a closed subspace of $\mathbf{V}$. Next, we review the Schur complement operators as introduced in [4].

The Schur complement on Q is the operator
$S_0 := \mathcal{C}^{-1}B{A_0}^{-1}B^* : Q \to Q$. The operator S_0 is symmetric and positive definite on Q, satisfying

$$\sigma(S_0) \subset [m^2, M^2], \ \text{and } m^2, M^2 \in \sigma(S_0). \tag{2.5}$$

Here, $\sigma(S_0)$ denotes the spectrum of the operator S_0. Consequently,

$$\|p\|_{S_0} := (S_0p, p)^{1/2} = |{A_0}^{-1}B^*p|_{\mathbf{V}} \geq m\|p\| \quad \text{for all } \ p \in Q. \tag{2.6}$$

The Schur complement on $\mathbf{V}$ is defined as the operator
$S := {A_0}^{-1}B^*\mathcal{C}^{-1}B : \mathbf{V} \to \mathbf{V}$. The operator S is symmetric and non-negative definite on $\mathbf{V}$, with $Ker(S) = \mathbf{V}_0$, $S(\mathbf{V}) = \mathbf{V}_1$. *The Schur complement on* $\mathbf{V}_1 = \mathbf{V}_0^{\perp}$ is the restriction of S to $\mathbf{V}_1$, i.e., $S_1 := {A_0}^{-1}B^*\mathcal{C}^{-1}B : \mathbf{V}_1 \to \mathbf{V}_1$. The operator S_1 is symmetric and positive definite on $\mathbf{V}_1$, satisfying

$$\sigma(S_1) = \sigma(S_0) \subset [m^2, M^2]. \tag{2.7}$$

3. Uzawa and gradient Algorihms

The Uzawa algorithm for solving the Stokes system was first introduced in [1]. In this section, we present the connection between the Schur complements and the Uzawa or Uzawa gradient iterations. First, we *review* the Uzawa algorithm for the symmetric saddle point problem. We assume that the bilinear form $a(\cdot,\cdot)$ is defined on $\mathbf{V}\times\mathbf{V}$ and that the form is *symmetric, bounded and coercive on the whole space* $\mathbf{V}$. For $f\in\mathbf{V}^*$, $g\in Q^*$, we consider the following variational problem: Find $(\mathbf{u},p)\in\mathbf{V}\times Q$ such that

$$\begin{array}{llll} a(\mathbf{u},\mathbf{v}) & + \quad b(\mathbf{v},p) & =\langle \mathbf{f},\mathbf{v}\rangle & \text{for all } \mathbf{v}\in\mathbf{V},\\ b(\mathbf{u},q) & & =\langle g,q\rangle & \text{for all } q\in Q, \end{array} \tag{3.1}$$

where the bilinear form $b:\mathbf{V}\times Q\to\mathbb{R}$ satisfies (2.3) and (2.4). If $A:V\to V^*$ is the standard operator associated with the form $a(\cdot,\cdot)$, then the problem (3.1) is equivalent to: Find $(\mathbf{u},p)\in\mathbf{V}\times Q$ such that

$$\begin{array}{lll} A\mathbf{u} & + \quad B^*p & =\mathbf{f},\\ B\mathbf{u} & & =g. \end{array} \tag{3.2}$$

It is known that the above variational problem or system has a unique solution for any $f\in\mathbf{V}^*$, $g\in Q^*$ (see [2, 16, 25, 26, 27, 33, 39]).

The form $a(\cdot,\cdot)$ might not agree with the natural inner product on $\mathbf{V}$, but due to the symmetry, boundness, and the coercivity assumptions, $a(\cdot,\cdot)$ induces a norm that is equivalent with the natural norm on $\mathbf{V}$. Thus, without loss of generality, we assume from now on that the inner product $a_0(\cdot,\cdot)$ on $\mathbf{V}$ is the same as $a(\cdot,\cdot)$, and consequently, $A=A_0$.

Given a parameter $\alpha>0$, called the relaxation parameter, the Uzawa algorithm for approximating the solution $(\mathbf{u},p)$ of (3.1) is:

Algorithm 3.1 (Uzawa Method **(UM)**)**.** *Let p_0 be any initial guess for p. For $j=1,2,\dots,$ construct $(\mathbf{u}_j,p_j)$ given by*

$$\begin{aligned} \mathbf{u}_j &= A^{-1}(\mathbf{f}-B^*p_{j-1}),\\ p_j &= p_{j-1}+\alpha\mathcal{C}^{-1}(B\mathbf{u}_j-g). \end{aligned} \tag{3.3}$$

The convergence of the **UM** is discussed in many publications, e.g., [14, 26, 32, 33, 38]. It is easy to check that

$$p-p_j=(I-\alpha\mathcal{C}^{-1}BA^{-1}B^*)(p-p_{j-1})=(I-\alpha S_0)(p-p_{j-1}). \tag{3.4}$$

Thus, from (2.5), we get that the convergence rate of **UM** is

$$\|I-\alpha S_0\|=\max\{|1-\alpha m^2|,|1-\alpha M^2|\}<1, \text{ provided } \alpha\in\left(0,\frac{2}{M^2}\right). \tag{3.5}$$

For $j\ge 1$ we also have

$$\mathbf{u}-\mathbf{u}_{j+1}=(I-\alpha A^{-1}B^*\mathcal{C}^{-1}B)(\mathbf{u}-\mathbf{u}_j)=(I-\alpha S)(\mathbf{u}-\mathbf{u}_j) \tag{3.6}$$

To anticipate the connection with the Uzawa Gradient (UG) method, we introduce, $\mathbf{w}_1:=\mathbf{u}_1$ and for any $j\ge 1$, $\mathbf{w}_{j+1}:=\mathbf{u}_{j+1}-\mathbf{u}_j$,
$q_j:=\mathcal{C}^{-1}(B\mathbf{u}_j-g)$, and $\mathbf{h}_j:=-A^{-1}B^*q_j$.

With the above notation, then we have: $p_j = p_{j-1} + \alpha\, q_j$ and

$$\mathbf{w}_{j+1} = -A^{-1}B^*(p_j - p_{j-1}) = -\alpha A^{-1}B^* q_j = \alpha \mathbf{h}_j.$$

Thus,

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{w}_{j+1} = \mathbf{u}_j + \alpha\, \mathbf{h}_j. \tag{3.7}$$

From (3.6), we get

$$\mathbf{w}_{j+2} = (I - \alpha S)\, \mathbf{w}_{j+1} = (I - \alpha S_1)\, \mathbf{w}_{j+1}, \qquad \text{for all} \quad j \geq 1, \tag{3.8}$$

and from (3.7), and the definition of q_{j+1} we have

$$q_{j+1} = (I - \alpha S_0)\, q_j, \qquad \text{for all} \quad j \geq 1. \tag{3.9}$$

Next, we relate q_j and the error $p - p_{j-1}$ using the Schur complement S_0. First we have $q_1 = \mathcal{C}^{-1}(B\mathbf{u}_1 - g) = \mathcal{C}^{-1}B(\mathbf{u}_1 - \mathbf{u})$, and

$$\mathbf{u}_1 = A^{-1}(\mathbf{f} - B^* p_0) = A^{-1}(A\mathbf{u} + B^*p - B^*p_0) = \mathbf{u} + A^{-1}B^*(p - p_0).$$

Thus, $\mathbf{u}_1 - \mathbf{u} = A^{-1}B^*(p - p_0)$ and

$$q_1 = \mathcal{C}^{-1}BA^{-1}B^*(p - p_0) = S_0(p - p_0).$$

Using induction and (3.9) we have

$$q_{j+1} = S_0(p - p_j), \qquad \text{for all } j \geq 0. \tag{3.10}$$

For the multilevel version of Uzawa Algorithm (**UA**) in Section 5, next, we need to estimate the ratios $R_j := \frac{\|p_j - p_{j-1}\|}{|\mathbf{u}_{j+1} - \mathbf{u}_j|} = \frac{\|p_j - p_{j-1}\|}{|\mathbf{w}_{j+1}|}$, and $r_j := \frac{|\mathbf{w}_j|}{|\mathbf{w}_{j+1}|}$. From (3.7), and (2.6), we get

$$R_j = \frac{\|p_j - p_{j-1}\|}{|\mathbf{w}_{j+1}|} = \frac{\alpha\, \|q_j\|}{\alpha\, |\mathbf{h}_j|} = \frac{\|q_j\|}{(S_0 q_j, q_j)^{1/2}}. \tag{3.11}$$

Using the spectral propperties of S_0 we get that

$$R_j^2 \in \left[\frac{1}{M^2}, \frac{1}{m^2}\right]. \tag{3.12}$$

In addition, in the case when $\mathbf{V}$ and Q are finite dimensional spaces, from (3.9) and (3.8) we get that

$$R_j^2 \nearrow \frac{1}{m^2}, \text{ and } r_j \searrow \frac{1}{1 - \alpha m^2}, \text{ for } 0 < \alpha \leq \frac{2}{m^2 + M^2}, \tag{3.13}$$

and

$$R_j^2 \searrow \frac{1}{M^2}, \text{ and } r_j \searrow \frac{1}{\alpha M^2 - 1}, \text{ for } \frac{2}{m^2 + M^2} < \alpha < \frac{2}{M^2}. \tag{3.14}$$

Thus, R_j or r_j can be used to estimate m^2 and M^2 the extreme eigenvalues of S_0 (or S_1). Here, the symbol $\searrow$ indicates that the convergence is from the right and it dos not necessarily mean that we are dealing with a monotonic sequence. A similar interpretation is valid for the symbol $\nearrow$.

