CASCADIC MULTILEVEL ALGORITHMS FOR
SYMMETRIC SADDLE POINT SYSTEMS

CONSTANTIN BACUTA

Abstract. In this paper, we introduce a multilevel algorithm for approximating variational formulations of symmetric saddle point systems. The algorithm is based on availability of families of stable finite element pairs and on the availability of fast and accurate solvers for symmetric positive definite systems. On each fixed level an efficient solver such as the gradient or the conjugate gradient algorithm for inverting a Schur complement is implemented. The level change criterion follows the cascade principle and requires that the iteration error be close to the expected discretization error. We prove new estimates that relate the iteration error and the residual for the constraint equation. The new estimates are the key ingredients in imposing an efficient level change criterion. The first iteration on each new level uses information about the best approximation of the discrete solution from the previous level. The theoretical results and experiments show that the algorithms achieve optimal or close to optimal approximation rates by performing a non-increasing number of iterations on each level. Even though numerical results supporting the efficiency of the algorithms are presented for the Stokes system, the algorithms can be applied to a large class of boundary value problems, including first order systems that can be reformulated at the continuous level as symmetric saddle point problems, such as the Maxwell equations.

1. Introduction

The cascade principle for elliptic partial differential equations (PDEs) was introduced by Deuflhard, Leinen and Yserentant in [26]. The main advantage of cascadic methods is that the iteration on each level is terminated as soon as the algebraic error is below the truncation or discretization error. Shaidurov [31] introduced a cascadic conjugate gradient and proved optimality in the energy norm for elliptic problems in two dimensions. The results were extended by Bornemann and Deuflhard to the three dimensional problem in [9].

In this paper we adopt the cascade principle in the context of multilevel discretization of symmetric and coercive saddle point (SP) systems. We let \( V \) and \( Q \) be Hilbert spaces, and assume that \( a(\cdot, \cdot) \) is a symmetric bounded and coercive bilinear form on \( V \times V \) that defines also the inner product on

2000 Mathematics Subject Classification. 74S05, 74B05, 65N22, 65N55.
Key words and phrases. Uzawa algorithms, Uzawa gradient, Uzawa conjugate gradient, symmetric saddle point system, multilevel methods, cascadic algorithm, cascade principle.
2 CONSTANTIN BACUTA

V, and that \( b(\cdot, \cdot) \) is a continuous bounded bilinear form on \( V \times Q \) satisfying a continuous (LBB) or inf-sup condition. We denote the inner product on \( Q \) by \( \langle \cdot, \cdot \rangle \), and assume that the data \( f \) and \( g \) belong to the dual spaces \( V^* \) and \( Q^* \), respectively. We consider the variational problem: Find \((u, p) \in V \times Q\) such that

\[
\begin{align*}
  a(u, v) + b(v, p) &= \langle f, v \rangle, & \text{for all } v \in V, \\
  b(u, q) &= \langle g, q \rangle, & \text{for all } q \in Q.
\end{align*}
\]

(1.1)

There is a broad literature on the multilevel finite element discretization for (1.1), see [1, 2, 17, 15, 27, 14, 16, 21, 8, 32, 34, 35]. More recent work in multilevel approximation of variational formulations of saddle point type systems can be found in [2, 7, 20, 25]. A cascadic approach for discretizing (1.1) was done by Braess, Dahmen and Sarazin for the Stokes systems in [12, 11].

In this paper, we present a general Cascadic Multilevel (CM) algorithm for solving the problem (1.1). We start by assuming that a sequence of pairs \( \{(V_k, M_k)\}_{k \geq 1} \) that satisfies a discrete inf–sup condition for every \( k \geq 1 \), and a sequence of prolongation operators \( P_{k,k+1} : M_k \to M_{k+1} \) are available.

**Algorithm 1.1. CM Algorithm**

- **Set** \( j = 1, k = 1, \quad u_0 = 0 \in V_1, \) and \( p_0 \in M_1 \).
- **Step CM1:** Solve for \( u_j \in V_k \) and \( q_j \in M_k \):
  \[
  \begin{align*}
  a(u_j, v) &= \langle f, v \rangle - b(v, p_{j-1}), & \text{for all } v \in V_k, \\
  (q_j, q) &= b(u_j, q) - \langle g, q \rangle, & \text{for all } q \in M_k.
  \end{align*}
  \]
- **Step CM2:** Compute \((u_{j+1}, p_j)\) from \((u_j, p_{j-1})\) by a process on \((V_k, M_k)\).
- **Check a level change condition (LC).**
- **Step CM3:** Repeat CM2 with \( j \to j + 1 \) until (LC) is satisfied.
  - Define \( p_0^{(k+1)} := P_{k,k+1}(p_j) \), increase the level \((k \to k + 1)\).
  - Increase \( j \to j + 1 \) and Go To CM1 with \( p_{j-1} = p_0^{(k+1)} \).

