

Research Article

The Mixed Finite Element Multigrid Method for Stokes Equations

K. Muzhinji,¹ S. Shateyi,¹ and S. S. Motsa²

¹ Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa

² Department of Mathematics, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 28 July 2014; Revised 8 September 2014; Accepted 8 September 2014

Academic Editor: Emran Tohidi

Copyright © 2015 K. Muzhinji et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The stable finite element discretization of the Stokes problem produces a symmetric indefinite system of linear algebraic equations. A variety of iterative solvers have been proposed for such systems in an attempt to construct efficient, fast, and robust solution techniques. This paper investigates one of such iterative solvers, the geometric multigrid solver, to find the approximate solution of the indefinite systems. The main ingredient of the multigrid method is the choice of an appropriate smoothing strategy. This study considers the application of different smoothers and compares their effects in the overall performance of the multigrid solver. We study the multigrid method with the following smoothers: distributed Gauss Seidel, inexact Uzawa, preconditioned MINRES, and Braess-Sarazin type smoothers. A comparative study of the smoothers shows that the Braess-Sarazin smoothers enhance good performance of the multigrid method. We study the problem in a two-dimensional domain using stable Hood-Taylor Q_2-Q_1 pair of finite rectangular elements. We also give the main theoretical convergence results. We present the numerical results to demonstrate the efficiency and robustness of the multigrid method and confirm the theoretical results.

1. Introduction

This study considers the numerical solution of the large scale linear algebraic system arising from the discretization of the partial differential equations. The discretization is achieved by the finite element method. For positive definite linear systems, linked to the Poisson equations, the multigrid (MGM) methods are proven to be the most effective and fast methods [1, 2]. However it is more challenging for linear indefinite algebraic systems. In this paper we consider multigrid methods for solving linear indefinite algebraic system of equations arising from the mixed finite element discretization of the steady state Stokes problem:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \Omega, \quad (1)$$

$$\operatorname{div} \mathbf{u} = 0, \quad \text{in } \Omega, \quad (2)$$

$$\mathbf{u} = 0, \quad \text{on } \partial\Omega, \quad (3)$$

where \mathbf{u} is a velocity field, p represents pressure, and \mathbf{f} is an external force field. The problem is considered with (1)–(3) defined on the domain $\Omega \subseteq \mathcal{R}^2$ with boundary $\partial\Omega$.

The main goal of this work is to construct and analyze numerical methods that produce an appropriate solution to the Stokes problem. The main thrust is to apply an iterative method, the multigrid method, to solve the linear system of equations that arise from the discretization of the Stokes equations. The MFEM applied to (1)–(3) with carefully chosen finite element spaces results in the algebraic system which must be solved. The velocity variable \mathbf{u} together with the pressure variable p is the solutions of the system. We discretize the domain of the Stokes problem by the rectangular grids with a pair of conforming mixed finite element spaces that are inf-sup stable. In our experiment we use Hood-Taylor Q_2-Q_1 pair as used by [3]. The process produces a symmetric indefinite system of linear algebraic equations. In this paper we study an efficient solver for this system. This work on multigrid method has been motivated

by the need to effectively and efficiently solve large application problems. The multigrid method has been shown to be very efficient and successful in solving control problems [1, 2, 4–6] and elliptic partial differential equations [7–9] in an accurate and computationally efficient way. The multigrid method has been applied to problems discretized by the finite difference method and widely by finite element method [3, 8, 10–14]. The effectiveness of the multigrid method depends on the correct choice of the smoothers. Various smoothers have been suggested in literature weighted Jacobi, Gauss Seidel [11], Ilu [8], Vanka-type [9, 12, 13, 15], Braess-Sarazin-type ([13, 16–18]), Semi implicit method for pressure linked equations [15], SOR/Richardson [18], and inexact Uzawa [18]. It is the purpose of this study to apply the multigrid solver to the Stokes problem with the following iterative solvers as smoothers: Braess-Sarazin, inexact Uzawa, preconditioned MINRES, and the distributed Gauss Seidel. The inner solver of these smoothers can also be taken as the multigrid method for the definite subsystems. There is no work known where a comparative study is made on the effects of these four smoothers on the performance of the multigrid method for indefinite systems. The first step is to transform the continuous problem to the discrete system and apply the MFEM that produces the linear algebraic system on which the multigrid method is developed, analyzed, and finally numerically and computationally implemented.

The key features and ingredients of the multigrid method are smoothing and coarse grid correction that involves the intergrid transfers and a solution correction step. The main results of the work are the convergence of the multigrid method in calculating the velocity and pressure variables in an appropriate norm which is based on the smoothing and approximation properties [9, 18]. The rest of the paper is organized as follows. In Section 2 we give the discrete system of the Stokes problem by mixed finite element method. In Section 3 the iterative solution technique, the geometric method, and smoothers are outlined. The known theoretical convergence analysis results are also outlined. In Section 4 a numerical experimental and comparative analysis on the effects of smoothers on the performance multigrid method is presented and discussed and the conclusion is given.

2. The Stokes Discrete System

For the discretization of the Stokes equations in the domain Ω we need to transform the system (1)–(3) to the weak variational form. For the weak variational formulation of the Stokes equations we define the following solution and test spaces:

$$\begin{aligned} H^1(\Omega) &:= \{ \mathbf{u} : \Omega \rightarrow \mathcal{R} \mid \mathbf{u}, \nabla \mathbf{u} \in L^2(\Omega) \}, \\ H_0^1(\Omega) &:= \{ \mathbf{v} : \mathbf{v} \in H^1 \mid \mathbf{v} = 0 \text{ on } \partial\Omega \}. \end{aligned} \tag{4}$$

By multiplication of the first equation (1) with $\mathbf{v} \in H_0^1$ and the second equation (2) with $q \in L^2(\Omega)$, subsequently integrating over the domain Ω , applying the Gauss theorem, and incorporating the boundary condition (3), we obtain the variational form.