With the above notation and considerations we reformulate the Uzawa algorithm as follows:

Algorithm 3.2. Uzawa Algorithm (UA)

Step 1: Set $\mathbf{u}_0 = 0 \in \mathbf{V}$, $p_0 \in Q$, *and* **compute** $\mathbf{u}_1 \in \mathbf{V}$ *and* $q_1 \in Q$ *by*

$$a(\mathbf{u}_1, \mathbf{v}) = < \mathbf{f}, \mathbf{v} > - b(\mathbf{v}, p_0), \ \ \mathbf{v} \in \mathbf{V}$$
$$q_1 := \mathcal{C}^{-1}(B\mathbf{u}_1 - g).$$

Step 2: *For* $j = 1, 2, \ldots,$ **compute** $\mathbf{h}_j, \mathbf{u}_{j+1} \in \mathbf{V}$ *and* $p_j, q_{j+1} \in Q$ *by*

$$(\mathbf{U1}) \qquad a(\mathbf{h}_j, \mathbf{v}) = -b(\mathbf{v}, q_j), \ \ \mathbf{v} \in \mathbf{V}$$
$$(\mathbf{U2}) \qquad p_j := p_{j-1} + \alpha \ q_j$$
$$(\mathbf{U3}) \qquad \mathbf{u}_{j+1} = \mathbf{u}_j + \alpha \ \mathbf{h}_j$$
$$(\mathbf{U4}) \qquad q_{j+1} := \mathcal{C}^{-1}(B\mathbf{u}_{j+1} - g)$$

End

Next, we present the Uzawa Gradient algorithm as an Uzawa algorithm with variable relaxation parameter α. For the finite dimensional case, the algorithm is presented by Braess in [16].

Algorithm 3.3. Uzawa Gradient Algorithm (UGA)

Step 1: Set $\mathbf{u}_0 = 0 \in \mathbf{V}$, $p_0 \in Q$, *and* **compute** $\mathbf{u}_1 \in \mathbf{V}$ *and* $q_1 \in Q$ *by*

$$a(\mathbf{u}_1, \mathbf{v}) = < \mathbf{f}, \mathbf{v} > - b(\mathbf{v}, p_0), \ \ \mathbf{v} \in \mathbf{V}$$
$$q_1 := \mathcal{C}^{-1}(B\mathbf{u}_1 - g).$$

Step 2: For $j = 1, 2, \ldots,$ **compute** $\mathbf{h}_j, \mathbf{u}_{j+1} \in \mathbf{V}$ *and* $p_j, q_{j+1} \in Q$ *by*

$$(\mathbf{UG1}) \qquad a(\mathbf{h}_j, \mathbf{v}) = -b(\mathbf{v}, q_j), \ \ \mathbf{v} \in \mathbf{V}$$
$$(\mathbf{UG}\alpha) \qquad \alpha_j = -\frac{(q_j, q_j)}{b(\mathbf{h}_j, q_j)} = \frac{(q_j, q_j)}{(S_0 q_j, q_j)}$$
$$(\mathbf{UG2}) \qquad p_j := p_{j-1} + \alpha_j \ q_j$$
$$(\mathbf{UG3}) \qquad \mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_j \ \mathbf{h}_j$$
$$(\mathbf{UG4}) \qquad q_{j+1} := \mathcal{C}^{-1}(B\mathbf{u}_{j+1} - g)$$

End

Next, in order to analyze the algorithm, for $j \geq 0$ we define $\mathbf{w}_{j+1}$ by

$$a(\mathbf{w}_{j+1}, \mathbf{v}) = \langle f, \mathbf{v} \rangle - a(\mathbf{u}_j, \mathbf{v}) - b(\mathbf{v}, p_j), \qquad \text{for all } \mathbf{v} \in \mathbf{V}. \tag{3.15}$$

Using induction over $j \geq 0$, it is easy to check that

$$\mathbf{w}_{j+1} = \mathbf{u}_{j+1} - \mathbf{u}_j, \tag{3.16}$$

and for $j \geq 1$, using **(UG3)**, we also have

$$\mathbf{w}_{j+1} = \alpha_j \ \mathbf{h}_j. \tag{3.17}$$

From (**UG1**), (**UG3**) and (**UG4**) we obtain

$$q_{j+1} = (I - \alpha_j S_0)\ q_j, \qquad \text{for all} \quad j \geq 1, \tag{3.18}$$

which proves that the choice of α_j in **(UGα)** implies

$$(q_j, q_{j+1}) = 0, \ \ j \geq 1. \tag{3.19}$$

The next result deals with the convergence of the **UGA** and provides estimates for the error $\mathbf{u} - \mathbf{u}_j$, the pressure error $p - p_j$ and the residual $\mathbf{w}_{j+1}$ defined in (3.15). The result holds for the infinite dimensional case, and, by the authors' best knowledge, the formula for error reduction is new.

Theorem 3.4. *The sequence* $(\mathbf{u}_j, p_j)$ *produced by* **UGA** *converges to* $(\mathbf{u}, p)$ *the solution of* (3.1)*. In addition, the following formulas hold:*

$$\mathbf{u}_{j+1} - \mathbf{u} = A^{-1}B^*(p - p_j), \quad j = 0, 1, \cdots, \ \textit{and} \tag{3.20}$$

$$|\mathbf{u} - \mathbf{u}_j|^2 - |\mathbf{u} - \mathbf{u}_{j+1}|^2 = \|p - p_{j-1}\|^2_{S_0} - \|p - p_j\|^2_{S_0} = |\mathbf{w}_{j+1}|^2. \tag{3.21}$$

Proof. Since **Step 1** of **UA** and **Step 1** of **UGA** are identical we have $q_1 = S_0(p - p_0)$, and by induction, it follows that (3.10) remains valid for the **UGA**. Thus, from **(UG2)** we get

$$p - p_j = p - p_{j-1} - \alpha_j S_0(p - p_{j-1}),$$

with

$$\alpha_j = \frac{(q_j, q_j)}{(S_0 q_j, q_j)} = \frac{(p - p_{j-1}, S_0(p - p_{j-1}))_{S_0}}{(S_0(p - p_{j-1}), S_0(p - p_{j-1}))_{S_0}},$$

which says that $p - p_j$ is exactly the difference between $p - p_{j-1}$ and the orthogonal projection of $S_0(p - p_{j-1})$ onto $p - p_{j-1}$ with respect to the $(\cdot, \cdot)_{S_0}$ -inner product. Consequently we have

$$\|p - p_{j-1}\|^2_{S_0} = \|p - p_j\|^2_{S_0} + \alpha_j^2 \|S_0(p - p_{j-1})\|^2_{S_0},$$

which leads to the following error reduction formula

$$\|p - p_j\|^2_{S_0} = \|p - p_{j-1}\|^2_{S_0} - \frac{(q_j, q_j)^2}{(S_0 q_j, q_j)}, \ \text{or} \tag{3.22}$$

$$\|p - p_j\|^2_{S_0} = \|p - p_{j-1}\|^2_{S_0} \left(1 - \frac{(q_j, q_j)^2}{(S_0^{-1}\ q_j, q_j)(S_0 q_j, q_j)} \right). \tag{3.23}$$

Using the Kantorowitch inequality

$$\frac{(S_0^{-1}\ q_j, q_j)(S_0 q_j, q_j)}{(q_j, q_j)^2} \leq \frac{1}{4} \left(\sqrt{\xi} + \frac{1}{\sqrt{\xi}} \right)^2,$$

where $\xi := \frac{\max(\sigma(S_0))}{\min(\sigma(S_0))} = \frac{M^2}{m^2}$, we get

$$\|p - p_j\|_{S_0} \le \frac{M^2 - m^2}{M^2 + m^2} \, \|p - p_{j-1}\|_{S_0}. \tag{3.24}$$

Next, we will justify (3.21). From **Step 1** of **UGA** and the first equation of (3.1), we get

$$a(\mathbf{u}_1 - \mathbf{u}, \mathbf{v}) = b(\mathbf{v}, p_0 - p), \ \mathbf{v} \in \mathbf{V}.$$

Using the equations of **Step 2** in **UGA**, and induction over $j \ge 0$ we get that

$$a(\mathbf{u}_{j+1} - \mathbf{u}, \mathbf{v}) = b(\mathbf{v}, p_j - p), \ \mathbf{v} \in \mathbf{V}, \ j = 0, 1, \cdots,$$

which proves (3.21). Combining (3.21) with the identity (2.6), we get

$$|\mathbf{u} - \mathbf{u}_{j+1}| = \|p - p_j\|_{S_0}. \tag{3.25}$$

From (3.24) and (3.25) we get that $p_j \to p$ and $\mathbf{u}_j \to \mathbf{u}$.