The algorithm is quite general, and we will consider the cases when the process of Step CM2 is executed by Uzawa (U), Uzawa Gradient (UG) or Uzawa Conjugate Gradient (UCG) “one step” iteration. This means that our proposed CM algorithm is a Schur complement type process. The main computational challenge for a typical one step iteration is to invert the discrete operator \( A_k \) associated with the form \( a(\cdot, \cdot) \) on \( M_k \). When a fixed level iteration ends due to the level change criterion (LC), only \( p_j \) needs
to be prolongated to the next space $\mathcal{M}_{k+1}$. Thus, the CM algorithm is a simple to implement iterative process. On the other hand, this algorithm is build on the premise that the action of $A_k^{-1}$ is fast and exact.

Some other multilevel approaches for solving (1.1) that are related with the proposed CM algorithm are as follows. In [33], Verfürth uses an inexact conjugate gradient algorithm on a single fine level where the inexact elliptic process is provided by a multigrid algorithm that requires a multilevel structure. Level wise, our CM algorithm moves always upwards and an exact elliptic solver is called at each iteration. In [12], Braess and Sarazin, develop a multigrid algorithm for discretizing the Stokes system that is based on a smoother acting on the residual of the global system. In [11], Braess and Dahmen provide sharp estimates of a cascadic approach for the Stokes problem that uses the smoothing procedure proposed in [12]. We emphasize that, according to the terminology of [12], the Braess-Sarazin-Dahmen approach is $u$-dominated, i.e., $(u_{j+1}, p_{j})$, mainly depends on $u_{j}$, while our proposed CM algorithm is $p$ dominated, i.e., $(u_{j+1}, p_{j})$ mainly depends on $p_{j-1}$.

In [6, 3], we investigated similar multilevel algorithms based on the inexact Uzawa algorithms at the continuous level and on inexact processes for approximating continuous residuals. When the inexact process acting on residuals of the first equation is a standard Galerkin projection, the algorithms proposed in [6, 3] become particular versions of the proposed CM algorithm. Nevertheless, the level change criterion we propose, and the choice of stable families of approximation spaces we use in this paper, lead to a different type of CM algorithm.

One novelty of the CM algorithm we propose in Section 4, is the level change condition that takes full advantage of the maximum expected order of the discretization error. We find an iteration error estimator that is easy to compute and works well with all three (U, UG and UCG) choices of iterative processes. For non-convex domains, where the full regularity of the solution might be lost, we consider special discrete spaces based on graded meshes, and using the appropriate level change condition we are able to recover optimal or close to optimal rates of approximation for the continuous solution.

The rest of the paper is organized as follows. In Section 2, we introduce the needed notation for building the theory and the analysis for the CM algorithm. In Section 3, we review the Uzawa, UG and the UCG algorithms and find a sharp error estimator for the iteration error. In Section 4, we specify the (LC) condition and concrete level solvers in order to define implementable CM algorithms. In Section 5, we present the performance of a few versions of CM algorithms for different choices of solvers and discrete spaces for approximating the solution of the Stokes system. We summarize our conclusions in Section 6.
2. General Framework and Notation

We consider the standard notation for the saddle point problem (SPP) abstract framework. We let $\mathbf{V}$ and $\mathbf{Q}$ be two Hilbert spaces with inner products $\langle \cdot, \cdot \rangle$ and $(\cdot, \cdot)$ respectively, with the corresponding induced norms $| \cdot |_{\mathbf{V}} = | \cdot | = a(\cdot, \cdot)^{1/2}$ and $\| \cdot \|_{\mathbf{Q}} = \| \cdot \| = (\cdot, \cdot)^{1/2}$. The dual pairings on $\mathbf{V}^* \times \mathbf{V}$ and $\mathbf{Q}^* \times \mathbf{Q}$ are denoted by $\langle \cdot, \cdot \rangle$. Here, $\mathbf{V}^*$ and $\mathbf{Q}^*$ denote the duals of $\mathbf{V}$ and $\mathbf{Q}$, respectively. We assume that $b(\cdot, \cdot)$ is a bilinear form on $\mathbf{V} \times \mathbf{Q}$, satisfying the following conditions.

\begin{align}
(2.1) \quad \inf_{p \in \mathbf{Q}} \sup_{v \in \mathbf{V}} \frac{b(v, p)}{\|p\| \|v\|} &= m > 0, \quad \text{and} \quad \sup_{p \in \mathbf{Q}} \sup_{v \in \mathbf{V}} \frac{b(v, p)}{\|p\| \|v\|} = M < \infty.
\end{align}

For $f \in \mathbf{V}^*$, $g \in \mathbf{Q}^*$ we consider the variational problem (1.1). It is known that the variational problem (1.1) has a unique solution $(u, p)$ for any $f \in \mathbf{V}^*$, $g \in \mathbf{Q}^*$, see [18, 19, 22, 29, 24, 2].