To end the proof, we have to justify the error reduction formula (3.21). By taking $\mathbf{v} = \mathbf{h}_j$ in (**UG1**) and using (**UG**α) and (3.17), we get

$$\alpha_j = \frac{(q_j, q_j)}{|\mathbf{h}_j|^2} = \frac{\|q_j\|^2}{|\mathbf{w}_{j+1}|^2} {\alpha_j}^2,$$

which gives

$$\alpha_j = \frac{|\mathbf{w}_{j+1}|^2}{\|q_j\|^2}, \quad \text{and} \tag{3.26}$$

$$\frac{(q_j, q_j)^2}{(S_0 q_j, q_j)} = \alpha_j \|q_j\|^2 = |\mathbf{w}_{j+1}|^2.$$

The above identity together with (3.22) and (3.25) lead to (3.21).

□

For the multilevel version **UGA** in Section 5, now we need to estimate the ratio $R_j := \frac{\|p_j - p_{j-1}\|}{|\mathbf{u}_{j+1} - \mathbf{u}_j|} = \frac{\|p_j - p_{j-1}\|}{|\mathbf{w}_{j+1}|}$. From (3.17), (2.6), **(UG2)**, and **(UG**α**)** we get

$$R_j = \frac{\|p_j - p_{j-1}\|}{|\mathbf{w}_{j+1}|} = \frac{\alpha_j \ \|q_j\|}{\alpha_j \ |\mathbf{h}_j|} = \frac{\|q_j\|}{(S_0 q_j, q_j)^{1/2}} = \sqrt{\alpha_j}. \tag{3.27}$$

Using the spectral propperties of S_0 we get that

$$R_j^2 = \alpha_j \in \left[\frac{1}{M^2}, \frac{1}{m^2} \right]. \tag{3.28}$$

Each step of the **UGA** is as a particular step of the Uzawa algorithm provided that for some fixed $\alpha^0, \alpha^1 \in (0, 2/M^2)$ we have

$$\alpha^0 \le \alpha_j = R_j^2 \le \alpha^1. \tag{3.29}$$

The restriction (3.29) can be used as a stopping criteria for a multilevel **UGA** that is viewed as a double inexact Uzawa algorithm at the continuous level as we will proceed in Section 5. Under the assumption (3.29), from (3.18), (2.5), and (3.28), it follows that there is $\rho_0 \in (0, 1)$ such that:

$$\frac{\|q_{j+1}\|}{\|q_j\|} \le \rho_0 < 1. \tag{3.30}$$

Equation (3.30) can be also used as stopping criteria for a multilevel **UGA**.

4. A new look at discretization of saddle point problems

Let $\mathbf{V}_h$ be a subset of $\mathbf{V}$ and let $\tilde{M}_h$ be a finite dimensional subspace of Q. We consider that $\tilde{M}_h$ is a Hilbert space equipped with an inner product $(\cdot, \cdot)_h$ which might not agree with the restriction of the original inner product $(\cdot, \cdot)$ to $\tilde{M}_h \times \tilde{M}_h$. First, we define the representation operator $\mathcal{R}_h : Q \to \tilde{M}_h$ by

$$(\mathcal{R}_h p, \tilde{q}_h)_h = (p, \tilde{q}_h), \qquad \text{for all } \ \tilde{q}_h \in \tilde{M}_h, \tag{4.1}$$

i.e., $\mathcal{R}_h p$ is the Riesz representation of $\tilde{q}_h \to (p, \tilde{q}_h)$ as a functional on $(\tilde{M}_h, (\cdot, \cdot)_h)$. We notice that in the particular case when $(\cdot, \cdot)_h$ agrees with the inner product on Q, we have that $\mathcal{R}_h$ becomes the orthogonal projection onto $\tilde{M}_h$. Next, we define M_h by

$$M_h := \mathcal{R}_h \mathcal{C}^{-1} B \mathbf{V}_h$$

and equip M_h with the same inner product $(\cdot, \cdot)_h$. We consider the restrictions of the forms a and b to the discrete spaces $\mathbf{V}_h$ and M_h and define the corresponding discrete operators $A_h, \mathcal{C}_h, B_h$. For example, A_h is the discrete version of $A_0 = A$, and is defined by

$$\langle A_h \ \mathbf{u}_h, \mathbf{v}_h \rangle = a(\mathbf{u}_h, \mathbf{v}_h), \qquad \text{for all } \mathbf{u}_h \in \mathbf{V}_h, \mathbf{v}_h \in \mathbf{V}_h.$$

For any $q_h \in M_h, \mathbf{v}_h \in \mathbf{V}_h$ we have

$$b(\mathbf{v}_h, q_h) = \langle B_h \mathbf{v}_h, q_h \rangle = (\mathcal{C}_h^{-1} B_h \mathbf{v}_h, q_h)_h.$$

On the other hand, since $q_h \in M_h \subset Q$, and $\mathbf{v}_h \in \mathbf{V}_h \subset \mathbf{V}$, we have

$$b(\mathbf{v}_h, q_h) = (\mathcal{C}^{-1} B \mathbf{v}_h, q_h) = (R_h \mathcal{C}^{-1} B \mathbf{v}_h, q_h)_h.$$

From the above identities, we get

$$\mathcal{C}_h^{-1} B_h \mathbf{v}_h = R_h \mathcal{C}^{-1} B \mathbf{v}_h \qquad \text{for all } \mathbf{v}_h \in \mathbf{V}_h. \tag{4.2}$$

This implies that $\mathcal{C}_h^{-1} B_h$ is onto M_h and, using that $\mathbf{V}_h$ and M_h are finite dimensional spaces, a discrete inf-sup condition holds, i.e., there exists $m_h > 0$ such that

$$\inf_{p_h \in M_h} \sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, p_h)}{\|p_h\|_h \ |\mathbf{v}_h|} = m_h > 0. \tag{4.3}$$

We also introduce M^h, the norm of $b(\cdot, \cdot)$ on $\mathbf{V}_h \times M_h$, by

$$\sup_{p_h \in M_h} \sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, p_h)}{\|p_h\|_h \ |\mathbf{v}_h|} := M^h. \tag{4.4}$$

Here, we have $\|p_h\|_h := (p_h, p_h)_h^{1/2}$.

Thus, the problem: Find $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times M_h$ such that

$$
\begin{array}{llll}
(4.5) & a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) & = \langle \mathbf{f}, \mathbf{v}_h \rangle & \text{for all } \mathbf{v}_h \in \mathbf{V}_h, \\
& b(\mathbf{u}, q_h) & = \langle g, q_h \rangle & \text{for all } q_h \in M_h,
\end{array}
$$

has unique solution $(\mathbf{u}_h, p_h)$. Sufficient conditions that guarantee the convergence of $(\mathbf{u}_h, p_h)$ to the continuous solution $(\mathbf{u}, p)$ are well known, see [16, 25, 40, 42]. We also note that

$$\mathcal{C}_h^{-1} g = R_h \mathcal{C}^{-1} g,$$

where g in the left side is viewed as the functional $q_h \to \langle g, q_h \rangle$ on M_h. Thus, both **UA** and **UGA** can be applied for solving (4.5) just by replacing $A, \mathcal{C}$ and B by the corresponding discrete versions $A_h, \mathcal{C}_h$, and B_h and

$$\mathcal{C}_h^{-1}(B_h \mathbf{u}_{j,h} - g) \quad \text{by} \quad R_h \mathcal{C}^{-1}(B \mathbf{u}_{j,h} - g).$$

We will refer to the two algorithms on $(\mathbf{V}_h, M_h)$ as the discrete versions of **UA** and **UGA**. We further notice that, from the choice of our discrete space M_h, we get the convergence of the discrete algorithms to the discrete solution, and that both algorithms compute the residual for the constraint equation at the continuous level and then project or represent it using the larger space $\tilde{M}_h$. If the solution of (4.5) is approximated using the **UGA** on $(\mathbf{V}_h, M_h)$, then the special $(\cdot, \cdot)_h$ has to be used to compute α_j in **(UGα)**.

Remark 4.1. *If we consider a pair $(\mathbf{V}_h, \tilde{M}_h)$ for which a discrete inf-sup condition holds with a constant $\tilde{m}_h > 0$, and consider the associated pair $(\mathbf{V}_h, M_h)$ with the space M_h defined above, then from $M_h \subset \tilde{M}_h$, we get*

$$
\tilde{m}_h := \inf_{p_h \in \tilde{M}_h} \sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, p_h)}{\|p_h\|_h \; |\mathbf{v}_h|} \leq \inf_{p_h \in M_h} \sup_{\mathbf{v}_h \in \mathbf{V}_h} \frac{b(\mathbf{v}_h, p_h)}{\|p_h\|_h \; |\mathbf{v}_h|} := m_h. \tag{4.6}
$$

Thus, the pair $(\mathbf{V}_h, M_h)$ has a better inf-sup condition constant. Since the problem (4.5) has unique solution $(\mathbf{u}_h, p_h)$ in $\mathbf{V}_h \times M_h$, and $M_h \subset \tilde{M}_h$, we also have that $(\mathbf{u}_h, p_h)$ is the unique solution of (4.5) with M_h replaced by $\tilde{M}_h$. Writing a (global) linear system for (4.5) might not be possible, because bases for M_h are difficult to find, and hence the linear system associated with (4.5) can not always be assembled. Nevertheless, by using either **UA** *or* **UGA** *for solving (4.5), the iterates p_j approximating $p_h \in M_h$ are computed using bases for $\tilde{M}_h$.*

Next, we provide an example of representation operator as a sum of local projections. We consider a particular case of representation operator that appears in a natural way as sum of local projections when we try to smoothly approximate the residual associated with the constraint equation. To be more precise, let us assume that $\{\phi_1, \phi_2, \ldots, \phi_m\}$ is an **orthogonal** basis with respect to the $(\cdot, \cdot)_h$ inner product on $\tilde{M}_h$, consisting of locally supported functions. Having in mind that a typical example for $\tilde{M}_h$ is the space of continuous piecewise linear functions with respect to a mesh $\mathcal{T}_h$ and that the set of all local nodal functions $\{\phi_i\}_{i=1}^m$, form a basis for $\tilde{M}_h$, we can define $(\cdot,\cdot)_h$ by

$$(\phi_i,\phi_j)_h := \delta_i^j \ (1,\phi_i), \quad i,j = 1,2,\ldots,m, \tag{4.7}$$

and extend to $\tilde{M}_h \times \tilde{M}_h$ by

$$\left(\sum_{i=1}^m \alpha_i\phi_i, \sum_{j=1}^m \beta_j\phi_j\right)_h := \sum_{i=1}^m \alpha_i\beta_i \ (1,\phi_i).$$

Nothing, but the observation that

$$\left(\sum_{i=1}^m \frac{(p,\phi_i)}{(1,\phi_i)}\phi_i, \phi_j\right)_h = (p,\phi_j), \ j = 1,2,\ldots,m,$$

tells us that the representation operator $\mathcal{R}_h : Q \to \tilde{M}_h$ defined by (4.1) is

$$\mathcal{R}_h p = \sum_{i=1}^m \frac{(p,\phi_i)}{(1,\phi_i)}\phi_i, \qquad \text{for all } p \in Q. \tag{4.8}$$

Thus, in the case when $\{\phi_1,\phi_2,\ldots,\phi_m\}$ is a basis of local function, for the inner product defined by (4.7), we have that the representation operator $\mathcal{R}_h$ is a sum of local projectors. Using the identity (4.2), the computation of $\mathcal{C}_h^{-1}B_h\mathbf{v}_h$ becomes

$$\mathcal{C}_h^{-1}B_h\mathbf{v}_h = \sum_{i=1}^m \frac{(\mathcal{C}^{-1}B\mathbf{v}_h,\phi_i)}{(1,\phi_i)}\phi_i = \sum_{i=1}^m \frac{b(\mathbf{v}_h,\phi_i)}{(1,\phi_i)}\phi_i.$$

In implementations, this formula is more efficient because it does not require a matrix inversion. For the case when $\tilde{M}_h$ is the space of continuous piecewise linear functions with respect to a mesh $\mathcal{T}_h$, the formula is related to "lumping".

5. A Double Inexact Uzawa Algorithm and Multilevel Algorithms

In this section, following the ideas in [23, 30], we *review* an abstract (double) inexact Uzawa algorithm, that was introduced in [11], and then use it to describe the multilevel *Uzawa* and the *Uzawa-gradient* algorithms. We use the work of [11] to present a *unified and simplified* description and analysis for the two algorithms.

In what follows in this section, we assume that $\mathbf{V}$ and Q are infinite dimensional spaces. The description of the double inexact Uzawa algorithm is based on modifying the standard Uzawa algorithm as follows. First, the exact solve of the elliptic problem or the action of A^{-1} in the standard Uzawa algorithm, is replaced by an approximation process acting on the residual $\mathbf{r}_{j-1} := \mathbf{f} - A\mathbf{u}_{j-1} - B^*p_{j-1}$. The approximate process is described as a map Ψ defined on a subset of $\mathbf{V}^*$ which for $\phi \in \mathbf{V}^*$ returns an approximation of $\xi = A^{-1}\phi$. A common choice for $\Psi(\phi)$ is the discrete Galerkin projection of ξ on an appropriate subspace, see [3, 4, 11]. Second, the exact action of $\mathcal{C}^{-1}$ in the Uzawa algorithm, is also replaced by a linear process acting on the residual $s_j := B\mathbf{u}_j - g$. The approximate process is described as a map Φ defined on a subset of Q^*, which for a $q^* \in Q^*$, returns an approximation of $\eta := \mathcal{C}^{-1}q^*$. As an example of a linear process $\Phi(q^*)$ one can take the projection of $\mathcal{C}^{-1}q^*$ on a finite dimensional subspace of Q. The role of Φ is to provide a good approximation of the natural representation operator $\mathcal{C}^{-1}$. By using a smooth approximation $\Phi(q^*)$ of $\mathcal{C}^{-1}q^*$ we get that the successive approximations p_j of p are smooth functions and in the presence of elliptic regularity (see [5, 7, 8, 9, 10, 12, 29, 33, 34, 35]), the representation $A^{-1}\mathbf{r}_j$ can be better approximated by $\Psi(\mathbf{r}_j)$ on discrete spaces.

Let $\alpha \in \left(0, \frac{2}{M^2}\right)$ be a given relaxation parameter. The double Inexact Uzawa (**IU**) algorithm for approximating the solution $(\mathbf{u}, p)$ of (3.1) is:

Algorithm 5.1 (**Inexact Uzawa (IU)**). *Let* $(\mathbf{u}_0, p_0)$ *be any approximation for* $(\mathbf{u}, p)$*. For* $j = 1, 2, \cdots$*, construct* $(\mathbf{u}_j, p_j)$ *by*

$$\begin{aligned} \mathbf{u}_j &= \mathbf{u}_{j-1} + \Psi(\mathbf{f} - A\mathbf{u}_{j-1} - B^* p_{j-1}),\\ p_j &= p_{j-1} + \alpha\Phi(B\mathbf{u}_j - g). \end{aligned}$$

The next result describes a convergence result for the **(IU)** algorithm.