For the SP discretization, we let $\mathbf{V}_h \subset \mathbf{V}$, $\mathcal{M}_h \subset \mathbf{Q}$ and assume that

\begin{align}
(2.2) \quad \inf_{p_h \in \mathcal{M}_h} \sup_{v_h \in \mathbf{V}_h} \frac{b(v_h, p_h)}{\|p_h\| \|v_h\|} &= m_h > 0,
\end{align}

and define the constant $M_h$ as

\begin{align}
(2.3) \quad M_h := \sup_{p_h \in \mathcal{M}_h} \sup_{v_h \in \mathbf{V}_h} \frac{b(v_h, p_h)}{\|p_h\| \|v_h\|} \leq M.
\end{align}

Let the discrete operators $A_h : \mathbf{V}_h \to \mathbf{V}_h$ and $B_h : \mathbf{V}_h \to \mathcal{M}_h$ be defined by

\begin{align}
(A_h u_h, v_h) &= a(u_h, v_h) \\
(B_h u_h, q_h) &= (u_h, B^T h q_h) = b(u_h, q_h)
\end{align}

for all $u_h, v_h \in \mathbf{V}_h$, $u_h, q_h \in \mathcal{M}_h$.

where $\langle (\cdot, \cdot) \rangle$ is an inner product on $\mathbf{V}_h \times \mathbf{V}_h$, that is usually associated with a basis on $\mathbf{V}_h$. The discrete version of (1.1) is:

Find $(u_h, p_h) \in \mathbf{V}_h \times \mathcal{M}_h$ such that

\begin{align}
(2.4) \quad a(u_h, v_h) + b(v_h, p_h) &= \langle f_h, v_h \rangle \\
b(u_h, q_h) &= \langle g_h, q_h \rangle
\end{align}

for all $v_h \in \mathbf{V}_h$, $q_h \in \mathcal{M}_h$, where $f_h \in \mathbf{V}_h$ and $g_h \in \mathcal{M}_h$ are defined by

\begin{align}
(2.5) \quad \langle f_h, v_h \rangle &= \langle f_h, v_h \rangle, \quad v_h \in \mathbf{V}_h, \quad (g_h, q_h) = (g_h, q_h), \quad q_h \in \mathcal{M}_h.
\end{align}

The matrix or operatorial form of (2.4) is:

\begin{align}
(2.6) \quad A_h u_h + B_h^T p_h &= f_h, \\
\quad B_h u_h &= g_h.
\end{align}

It is well known from [10, 18, 30, 36] that, under the assumption (2.2), the problem (2.4) has a unique solution $(u_h, p_h)$ and

\begin{align}
\quad |u - u_h| + \|p - p_h\| \leq C(m_0, M) \left( \inf_{v_h \in \mathbf{V}_h} |u - v_h| + \inf_{q_h \in \mathcal{M}_h} \|p - q_h\| \right),
\end{align}

where $(u, p)$ is the solution of the continuous problem (1.1).
Let $S_h : M_h \to M_h$, be the discrete Schur complement defined by $S_h := B_h A_h^{-1} B_h^T$. It is easy to check that $S_h$ is a symmetric and positive definite operator on $M_h$. We have that $(\cdot, \cdot)_{S_h} := (S_h \cdot, \cdot)$ is another inner product on $M_h$ with the induced normed denoted by $\| \cdot \|_{S_h}$. It is well known that the lowest and the largest eigenvalues of $S_h$ are $m_h^2$ and $M_h^2$, respectively. Thus,

\begin{equation}
(2.7) \quad m_h \| q_h \| \leq \| q_h \|_{S_h} = (S_h q_h, q_h)^{1/2} \leq M_h \| q_h \| \quad \text{for all } q_h \in M_h.
\end{equation}

**Remark 2.1.** On $V_h$, we consider the same norm as the norm on $V$. The inner product $(\cdot, \cdot)$ on $V_h \times V_h$ is not the restriction of the inner product $a(\cdot, \cdot)$, and is used only for defining the discrete operators $A_h$ and $B_h^T$. In what follows, we will need in fact only to work with $A_h^{-1} B_h^T : M_h \to V_h$ and $S_h = B_h A_h^{-1} B_h^T$ which are independent of the choice of the inner product $(\cdot, \cdot)$. Indeed, if $q_h \in M_h$ is arbitrary, then $w_h = A_h^{-1} B_h^T q_h$ is the unique solution of the problem

$$a(w_h, v) = b(v, q_h), \quad \text{for all } v \in V_h,$$

and $S_h q_h = B_h A_h^{-1} B_h^T q_h = B_h w_h$ does not depend on the inner product $(\cdot, \cdot)$. We also note that if $r_h \in M_h$ is arbitrary and $v_h = A_h^{-1} B_h^T r_h$, then

\begin{equation}
(2.8) \quad a(w_h, v_h) = b(v_h, q_h) = (B_h A_h^{-1} B_h^T r_h, q_h) = (S_h q_h, r_h) = (q_h, r_h)_{S_h}.
\end{equation}