Theorem 5.2. *There are positive numbers* ϵ^1 *and* δ^1*, depending on* m, M *and* α *only, such that if* Ψ *and* Φ *satisfy*

$$\left|\Psi(\mathbf{r}_j) - A^{-1}\mathbf{r}_j\right|_{\mathbf{V}} \le \delta \left|A^{-1}\mathbf{r}_j\right|_{\mathbf{V}}, \quad j = 0, 1, \dots, \tag{5.1}$$

$$\left\|\Phi(s_j) - \mathcal{C}^{-1}s_j\right\| \le \epsilon \left\|\mathcal{C}^{-1}s_j\right\|, \quad j = 1, 2, \dots, \tag{5.2}$$

for a positive $\epsilon < \epsilon^1$ *and a positive* $\delta < \delta^1$*, then the sequence* $(\mathbf{u}_j, p_j)$ *produced by* **(IU)** *converges to* $(\mathbf{u}, p)$ *the solution of* (3.1)*. The rate of convergence depends only on* ϵ, δ, m, M*, and the relaxation parameter* α*.*

The proof of Theorem 5.2, together with concrete estimates for ϵ^1 and δ^1 follow from Theorem 1 of [11]. Next, we interpret Algorithm 5.1 as a multilevel algorithm, by defining the approximate inverse $\Psi(\mathbf{r}_j)$ as a Galerkin projection of $A^{-1}(\mathbf{r}_j)$ on an appropriate finite element subspace $\mathbf{V}_k$ of $\mathbf{V}$ that changes when j increases, and by defining a computable approximation $\Phi(s_j)$ of $\mathcal{C}^{-1}s_j$. We consider that (3.1) is the variational formulation of a boundary value problem on a fixed domain Ω, and assume that two sequences of nested finite element spaces

$$\mathbf{V}_1 \subset \mathbf{V}_2 \subset \cdots \subset \mathbf{V}, \text{ and } \tilde{M}_1 \subset \tilde{M}_2 \subset \cdots \subset Q,$$

are given. We note that a discrete stability condition for the family $\{(\mathbf{V}_k, \tilde{M}_k)\}$ is not required, but we assume that the sequence $\{\mathbf{V}_k\}$ is **dense** in $\mathbf{V}$. The subspaces $\mathbf{V}_k$ and $\tilde{M}_k$ could be standard multilevel spaces of functions on Ω associated with uniform or non-uniform meshes $\{\mathcal{T}_k\}$ on Ω. We consider that each $\tilde{M}_k$ is a finite dimensional Hilbert space equipped with an inner product $(\cdot,\cdot)_k$. We define first representation operators $\mathcal{R}_k : Q \to \tilde{M}_k$ by

$$(\mathcal{R}_k p, \tilde{q})_k = (p, \tilde{q}), \qquad \text{for all} \;\; q \in \tilde{M}_k,$$

i.e., $\mathcal{R}_k p$ is the Riesz representation of $q \to (p, \tilde{q})$ as a functional on $(\tilde{M}_k, (\cdot,\cdot)_k)$. We notice that in the particular case when $(\cdot,\cdot)_k$ agrees with the inner product on Q, we have that $\mathcal{R}_k$ agrees with the orthogonal projection onto $\tilde{M}_k$. The induced norm on $(\tilde{M}_k, (\cdot,\cdot)_k)$ is denoted by $\|\cdot\|_k$. We define our second approximate process by

$$\Phi(s_j) = \Phi(B\mathbf{u}_j - g) := \mathcal{R}_k(\mathcal{C}^{-1} s_j), \text{ if } \mathbf{u}_j \text{ is computed on } \mathbf{V}_k.$$

We will refer to the index k of $\mathbf{V}_k$ as the *iteration level.* The main idea of the multilevel algorithm is that as j increases, the approximations $\mathbf{u}_j$ of $\mathbf{u}$ are determined by solving *elliptic, symmetric and positive definite* problems on larger and larger subspaces $\mathbf{V}_k$ of $\mathbf{V}$. The multilevel (double) inexact Uzawa algorithm for solving (3.1) is:

Algorithm 5.3. Multilevel Inexact Uzawa (MIU).
Let $\alpha > 0$, and $r_0 > 1, R_0 > 0$ be fixed numbers.

Set $\mathbf{u}_0 = 0 \in \mathbf{V}$, $p_0 = 0 \in Q$, $j = 1$, $k = 1$.

Step 1. *Solve for $\mathbf{w}_j \in \mathbf{V}_k$ and find $\mathbf{u}_j \in \mathbf{V}_k$, $p_j \in M_k$:*

$$a(\mathbf{w}_j, \mathbf{v}) = < f, \mathbf{v} > - a(\mathbf{u}_{j-1}, v) - b(\mathbf{v}, p_{j-1}), \;\; \mathbf{v} \in \mathbf{V}_k,$$

$$\mathbf{u}_j = \mathbf{u}_{j-1} + \mathbf{w}_j, \quad p_j = p_{j-1} + \alpha \mathcal{R}_k \, \mathcal{C}^{-1}(B\mathbf{u}_j - g),$$

Step 2. *Solve for $\mathbf{w}_{j+1} \in \mathbf{V}_k$:*

$$a(\mathbf{w}_{j+1}, \mathbf{v}) = < f, \mathbf{v} > - a(\mathbf{u}_j, v) - b(\mathbf{v}, p_j), \;\; \mathbf{v} \in \mathbf{V}_k,$$

If $(M\|p_j - p_{j-1}\| < R_0|\mathbf{w}_{j+1}|$ **and** $|\mathbf{w}_j| < r_0|\mathbf{w}_{j+1}|)$,

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{w}_{j+1}, \quad p_{j+1} = p_j + \alpha \mathcal{R}_k \, \mathcal{C}^{-1}(B\mathbf{u}_{j+1} - g),$$

$j = j + 1$, **Go To Step 2**

Else: $k = k + 1$, $j = j + 1$, **Go To Step 1**

End

We note here that by taking $\Psi(\mathbf{r}_j) = \mathbf{w}_{j+1} \in \mathbf{V}_k \subset \mathbf{V}$, where $\mathbf{w}_{j+1}$ is defined in Step 2 of **MIU**, and $\Phi(s_j) = \mathcal{R}_k(\mathcal{C}^{-1} s_j) \in \tilde{M}_k \subset Q$ we can view **MIU** algorithm as a particular case of **(IU)** algorithm. To study the convergence of **MIU**, we introduce $F(\delta) := \frac{1}{\sqrt{1-\delta^2}}$, $\delta \in (0,1)$, and let δ_0, δ_{01}, and δ_{00} be such that $0 < \delta_0 < \delta_{01} < \delta_{00} < \delta^1$, where δ^1 is the threshold value of Theorem 5.2. Then, we assume that r_0 and R_0 are chosen such that

$$\delta_{01} F(\delta_{01}) \, r_0 + M \; R_0 \leq F(\delta_{00}). \tag{5.3}$$

The following convergence result was proved in [11].

Theorem 5.4. *Let $\alpha \in (0, 2/M^2)$ and ϵ^1, δ^1 be chosen such that the hypothesis of Theorem 5.2 hold. Assume that $\Phi(s_j) = \mathcal{R}_k \mathcal{C}^{-1}(s_j)$ satisfies (5.2) with some $\epsilon < \epsilon^1$. Let $\delta_0, \delta_{01}, \delta_{00}$ be such that $0 < \delta_0 < \delta_{01} < \delta_{00} < \delta^1$*

and assume r_0, R_0 *are chosen such that* (5.3) *is satisfied. We assume in addition that the condition* (5.1) *is satisfied for any first iteration* j *on a new level* $k+1$, *and that the space* $\mathbf{V}_1$ *is chosen such that the Galerkin approximation* $\mathbf{w}_1 \in \mathbf{V}_1$ *of* $\mathbf{u^f} = A^{-1}\mathbf{f}$ *satisfies* $|\mathbf{u^f} - \mathbf{w}_1| \leq \delta_0\ |\mathbf{u^f}|$. *Then, the sequence* $(\mathbf{u}_j, p_j)$ *produced by* **MIU** *converges to* $(\mathbf{u}, p)$ *the solution of* (3.1).

Remark 5.5. *The hypothesis of Theorem 5.4 might be seen as too restrictive. The assumption about the first iteration* j *on a new level* $k+1$ *can be satisfied if more approximation properties for the sequence* $\{\mathbf{V}_k\}$ *are given, see* [11]. *The assumption that* r_0, R_0 *satisfy* (5.3) *can lead to small* r_0 *and* R_0 *that forces the algorithm to jump to high levels* k *too fast. The purpose of Theorem 5.4 is to justify that convergence can be achieved, and that the driving parameters* α, r_0 *and* R_0 *are independent of level or the data* $\mathbf{f}$ *and* g. *In a practical implementation of* **MIU** *algorithm, good choices for the driving parameters can be found if the exact solution for a particular problem is known. The numerical experiments we performed so far showed that the restrictions on* r_0, R_0 *can be relaxed and they are level and data-independent.*

Next, we observe that as long as the **MIU** iteration is performed on the same level k, the iterations $(\mathbf{u}_j, p_j)$ corespond to standard Uzawa iterations on $(\mathbf{V}_k, M_k := \mathcal{R}_k\mathcal{C}^{-1}B\mathbf{V}_k)$ (see also the paragraph before Remark 4.1). Using the equivalence of Algorithm 3.1 and Algorithm 3.2 and the similarity between Algorithm 3.2 and Algorithm 3.3, we are led to a multilevel inexact Uzawa gradient algorithm:

Algorithm 5.6. Multilevel Inexact Uzawa Gradient (MIUG).
Let $r_0 > 1$ *and* $0 < R_0 < \frac{\sqrt{2}}{M}$ *be fixed numbers.*

Set $\mathbf{u}_0 = 0 \in \mathbf{V}$, $p_0 = 0 \in Q$, $j = 1$, $k = 1$.

Step 1. *Solve for* $\mathbf{w}_j \in \mathbf{V}_k$ *and find* $\mathbf{u}_j \in \mathbf{V}_k$, *and* $q_j \in M_k$ *by:*

$$a(\mathbf{w}_j, \mathbf{v}) = < f, \mathbf{v} > - a(\mathbf{u}_{j-1}, v) - b(\mathbf{v}, p_{j-1}),\ \ \mathbf{v} \in \mathbf{V}_k,$$

$$\mathbf{u}_j = \mathbf{u}_{j-1} + \mathbf{w}_j, \quad q_j = \mathcal{R}_k\ \mathcal{C}^{-1}(B\mathbf{u}_j - g),$$

Step 2. *Solve for* $\mathbf{h}_j, \mathbf{w}_{j+1} \in \mathbf{V}_k$ *and* α_j*:*

$$(\mathbf{UG1}) \qquad a(\mathbf{h}_j, \mathbf{v}) = -b(\mathbf{v}, q_j),\ \ \mathbf{v} \in \mathbf{V}$$

$$(\mathbf{UG}\alpha) \qquad \alpha_j = -\frac{(q_j, q_j)_k}{b(\mathbf{h}_j, q_j)}$$

$$(\mathbf{UGw}) \qquad \mathbf{w}_{j+1} = \alpha_j\ \mathbf{h}_j$$

Step 3. If $(M\|p_j - p_{j-1}\| < R_0|\mathbf{w}_{j+1}|$ **and** $|\mathbf{w}_j| < r_0|\mathbf{w}_{j+1}|)$*:*

$$(\mathbf{UG2}) \qquad p_j = p_{j-1} + \alpha_j\ q_j$$

$$(\mathbf{UG3}) \qquad \mathbf{u}_{j+1} = \mathbf{u}_j + \alpha_j\ \mathbf{h}_j$$

$$(\mathbf{UG4}) \qquad q_{j+1} = \mathcal{R}_k\ \mathcal{C}^{-1}(B\mathbf{u}_{j+1} - g)$$

$j = j+1$, **Go To Step 2**

Else $k = k+1$, $j = j+1$, **Go To Step 1**

End

Theorem 5.7. *Let* $\alpha^1 \in (0, 2/M^2)$ *and assume that all the conditions of Theorem 5.4, are satisfied for* $\alpha = \alpha^1$ *and* $R_0 = \sqrt{\alpha^1}$*. Then, the sequence* $(\mathbf{u}_j, p_j)$ *produced by* **MIUG** *converges to* $(\mathbf{u}, p)$ *the solution of* (3.1).

Proof. Since for each fixed iteration (and each fixed level) the Uzawa gradient algorithm is a particular case of the Uzawa algorithm, to justify the convergence of **MIUG**, we will use Theorem 5.4. The only extra condition we have to take care of is to verify that for each iteration j of **MIUG**, the α_j computed at **Step 2**, satisfies $\alpha_j < \alpha^1 < 2/M^2$. From (3.27), we have that $\sqrt{\alpha_j} = R_j = \frac{\|p_j - p_{j-1}\|}{|\mathbf{w}_{j+1}|}$, and the restriction for α_j follows from the condition $M\|p_j - p_{j-1}\| < R_0|\mathbf{w}_{j+1}|$ that is imposed at **Step 2**, and from the extra restriction $R_0 < \frac{\sqrt{2}}{M}$ that we assume for R_0. □

A useful note similar to Remark 5.5 holds for Theorem 5.7 as well.

6. Numerical results for the Stokes system

In this section we consider application of the multilevel algorithms developed in the previous sections to finite element discretization for the standard Stokes system:

$$
\begin{array}{lcll} -\Delta \mathbf{u} & + & \nabla p & = \mathbf{f}, \text{ in } \Omega \\ \operatorname{div} \mathbf{u} & & & = g, \text{ in } \Omega, \end{array} \tag{6.1}
$$

with vanishing Dirichlet boundary condition $\mathbf{u} = 0$ on $\partial\Omega$ and

$$
\int_\Omega g \, dx = 0,
$$

where Ω is the unit square and Δ is the componentwise Laplace operator. Define $\mathbf{V} := (H_0^1(\Omega))^2$, and $Q = L_0^2(\Omega) := \{h \in L^2(\Omega) | \int_\Omega h \, dx = 0\}$ and assume that $\mathbf{f} \in (L^2(\Omega))^2$ and $g \in L_0^2(\Omega)$. The variational formulation of the problem is:

Find $\mathbf{u} \in \mathbf{V}, p \in Q$ such that

$$
\begin{array}{lll} \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} & - \int_\Omega p \ \operatorname{div} \mathbf{v} & = \int_\Omega \mathbf{f} \cdot \mathbf{v}, \ \mathbf{v} \in \mathbf{V}. \\ \int_\Omega q \ \operatorname{div} \mathbf{u} & & = \int_\Omega g \ q, \quad q \in Q. \end{array}
$$

We introduce $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ as the bilinear forms defined by

$$
a(\mathbf{u}, \mathbf{v}) = a_0(\mathbf{u}, \mathbf{v}) := \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} = \int_\Omega \sum_{i=1}^{2} \nabla \mathbf{u}_i \cdot \nabla \mathbf{v}_i,
$$

$$
b(\mathbf{v}, q) := -\int_\Omega q \ \operatorname{div} \mathbf{v}, \ \mathbf{v} \in \mathbf{V}, q \in Q.
$$

Let $\langle \mathbf{f}, \mathbf{v} \rangle := \int_\Omega \mathbf{f} \cdot \mathbf{v}$, and $\langle g, q \rangle := \int_\Omega g \ q$. The corresponding spaces and operators for the Stokes system are

$\mathbf{V} := (H_0^1(\Omega))^2$, and $Q = Q^* = L_0^2(\Omega)$, $A : V \to V^*$ is $(-\Delta)^2 : (H_0^1(\Omega))^2 \to (H^{-1}(\Omega))^2$, $\mathcal{C} = \mathcal{C}^{-1} = I$ on $L_0^2(\Omega)$.

$B : V \to Q^*$ and $B^* : Q \to V^*$ are defined by

$$\langle B\mathbf{v}, q\rangle = -\int_\Omega q \;\operatorname{div} \mathbf{v} = \langle B^* q, \mathbf{v}\rangle \quad \text{for all } \mathbf{v} \in \mathbf{V},\; q \in Q.$$

Thus, $B = -div$ and $B^* = grad$. We define an original mesh $\mathcal{T}_1$ on Ω by splitting the unit square into four squares of size 1/2 and then by splitting each of the small squares in two triangles using the diagonal passing through the point $(1/2, 1/2)$. The family of uniform meshes $\{\mathcal{T}_k\}_{k\geq 0}$ is defined by a uniform refinement strategy, i.e., $\mathcal{T}_k$ is obtained from $\mathcal{T}_{k-1}$ by splitting each triangle of $\mathcal{T}_{k-1}$ in four similar triangles. We used three finite element spaces on a triangular mesh which we denote as follows:

$(P1)^2c$ - Continuous piecewise linear vector functions.
$P0d$ - Discontinuous piecewise constant scalar functions.
$P1c$ - Continuous piecewise linear scalar functions.

We apply both **MIU** and **MIUG** for approximating the solution of the Stokes system with the exact solution $u_1 = 1/5\pi^2 \sin(\pi x)\sin(2\pi y)$, $u_2 = 1/5\pi^2 \sin(2\pi x)\sin(\pi y)$, and $p = 2/3 - x^2 - y^2$. For each of the two algorithms we consider $\mathbf{V}_k = (P1)^2c$ as the space of vector functions which vanish on $\partial\Omega$ and are continuous piecewise linear functions with respect to the mesh $\mathcal{T}_k$ and four different choices for the spaces $\tilde{M}_k$, hence the representation operator $\mathcal{R}_k$. Here are the choices:

M1: $\tilde{M}_k = P0d$ is the space of discontinuous piecewise constant functions on $\mathcal{T}_k$, with the standard L^2 inner product. We will also refer to the discretization on $(\mathbf{V}_k, \tilde{M}_k)$ as the $\mathbf{P1c} - \mathbf{P0d}$ discretization.

M2: $\tilde{M}_k = P0d(h_{k-1})$ is the space of discontinuous piecewise constant functions on $\mathcal{T}_{k-1}$, with the standard L^2 inner product. This will be the $\mathbf{P1c(h)} - \mathbf{P0d(2h)}$ **stable** discretization.

M3: $\tilde{M}_k = P1c$ is the space of continuous piecewise linear functions on $\mathcal{T}_k$, with the standard L^2 inner product. We will also refer to the discretization on $(\mathbf{V}_k, \tilde{M}_k)$ as the $\mathbf{P1c} - \mathbf{Q(P0d)}$ discretization.