In particular, we have

\begin{equation}
(2.9) \quad |w_h|^2 = a(w_h, w_h) = \| q_h \|^2_{S_h}.
\end{equation}

Using the Schur complement $S_h$, the system (2.6) can be decoupled to

\begin{equation}
(2.10) \quad \begin{aligned}
S_h p_h &= B_h A_h^{-1} f_h - g_h \\
0 &= A_h^{-1} (f_h - B_h^T p_h).
\end{aligned}
\end{equation}

### 3. Uzawa, Uzawa Gradient and Uzawa Conjugate Gradient Algorithms

First, we present a unified variational form of the Uzawa, the Uzawa gradient, and the Uzawa conjugate gradient algorithms for solving the SPP (2.4). The standard U and UG algorithms can be rewritten such that they differ only by the way the relaxation parameter $\alpha$ is chosen. For the Uzawa algorithm, we have to choose $\alpha = \alpha_0$ a fixed number in the interval $\left(0, \frac{2}{M_h^2}\right)$. For the UG algorithm, the parameter $\alpha$ is chosen to impose the orthogonality of consecutive residuals associated with the second equation in (2.4). The first step for Uzawa is identical with the first step of UG. We combine the two algorithm in:

**Algorithm 3.1.** *(U-UG) Algorithms*

**Step 1:** Set $u_0 = 0 \in V_h$, $p_0 \in M_h$, compute $u_1 \in V_h$, $q_1 \in M_h$ by

$$a(u_1, v) = (f_h, v) - b(v, p_0), \quad \text{for all } v \in V_h$$

$$q_1, q = b(u_1, q) - (g_h, q), \quad \text{for all } q \in M_h.$$
Step 2: For \( j = 1, 2, \ldots \), compute \( h_j, \alpha_j, p_j, u_{j+1}, q_{j+1} \) by

\[(U - UG) \quad a(h_j, v) = -b(v, q_j), \quad v \in V_h \]

\[(U\alpha) \quad \alpha_j = \alpha_0 \text{ for the Uzawa algorithm or} \]

\[(UG\alpha) \quad \alpha_j = -\frac{(q_j, q_j)}{b(h_j, q_j)} = \frac{(q_j, q_j)}{(q_j, q_j)_{S_h}}, \text{ for the UG algorithm} \]

\[(U - UG2) \quad p_j = p_{j-1} + \alpha_j q_j \]

\[(U - UG3) \quad u_{j+1} = u_j + \alpha_j h_j \]

\[(U - UG4) \quad (q_{j+1}, q) = b(u_{j+1}, q) - \langle g_h, q \rangle, \quad \text{for all } q \in M_h. \]

In the second identity in \((UG\alpha)\), we involved Remark 2.1 and \((UG1)\). One can slightly modify the UG algorithm to obtain the UCG algorithm, as done in, e.g., [10, 33].

Algorithm 3.2. \((UCG)\) Algorithm

Step 1: Set \( u_0 = 0 \in V_h, p_0 \in M_h \). Compute \( u_1 \in V_h, q_1, d_1 \in M_h \) by

\[a(u_1, v) = (f_h, v) - b(v, p_0), \quad v \in V_h\]

\[(q_1, q) = b(u_1, q) - \langle g_h, q \rangle, \quad \text{for all } q \in M_h, \quad d_1 := q_1.\]

Step 2: For \( j = 1, 2, \ldots \), compute \( h_j, \alpha_j, p_j, u_{j+1}, q_{j+1}, \beta_j, d_{j+1} \) by

\[(UCG1) \quad a(h_j, v) = -b(v, d_j), \quad v \in V_h \]

\[(UCG\alpha) \quad \alpha_j = -\frac{(q_j, q_j)}{b(h_j, q_j)} = \frac{(q_j, q_j)}{(q_j, q_j)_{S_h}} \]

\[(UCG2) \quad p_j = p_{j-1} + \alpha_j d_j \]

\[(UCG3) \quad u_{j+1} = u_j + \alpha_j h_j \]

\[(UCG4) \quad (q_{j+1}, q) = b(u_{j+1}, q) - \langle g_h, q \rangle, \quad \text{for all } q \in M_h \]

\[(UCG\beta) \quad \beta_j = \frac{(q_{j+1}, q_{j+1})}{(q_j, q_j)} \]

\[(UCG6) \quad d_{j+1} = q_{j+1} + \beta_j d_j \]

Remark 3.3. It is not difficult to check that the UG and UCG algorithms produce the standard gradient and the standard conjugate gradient algorithms for solving the first equation in (2.10).