M4: $\tilde{M}_k = P1c$ is the space of continuous piecewise linear functions on $\mathcal{T}_k$, with the special inner product given by (4.7), and $\mathcal{R}_k = \mathcal{R}_h$ the projection on $\tilde{M}_k = \tilde{M}_h$ defined by (4.8) . This will be the $\mathbf{P1c} - \tilde{\mathbf{Q}}(\mathbf{P0d})$ discretization.

Each particular choice $\mathbf{Mj}$ (with $\mathbf{j = 1, 2, 3, 4}$) for $\tilde{M}_k$ defines a different **MIU** or **MIUG** algorithm named **MjIU** or **MjIUG** algorithm, respectively. We start all the algorithms on the fourth level of refinement and record the error for the velocity and the pressure for the last iteration $j = j(k)$ before leaving the level k. We also record on the last column of the table the number of iterations performed by the algorithm on each level k. For all the **MIU** algorithms we choose the driving parameters: $\alpha = 0.6$, and

$r_0 = 3$. For **M1IU** and **M2IU** we took $R_0 = 2$, and for **M3IU** and **M4IU** we took $R_0 = 3$. Below are the numerical results for the four cases.

Remark 6.1. *We see that* **M2IU***, the case of stable pairs, performs better than* **M1IU***. Nevertheless, we notice the inter-level error reduction for both* **M3IU** *and* **M4IU** *is better (especially for the pressure) when compared to* **M2IU** *in spite of the absence of the classical discrete stability. We also applied the* **M2IU***, (the* $\mathbf{P1c(h) - P0d(2h)}$ *discretization) directly on level* $k = 8$*, using as the stopping criteria a maximum number of iterations,* $j = 40$*. We obtained* $|\mathbf{u} - \mathbf{u}_j| = 0.0010265$*, and* $\|p - p_j\| = 0.0051319$*. The running time is almost three times longer and the pressure error is worse by a factor of* 1.5 *when compared to the corresponding multilevel algorithm with levels* $k = 4, \ldots, 8$*. We also notice that, for the one level algorithm, after 35 iterations, the pressure error started to increase.*

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\\|p - p_j\\|$	ratio	# iter
k=4	0.0279534		0.0721607		12
k=5	0.0141637	1.97	0.0407461	1.77	6
k=6	0.0071186	1.99	0.0229744	1.77	6
k=7	0.0032312	2.20	0.0128632	1.79	7
k=8	0.0014961	2.16	0.0072447	1.78	7

TABLE 1. Solving (6.1) using **M1IU** ($P1c$-$P0d$). We see convergence despite the lack of an LBB condition.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\\|p - p_j\\|$	ratio	# iter
k=4	0.0210181		0.0646828		15
k=5	0.0097475	2.16	0.0293469	2.20	8
k=6	0.0047002	2.07	0.0142054	2.07	7
k=7	0.0022489	2.08	0.0069603	2.04	7
k=8	0.0010824	2.07	0.0034298	2.03	7

TABLE 2. Using **M2IU** ($\mathbf{P1c(h) - P0d(2h)}$ **stable** discretization). See Remark 6.1 for a discussion of the results.

We compare next **MjIU**, with **MjIUG** for $\mathbf{j = 1, 2, 3, 4}$. For the general **MIUG** algorithm the parameter α is variable and is computed by the algorithm. Since we solve the Stokes equations, we have that $M = 1$, and in light of (3.27), we choose $R_0 = \sqrt{1.99}$. On each level we imposed $|\mathbf{w}_j| < r_0|\mathbf{w}_{j+1}|$

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0152142		0.0149342		34
k=5	0.0074348	2.04	0.0038373	3.89	17
k=6	0.0036844	2.02	0.0011378	3.37	16
k=7	0.0018365	2.01	0.0003555	3.20	14
k=8	0.0009170	2.00	0.0001161	3.06	14

Table 3. Using **M3IU** ($P1c$-$Q(P0d)$ discretization). We see convergence despite the lack of an LBB condition.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0149215		0.0090548		35
k=5	0.0073723	2.02	0.0021528	4.21	17
k=6	0.0036705	2.01	0.0005301	4.06	17
k=7	0.0018334	2.00	0.0001478	3.59	14
k=8	0.0009163	2.00	0.0000440	3.36	13

Table 4. A summary of results for solving (6.1) using **M4IU** ($P1c$-$\tilde{Q}(P0d)$ discretization). We see convergence despite the lack of an LBB condition.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0157361		0.0969029		10
k=5	0.0083125	1.89	0.0522796	1.85	2
k=6	0.0044924	1.85	0.0278581	1.88	2
k=7	0.0025412	1.77	0.0148631	1.87	2
k=8	0.0015309	1.66	0.0079715	1.87	2

Table 5. Solving (6.1) using **M1IUG** ($P1c$-$P0d$). We see convergence despite the lack of an LBB condition.

for $r_0 = 4$, starting with the second iteration. The numerical results are recorded in Tables 1-8.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0431989		0.0513424		18
k=5	0.0224632	1.92	0.0261196	1.97	2
k=6	0.0114112	1.97	0.0129469	2.02	2
k=7	0.0057377	1.99	0.0064014	2.02	2
k=8	0.0028747	2.00	0.0031777	2.01	2

Table 6. Using **M2IUG** ($\mathbf{P1c(h) - P0d(2h)}$ **stable** discretization). See Remark 6.2 for a discussion of the results.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0151621		0.0153813		16
k=5	0.0075497	2.01	0.0042957	3.58	2
k=6	0.0036952	2.04	0.0012356	3.48	4
k=7	0.0018471	2.00	0.0005105	2.42	2
k=8	0.0009223	2.00	0.0002252	2.27	2

TABLE 7. Using **M3IUG** ($P1c$-$Q(P0d)$ discretization). We see convergence despite the lack of an LBB condition.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0148728		0.0085859		12
k=5	0076290	1.95	0.0045423	1.89	1
k=6	0.0036695	2.08	0.0005780	7.86	11
k=7	0.0018330	2.00	0.0001323	4.37	5
k=8	0.0009163	2.00	0.0000396	3.34	5

TABLE 8. A summary of results for solving (6.1) using **M4IUG** ($P1c$-$\tilde{Q}(P0d)$ discretization). We see convergence despite the lack of an LBB condition.

Remark 6.2. *We note that* **MIUG** *algorithms perform better than the corresponding* **MIU** *algorithms from the iteration number point of view. We also notice that the inter-level pressure error reduction is better for the* **MIUG** *algorithms, and we have convergence in spite of the absence of the classical discrete stability.*

We performed numerical experiments with a version of **MIUG**, where the change of level condition of **Step 3** was replaced by (3.30). We will refer to this modified Multilevel Gradient algorithm simply as the **MG** algorithm. In Table 9 we show the numerical results for the **M4** case, i.e. the $P1c$-$\tilde{Q}(P0d)$ discretization on each level, and with the residual reduction factor $\rho_{\mathbf{0}} = \mathbf{0.81}$. The numerical results of the **MG** algorithm suggest that a fixed (and small) number of iterations on each level of the multilevel Uzawa gradient would still produce a good error reduction. We modified **M4IUG** by imposing exact five iterations on each level. The results are recorded in Table 10. In this case, our numerical results show that the inter-level pressure error is $\|p - p_h\| \approx O(h^2)$. It is important to notice that for our Stokes model problem, we have found that the order of convergence for pressure can improve from $O(h)$ for the standard stable discretization to $O(h^2)$ for the multilevel and residual smoothing discretization cases. A proof of this statement in the Stokes or the general case remains to be further investigated.

To emphasize on the importance of the *multilevel* approximation, we applied the gradient method for the **M4** case, i.e. the $P1c$-$\tilde{Q}(P0d)$ discretization directly on level $k = 8$, for $j = 40$ iterations. We have found $|\mathbf{u} - \mathbf{u}_j| = 0.0009168$, and $\|p - p_j\| = 0.0001946$. The running time is three times longer and the pressure error is worse by a factor of 5 when compared to the corresponding multilevel algorithm with levels $k = 4, \ldots, 8$.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0148728		0.0085859		12
k=5	0.0073613	2.02	0.0019321	4.44	6
k=6	0.0036693	2.01	0.0004809	4.02	5
k=7	0.0018331	2.00	0.0001306	3.68	5
k=8	0.0009163	2.00	0.0000388	3.37	5

TABLE 9. A summary of results for solving (6.1) using the **MG** algorithm with $P1c$-$\tilde{Q}(P0d)$ discretization. We see **fast convergence** despite the lack of an LBB condition.