Theorem 3.4. Let \( (u_h, p_h) \) be the solution of (2.4), and let \( \{ (u_{j+1}, p_j) \} \) be the iterations produced by a U, UG, or UCG algorithm. Then, for \( j \geq 0 \),

\[(3.1) \quad u_{j+1} - u_h = A_h^{-1}B_h^T(p_h - p_j),\]

\[(3.2) \quad q_{j+1} = S_h(p_h - p_j),\]
and consequently, for $j \geq 1$,

\begin{equation}
\frac{1}{M_h^2} \|q_j\| \leq \|p_{j-1} - p_h\| \leq \frac{1}{m_h^2} \|q_j\|.
\end{equation}

\begin{equation}
\frac{m_h}{M_h^2} \|q_j\| \leq |u_j - u_h| \leq \frac{M_h}{m_h^2} \|q_j\|.
\end{equation}

**Proof.** By induction over $j$, it is easy to prove (for any of the U, UG, or UCG) that

\begin{equation}
a(u_{j+1} + 1, v) + b(v, p_j) = (f_h, v), \quad \text{for all } v \in V_h.
\end{equation}

Combining the first equation in (2.4) and (3.5), we get

\[ a(u_{j+1}, v) + b(v, p_j) = (f_h, v), \quad \text{for all } v \in V_h, \]

which gives (3.1). From (U4), (UG4), or (UCG4), the second equation of (2.6), and (3.1) we get

\[ q_{j+1} = B_h u_{j+1} - g_h = B_h (u_{j+1} - u_h) = S_h(p_h - p_j). \]

which proves (3.2). As a consequence of 3.1, the estimate (2.7), and Remark 2.1, for $j \geq 1$, we have

\begin{equation}
m_h \|p_h - p_{j-1}\| \leq |u_j - u_h| = \|p_h - p_{j-1}\| \leq M_h \|p_h - p_{j-1}\|.
\end{equation}

Using (3.2) and the fact that $m_h^2$ and $M_h^2$ are the extreme eigenvalues of $S_h$, we get

\begin{equation}
m_h^2 \|p_h - p_{j-1}\| \leq \|S_h(p_h - p_{j-1})\| = \|q_j\| \leq M_h^2 \|p_h - p_{j-1}\|.
\end{equation}

The estimates (3.3) and (3.4) are a direct consequence of (3.6) and (3.7). □

As a consequence of Theorem 3.4, we obtain

\begin{equation}
1 + \frac{m_h}{M_h^2} \|q_j\| \leq |u_h - u_j| + \|p_h - p_{j-1}\| \leq \frac{1 + M_h}{m_h^2} \|q_j\|,
\end{equation}

which says that $\|q_j\|$ is an estimator for the global iteration error providing good upper and lower bounds.

**Theorem 3.5.** Let $(u_h, p_h)$ be the solution of (2.4), and let $(u_{j+1}, p_j)_{j \geq 0}$ be the iterations produced by a U, UG or UCG algorithm. Then,

\[ (u_{j+1}, p_j) \to (u_h, p_h), \quad \text{and consequently } q_j \to 0. \]

**Proof.** For the Uzawa algorithm, it is easy to check that

\begin{equation}
p_h - p_j = (I - \alpha S_h)(p_h - p_{j-1}).
\end{equation}

Using that the eigenvalues of the symmetric operator $S_h$ are $m_h^2$ and $M_h^2$, we have

\begin{equation}
\|I - \alpha S_h\| = \max\{|1 - \alpha m_h^2|, |1 - \alpha M_h^2|\} < 1, \quad \text{for } \alpha \in \left(0, \frac{2}{M_h^2}\right).
\end{equation}
Thus, \( p_j \rightarrow p_h \). For the UG and UCG, by using Remark 3.3, the following estimates is well known from [23], [6] and others.

\[
\| p_h - p_j \|_{S_h} \leq \frac{M_h^2 - m_h^2}{M_h^2 + m_h^2} \| p_h - p_{j-1} \|_{S_h}.
\]

The estimate gives \( p_j \rightarrow p_h \). Using (3.1), we also get that \( u_{j+1} \rightarrow u_h \) for the U, UG, or UCG algorithm. The fact that \( q_j \rightarrow 0 \) follows from (3.3). \( \Box \)

4. CASCADIC ALGORITHM FOR SADDLE POINT PROBLEMS

In this section we will define concrete (CM) algorithms by specifying the process of Step CM2 and by defining a level change condition (LC). We will use the notation and the setting of the previous sections.

Assume that we can easily build a sequence of pairs \( \{ (V_k, M_k) \}_{k \geq 1} \) that satisfies a discrete inf – sup condition for every \( k \geq 1 \), and that \( h_k \) is a mesh parameter associated with the pair \( (V_k, M_k) \) such that \( h_k \rightarrow 0 \). We define

\[
m_k := \inf_{p_k \in M_k} \sup_{v_k \in V_k} \frac{b(v_k, p_k)}{\| p_k \| | v_k |}.
\]

and

\[
M_k := \sup_{p_k \in M_k} \sup_{v_k \in V_k} \frac{b(v_k, p_k)}{\| p_k \| | v_k |} \leq M.
\]

In order to prove the convergence of Algorithm 1.1 we further introduce the following assumptions:

(A_1) The family \( \{ (V_k, M_k) \}_{k \geq 1} \) is stable:
There exists \( m_0 > 0 \) such that \( m_k \geq m_0 \), for \( k = 1, 2, \ldots \).

(A_2) The process of Step CM2 is defined by Step 2 of U, UG or UCG algorithm. In the Uzawa solver case, we take \( \alpha_0 \in (0, 2/M^2) \).

(A_3) If \( (u, p) \) is the solution (1.1), and \( (u^{(k)}, p^{(k)}) \) the solution of (2.4) on \( (V_k, M_k) \), then there exist \( C_0 = C_0(u, p) \) and \( s > 0 \) independent of \( k \), such that

\[
|u - u^{(k)}| + \| p - p^{(k)} \| \leq C_0 h_k^s.
\]

(A_4) The level change condition is

\[
(LC) \quad \| q_{j+1} \| \leq C_{lc} h_k^s,
\]

where \( C_{lc} \) is a constant independent of \( k \).

We further consider that a sequence of prolongation operators \( P_{k,k+1} : M_k \rightarrow M_{k+1} \) is available. We are ready now to state our main result:
Theorem 4.1. Assume that (A_1) – (A_4) are satisfied. If (u_{j+1}, p_j) is the last iteration computed by the CM algorithm on (V_k, M_k), then there exists a constant C depending only on m_0, M, C_0, C_{lc}, and α_0 in the Uzawa level solver case, such that

\[ \| u - u_{j+1} \| + \| p - p_j \| \leq C h_k^s. \]  

Proof. From (4.5) and (4.2), we have that m_0 \leq m_k \leq M_k \leq M, for k = 1, 2, \ldots. Thus, from Theorem 3.4 or the equation (3.8) we get that

\[ \| u_{j+1} - u^{(k)} \| + \| p_j - p^{(k)} \| \leq C_1 \| q_{j+1} \|, \]

with C_1 depending only on m_0 and M (and α_0 in the U-case). If (u_{j+1}, p_j) is the last iteration computed by the CM algorithm on (V_k, M_k), then by assumption (A_4), we have \( \| q_{j+1} \| \leq C_{lc} h_k^s \), and consequently,

\[ \| u_{j+1} - u^{(k)} \| + \| p_j - p^{(k)} \| \leq C_1 C_{lc} h_k^s. \]

We note here that, due to Theorem 3.5, if infinitely many iterates would be performed on a fixed level, then we had that \( \| q_{j+1} \| \to 0 \), which contradicts the level change assumption (A_4). Consequently, on each level the algorithms perform a finite number of iterations. The convergence estimate (4.4) is a direct consequence of (A_3), (4.5), and the triangle inequality. \( \square \)

Remark 4.2. Let us make the following assumption on the prolongation operators \( p_{k,k+1} : M_k \to M_{k+1} \): There exists \( C_2 \) independent of \( k \) such that

\[ \| p_{k,k+1} p - p \| \leq C_2 h_k^s, \quad \text{for all } p \in M_k. \]

We further assume that (A_1) – (A_4) are satisfied with \( h_{k+1} \leq \rho_0 < 1 \), where \( \rho_0 \) is independent of \( k \). From Theorem 3.5, using (A_4), if \( p_{0,k} \) is the first iteration and \( p_{j,k} \) is the last iteration computed on (V_k, M_k), we have

\[ \| p^{(k)} - p_{j,k} \| \leq \rho^j \| p^{(k)} - p_{0,k} \| S_k, \]

where \( S_k \) is the discrete Schur complement for (V_k, M_k), and \( \rho \in (0,1) \) can be chosen independent of \( k \). Using (3.3) and the above two estimates, it is easy to prove that the number of iterations performed by CM on each level is bounded by a fixed number \( N_{\text{maxit}} \), depending only on \( \rho_0, m_0, M, C_0, C_2, C_{lc} \). Moreover, it is easy to check that \( N_{\text{maxit}} \) can be chosen independent of \( C_{lc} \) for \( k \geq 2 \). Thus, if the number of iteration (or the amount of work) of the CM algorithm is associated with a water cascade flow with the steps corresponding to our multilevel spaces, we can claim that the flow does not spread out. This makes our proposed algorithm a non-spread cascading iteration process.

The non-spread cascading phenomena can be also “watched” on the last column of Table 1-Table 4. In what follows, the CM algorithm with U, UG, or UCG as level solver defines the corresponding CMU, CMUG, or CMUCG algorithm.
5. Numerical results for the Stokes system

In this section, we show the numerical performance of the CM algorithm, emphasizing on the way one should choose the level change criterion once information about the order of the discretization error is available. We implemented the CM, CMUG, and CMUCG algorithms for the discretization of the standard Stokes system. For each level \( k \) we record the errors \(|u - u_{j+1}|\) and \(\|p - p_j\|\) where \((u_{j+1}, p_j)\) is the last iteration computed on \((V_k, M_k)\).