$h = 1/2^k$	$\|\mathbf{u} - \mathbf{u}_j\|$	ratio	$\|p - p_j\|$	ratio	# iter
k=4	0.0195289		0.0283165		5
k=5	0.0075733	2.58	0.0046353	6.11	5
k=6	0.0036837	2.06	0.0009461	4.90	5
k=7	0.0018342	2.01	0.0002128	4.45	5
k=8	0.0009164	2.00	0.0000526	4.05	5

TABLE 10. A summary of results for solving (6.1) using a modified **M4IUG** algorithm with $P1c$-$\tilde{Q}(P0d)$ discretization and 5 iterations on each level.

7. Conclusion

We presented multilevel algorithms for discretizing symmetric saddle point problems for the particular case when the form $a(\cdot,\cdot)$ is symmetric and coercive. The new algorithms are based on the inexact Uzawa method at the continuous level and on the existence of multilevel sequences of nested approximation spaces. The convergence of the algorithms is driven by the accuracy of the residual representation for the elliptic problem at each iteration step. By slightly modifying the update for the second variable one can find better approximations for p when compared to discretization on standard stable pairs where a Ladyshenskaya-Babuška-Brezzi condition holds true. The new introduced multilevel Uzawa gradient method, automatically selects the relaxation parameter, lowers the number of iterations on each level, and significantly improves the error reduction for the second component of the solution. The algorithms can be applied to a large class of first order systems of PDEs that can be reformulated at the continuous level as symmetric SPPs. Typical example of such systems including the div-curl system and the Maxwell equations. In solving such problems by our approach, the fact that we obtain fast convergence with families that are not necessarily stable, leads to efficient and simple to implement iterative solvers that can compete with standard discretization and approximation approaches.

References

[1] K. Arrow, L. Hurwicz and H. Uzawa. *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[2] A. Aziz and I. Babuška, Survey lectures on mathematical foundations of the finite element method. In A. Aziz, editor, *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, Academic Press, 1-362, New York, NY, 1972.

[3] C. Bacuta, A Unified Approach for Uzawa Algorithms, *SIAM Journal of Numerical Analysis*, Volume 44, Issue 6, 2006, pp. 2245-2649.

[4] C. Bacuta, Schur Complements on Hilbert Spaces and Saddle Point Systems, *Journal of Computational and Applied Mathematics*, Volume 225, Issue 2, 2009, pp 581-593.

[5] C. Bacuta, Subspace interpolation with applications to elliptic regularity, *Numerical Functional Analysis and Optimization*, Volume 29, Issue 1 & 2, 2008, pp 88 - 114.

[6] C.Bacuta, J. H. Bramble and J. Pasciak. Using finite element tools in proving shift theorems for elliptic boundary value problems. *Numerical Linear Algebra with Application*, Volume 10, Issue no 1-2, 33-64, 2003.

[7] C.Bacuta, J. H. Bramble and J. Pasciak. Shift theorems for the biharmonic Dirichlet problem, *Recent Progress in Computational and Applied PDEs*, Kluwer Academic/Plenum Publishers, volume of proceedings for the CAPDE conference, Zhangjiajie, China, July 2001.

[8] C.Bacuta, J. H. Bramble and J. Xu, Regularity estimates for elliptic boundary value problems in Besov spaces, *Mathematics of Computation*, 72, 2003, pp 1577-1595.

[9] C.Bacuta, J. H. Bramble and J. Xu, Regularity estimates for elliptic boundary value problems with smooth data on polygonal domains, *J. Numer. Math.*, 11 (2003), no. 2, 7594.

[10] C. Bacuta, A. Mazzucato, V. Nistor, and L. Zikatanov. Interface and mixed boundary value problems on n-dimensional polyhedral domains. *Doc. Math.*, 15:687–745, 2010.

[11] C. Bacuta, and P. Monk, Multilevel discretization of Symmetric Saddle Point Systems without the discrete LBB Condition, *Applied Numerical Mathematics*, Volume 62, Issue 6, 2012, pp 667-681.

[12] C. Bacuta, V. Nistor and L. Zikatanov, Improving the rate of convergence of 'high order finite elements' on polyhedra I: apriori estimates: Mesh Refinements and Interpolation, with V. Nistor and L. Zikatanov, *Numerical Functional Analysis and Optimization*, Vol. 26, No 6, 2005, pp. 613-639.

[13] E. Bansch, P. Morin and R.H. Nochetto. An adaptive Uzawa FEM for the Stokes problem: Convergence without the inf-sup condition. *Siam J. Numer. Anal.*, 40:1207-1229, 2002.

[14] M. Benzi, G.H. Golub and J. Liesen, Numerical Solutions of Saddle Point Problems. *Acta Numerica*, 14 (2005), pp. 1-137.

[15] F. A. Bornemann and P. Deuhard. The cascadic multigrid method for elliptic problems. *Numer. Math.*, 75:135152, 1996.

[16] D. Braess, *Finite elements*, Springer-Verlag, New York. 1992.

[17] D. Braess, and W. Dahmen, A cascadic multigrid algorithm for the Stokes problem, *Numer. Math.*, (1999), 82 :179-191.

[18] D. Braess and W. Dahmen. A cascadic multigrid algorithm for the Stokes equations. *Numer. Math.*, 82(2):179–191, 1999.
[19] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.
[20] J.H Bramble and J.E. Pasciak, A new approximation technique for div-curl systems, *Math. Comp.*, 73(2004), 1739-1762.
[21] J.H Bramble, J.E. Pasciak, an T. Kolev, A least-squares method for the time-harmonic Maxwell equations, *J. Numer. Math.* 13,(2005) pp.237-320.
[22] J.H Bramble, J.E. Pasciak and P.S. Vassilevski, Computational scales of Sobolev norms with application to preconditioning, *Math. Comp.*, 69 (2000) 463-480.
[23] J. H. Bramble, J. Pasciak and A.T. Vassilev, Analysis of the inexact Uzawa algorithm for saddle point problems. *Siam J. Numer. Anal.*, 34:1072-1092, 1997.
[24] J. H. Bramble and X. Zhang, The analysis of multigrid methods, *Handbook for Numerical Analysis*, Vol. VII, 173-415, P. Ciarlet and J.L. Lions, eds., North Holland, Amsterdam, 2000.
[25] S. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, New York, 1994.
[26] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York. 1991.
[27] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers, *RAIRO Anal. Numer.*, Vol. 8, No. R-2, pp. 129-151, 1974.
[28] S. Dahlke, W. Dahmen and K. Urban. Adaptive Wavelet methods for saddle point problems-Optimal convergence rates. *Siam J. Numer. Anal.*, 40:1230-1262, 2002.
[29] M. Dauge. *Elliptic Boundary Value Problems on Corner Domains.* Lecture Notes in Mathematics 1341. Springer-Verlag, Berlin, 1988.
[30] H.C. Elman and G.H. Golub. Inexact and preconditioned Uzawa Algorithms for saddle point problems. *Siam J. Numer. Anal.*, 31:1645-1661, 1994.
[31] H.C. Elman, D. Silvester and A. Wathen. *Finite Elements and Fast Iterative Solvers*, Oxford Science Publications, Oxford, 2005.
[32] M. Fortin and R. Glowinski *Augmented Lagrangian Methods: Applications to the numerical solutions of boundary value problems.* Studies in Mathematics and Applications, Vol 15, North-Holland (1983).
[33] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations.* Springer-Verlag, Berlin, 1986.
[34] P. Grisvard. *Singularities in Boundary Value Problems.* Masson, Paris, 1992.
[35] R. B. Kellogg . Interpolation between subspaces of a Hilbert space ,Technical note BN-719. Institute for Fluid Dynamics and Applied Mathematics, University of Maryland, College Park, 1971.
[36] Y. Kondratyuk and R. Stevenson. An Optimal Adaptive Finite Element Method for the Stokes Problem. *Siam J. Numer. Anal.*, 46:746-775, 2008.
[37] P. Morin, R.H. Nochetto and G. Siebert. Data oscillation and convergence of adaptive FEM. *SIAM J. Numer. Anal.*38(2):466-488, 2000.
[38] R.H. Nochetto and J. Pyo. Optimal relaxation parameter for the Uzawa Method. *Numer. Math.*, 98:695-702, 2004.
[39] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, Springer, Berlin, 1994.
[40] F.J. Sayas, Infimum-supremum *Bol. Soc. Esp. Mat. Apl.* No. 41 (2007), 19-40.
[41] R. R. Verfürth.. *A Review of A Posterirori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley-Teubner, Chichester, 1996.
[42] J. Xu and L. Zikatanov. Some observations on Babuška and Brezzi theories. *Numer. Math.*, 94(1):195-202, 2003.

University of Delaware, Department of Mathematics, 501 Ewing Hall 19716
E-mail address: bacuta@math.udel.edu

University of Delaware, Department of Mathematics, 501 Ewing Hall 19716
E-mail address: shu@math.udel.edu