First, we considered \( \Omega \) to be the unit square \((0, 1)^2\) and defined the data for the Stokes system, such that the exact solution is 

\[
p = \frac{2}{3} - x^2 - y^2
\]

and

\[
u_1 = \nu_2 = \frac{1}{2\pi^2} \sin(\pi x) \sin(\pi y).
\]

We discretize using two known stable families of pairs: \(P_2 - P_0\), and \((P_2 - P_1)\) - Taylor-Hood (T-H). To construct the spaces \((V_k, M_k)\), we define the original triangulation \(T_1\) on \(\Omega\) given by the Union Jack pattern. The family of uniform meshes \(\{T_k\}_{k \geq 1}\) is defined by a uniform refinement strategy, i.e., \(T_{k+1}\) is obtained from \(T_k\) by splitting each triangle of \(T_k\) in four similar triangles. Since the sequence of spaces \(\{M_k\}\) is nested, in both \(P_2 - P_0\) and T-H discretizations, the prolongation operators \(P_{k,k+1} : M_k \rightarrow M_{k+1}\) are simply the embedding operators.

For the \(P_2 - P_0\) element, the discretization error is \(O(h)\), so we use the level change condition: (LC) \(\|q_{j+1}\| \leq \frac{1}{16} h_k\). For comparison between CM, CMUG, and CMUCG, see Table 1.

| \(\text{CMU, } \alpha = 0.8\) | \(|u - u_{j+1}|\) | Rate | \(\|p - p_j\|\) | Rate | \# of iter |
|---|---|---|---|---|---|
| \(k=4\) | 0.0384038 | 0.0434820 | 16 |
| \(k=5\) | 0.0207067 | 0.0264921 | 72 | 8 |
| \(k=6\) | 0.0108764 | 0.0156126 | 76 | 10 |
| \(k=7\) | 0.0052167 | 0.008639 | 85 | 11 |
| \(k=8\) | 0.0026943 | 0.0045779 | 92 | 11 |

| \(\text{CMUG}\) |
|---|---|---|---|---|---|
| \(k=4\) | 0.0386718 | 0.0464940 | 13 |
| \(k=5\) | 0.0201762 | 0.027242 | 78 | 6 |
| \(k=6\) | 0.0103487 | 0.0150688 | 86 | 6 |
| \(k=7\) | 0.0052530 | 0.0079920 | 91 | 5 |
| \(k=8\) | 0.0026532 | 0.0041757 | 94 | 5 |

| \(\text{CMUCG}\) |
|---|---|---|---|---|---|
| \(k=4\) | 0.0415028 | 0.0732229 | 9 |
| \(k=5\) | 0.0213266 | 0.0373755 | 97 | 3 |
| \(k=6\) | 0.0107704 | 0.0188775 | 98 | 4 |
| \(k=7\) | 0.0054313 | 0.0095173 | 99 | 3 |
| \(k=8\) | 0.0027269 | 0.0045623 | 99 | 3 |

Table 1. CM \(P_2 - P_0\) discretization with LC: \(\|q_{j+1}\| \leq \frac{1}{16} h_k\)
For the Taylor-Hood element, the discretization error is $O(h^2)$, so we use the level change condition: $(LC)$ $\|q_{j+1}\| \leq \frac{1}{16} h_k^2$. For a comparison between CMU, CMUG, and CMUCG, see Table 2. When we impose the iteration only on the last level ($k = 8$) using the same stopping criterion, we obtain similar errors, by using 45, 29, and 20 iterations for CMU, CMUG, and CMUCG, respectively. For the convex case, it seems that CMUCG has only a slightly better performance when compared with CMUG. Nevertheless, the advantage of CMUCG is more significant in the non-convex case.

Secondly, we performed numerical experiments for the Stokes system on the L-shaped domain $\Omega := (-1,1)^2 \setminus [0,1] \times [-1,0]$ using the $(P_2 - P_1)$-T-H discretization. We chose the data such that the exact solution is $u_1 = u_2 = r^{2/3} \sin(\frac{2\theta}{3})(1-x)(1-y)^2$, and $p = 2/3 - x^2 - y^2$. Note that $u_1 \notin H^{1+3/2}$. We used quasi-uniform meshes and graded meshes also. For both types of refinement, we started with the initial triangulation $T_1$ being the Union Jack pattern on each of the three unit squares of the domain. For the uniform refinement case, the family of quasi-uniform meshes $\{T_k\}_{k \geq 1}$ is defined by a uniform refinement strategy as in the convex case. For the graded meshes $T_{k+1}$ is obtained from $T_k$ by splitting each triangle of $T_k$ in four triangles as follows: we refine by dividing all the edges that contain the singular point $(0,0)$ under a fixed ratio $\kappa > 0$ such that the segment containing the singular point is $\kappa$-times the other segment, (see e.g., [4, 5]). We used $N_k = N_{d.o.f}$
as the complexity measure on \((V_k, M_k)\), where \(N_{d.o.f}\) is the number of degrees of freedom associated with a scalar discrete Laplacian on \(V_k\). For the uniform refinement, the discretization error is \(O(N_k^{-1/3})\), so we used the level change condition: (LC) \(\|q_{j+1}\| \leq \frac{1}{8} N_k^{-1/3}\). The performances of CMUG, and CMUCG are similar, see Table 3. For the graded meshes refinement, we experimented with various values of \(\kappa < 1\) in order to approach the optimal order of convergence \(O(N_k^{-1})\) exhibited in the convex case, and used the stopping criterion: \(\|q_{j+1}\| \leq \frac{1}{8} N_k^{-1}\). For the comparison between CMUG, and CMUCG, see Table 4.

|                | \(|u - u_{j+1}|\) | Rate | \(\|p - p_j\|\) | Rate | # of iter |
|----------------|-------------------|------|-----------------|------|----------|
| \(k=4\)       | 0.0509160         |      | 0.0157147       |      | 7        |
| \(k=5\)       | 0.0320362         | 0.67 | 0.0093171       | 0.75 | 2        |
| \(k=6\)       | 0.020178          | 0.67 | 0.0060782       | 0.62 | 2        |
| \(k=7\)       | 0.0127138         | 0.67 | 0.0038458       | 0.66 | 2        |
| \(k=8\)       | 0.0080074         | 0.67 | 0.0023277       | 0.72 | 2        |
| **CMUG, \(\kappa = 1\)** |                  |      |                 |      |          |
| \(k=4\)       | 0.0518455         |      | 0.0219035       |      | 4        |
| \(k=5\)       | 0.0320592         | 0.69 | 0.0092441       | 1.24 | 2        |
| \(k=6\)       | 0.0127307         | 0.67 | 0.0056854       | 0.70 | 2        |
| \(k=7\)       | 0.0126586         | 0.67 | 0.0036609       | 0.64 | 2        |
| \(k=8\)       | 0.0080042         | 0.67 | 0.0023205       | 0.71 | 2        |

Table 3. CM T-H on uniform refinement, LC: \(\|q_{j+1}\| < \frac{1}{8} N_{d.o.f}\)

|                | \(|u - u_{j+1}|\) | Rate | \(\|p - p_j\|\) | Rate | # of iter |
|----------------|-------------------|------|-----------------|------|----------|
| \(k=4\)       | 0.0185020         |      | 0.0052936       |      | 13       |
| \(k=5\)       | 0.0062828         | 1.56 | 0.0015040       | 1.81 | 7        |
| \(k=6\)       | 0.0019275         | 1.70 | 0.0003630       | 2.05 | 9        |
| \(k=7\)       | 0.0005533         | 1.80 | 0.0000852       | 2.09 | 9        |
| \(k=8\)       | 0.0001524         | 1.86 | 0.0000204       | 2.06 | 8        |
| **CMUCG, \(\kappa = 1/8\)** |                  |      |                 |      |          |
| \(k=4\)       | 0.0185008         |      | 0.0052371       |      | 7        |
| \(k=5\)       | 0.0062794         | 1.56 | 0.0014294       | 1.87 | 4        |
| \(k=6\)       | 0.0019272         | 1.70 | 0.0003316       | 2.10 | 5        |
| \(k=7\)       | 0.0005533         | 1.80 | 0.0000822       | 2.01 | 3        |
| \(k=8\)       | 0.0001524         | 1.86 | 0.0000191       | 2.10 | 4        |

Table 4. CM T-H on graded meshes with LC: \(\|q_{j+1}\| < \frac{1}{8} N_{d.o.f}\)
We note that by choosing graded meshes and an appropriate level change condition, we can improve the rate of convergence of the CM algorithm. Both CMUG and CMUCG recover a better than expected rate of convergence for the pressure and a close to optimal rate of convergence for the velocity. The optimal choice of the running parameters $C_{lc}$ and $\kappa$, together with finding quasi-optimal approximation spaces for Stokes and other SPPs on polygonal or polyhedral domains are challenging problems that will be further investigated.

6. Conclusion

We presented cascadic type algorithms for discretizing saddle point problems for the particular case when the form $a(\cdot, \cdot)$ is symmetric and coercive. The new algorithms are based on the existence of multilevel sequences of nested approximation spaces that are stable. We focused on cascadic multilevel algorithms of Schur complement type with Uzawa, Uzawa gradient and the Uzawa conjugate gradient as level solvers. The level change criterion requires that the iteration error be close to the expected discretization error, and we enforced it by using an efficient and easy to compute residual estimators for the iteration error. The theoretical results and experiments show that the algorithms can achieve optimal or close to optimal approximation rates by performing a non-increasing number of iterations on each level. The main computational challenge for each iteration is inverting operators of discrete Laplacian type. If we efficiently invert these operators, we obtain a significant reduction of the overall running time as we compare our CM algorithm with other non-multilevel iterative methods. 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, such as the div-curl system and the Maxwell equations (see [3, 13, 28]).

References


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