Find $\mathbf{u} \in H_0^1(\Omega)$ and $p \in L^2(\Omega)$ such that

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) - b(\mathbf{v}, p) &= F(\mathbf{v}), \quad \forall \mathbf{v} \in H_0^1(\Omega), \\ b(\mathbf{u}, q) &= 0, \quad \forall q \in L^2(\Omega), \end{aligned} \tag{5}$$

where $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are continuous bilinear forms defined as

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} \, dx, \\ b(\mathbf{u}, q) &= \int_{\Omega} (\operatorname{div} \mathbf{v}) q \, dx, \\ F(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \end{aligned} \tag{6}$$

where $\nabla \mathbf{u} : \nabla \mathbf{v}$ represents a componentwise scalar product that is $\nabla u_x \cdot \nabla v_x + \nabla u_y \cdot \nabla v_y$, and $a : H_0^1(\Omega) \times H_0^1(\Omega) \rightarrow \mathcal{R}$ and $b : H_0^1(\Omega) \times L^2(\Omega) \rightarrow \mathcal{R}$. The well-posedness follows from the coercivity of $a(\cdot, \cdot)$ in the Lax-Milgram theorem [19, 20] and partly from the inf-sup condition [7, 8, 16, 21–23]. Below is a sketch of the analysis of the existence uniqueness and stability of the solution $(\mathbf{u}, p) \in \mathbf{V} \times W = H_0^1(\Omega) \times L^2(\Omega)$ of mixed problem (5):

(i) the bilinear form $a(\cdot, \cdot)$ is bounded or continuous if

$$|a(\mathbf{u}, \mathbf{v})| \leq \alpha \|\mathbf{u}\|_{\mathbf{V}} \|\mathbf{v}\|_{\mathbf{V}}, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}, \alpha \in \mathcal{R}; \tag{7}$$

(ii) the bilinear form $a(\cdot, \cdot)$ is coercive on $V := H_0^1(\Omega)$; that is, there exists a positive constant α_1 :

$$a(\mathbf{v}, \mathbf{v}) \geq \alpha_1 \|\mathbf{v}\|_{\mathbf{V}}^2, \tag{8}$$

$$\forall \mathbf{v} \in \mathbf{V} = \ker B = \{ \mathbf{v} \in \mathbf{V} : b(\mathbf{v}, q) = 0 \ \forall q \in W \};$$

(iii) the bilinear form $a(\cdot, \cdot)$ is symmetric and nonnegative if

$$a(\mathbf{u}, \mathbf{v}) = a(\mathbf{v}, \mathbf{u}), \quad a(\mathbf{v}, \mathbf{v}) \geq 0, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}; \tag{9}$$

(iv) the bilinear form $b(\cdot, \cdot)$ is bounded if

$$|b(\mathbf{u}, q)| \leq \alpha_0 \|\mathbf{u}\|_{\mathbf{V}} \|q\|_W, \quad \forall \mathbf{u} \in \mathbf{V}, q \in W, \alpha_0 \in \mathcal{R}; \tag{10}$$

(v) the bilinear form $b(\cdot, \cdot)$ satisfies the inf-sup condition; that is, there exists a constant β :

$$\inf_{0 \neq q \in W} \sup_{0 \neq \mathbf{v} \in \mathbf{V}} \frac{|b(\mathbf{v}, q)|}{\|\mathbf{v}\|_{\mathbf{V}} \|q\|_W} \geq \beta > 0. \tag{11}$$

For instance, in [22], it is shown that in our concrete case $b(\cdot, \cdot)$ fulfills the inf-sup condition; thus we can combine (i)–(v) to give the following theorem.

Theorem 1. *The variational problem (5) is uniquely solvable provided properties (i)–(v) are all satisfied.*

The proof relies on the closed range theorem and on the Lax-Milgram theorem. The details can be found in [22, 24].

2.1. The Mixed Finite Element Discretization. The mixed finite element discretization of the weak formulation of the Stokes equations produces a linear algebraic system of equations. The finite element method described here is based on [7, 16, 19, 21, 23, 25]. We will introduce the concept of mixed finite element methods. Details can be found in [7, 19–21, 25].

We assume that $\Omega \subseteq \mathcal{R}^2$. We define the finite dimensional spaces. Let W_h and V_h be subspaces of W and V , respectively.

Now we can formulate a discrete version of problem (5). Find a couple $(\mathbf{u}_h, p_h) \in V_h \times W_h$ such that

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}_h) - b(\mathbf{v}_h, p_h) &= F(\mathbf{v}_h), \quad \forall \mathbf{v}_h \in V_h, \\ b(\mathbf{u}_h, q_h) &= 0, \quad \forall q_h \in W_h. \end{aligned} \tag{12}$$

The finite element discretization should satisfy the inf-sup condition. The following theorem shows that again the inf-sup condition is of major importance (for the proof we refer to [22]).

Theorem 2. Assume that a is V_h -elliptic (with h independent ellipticity constant) and that there exists a constant $\beta > 0$ (independent of h) such that the discrete inf-sup condition

$$\inf_{0 \neq q_h \in W_h} \sup_{0 \neq \mathbf{v}_h \in V_h} \frac{|b(\mathbf{v}_h, q_h)|}{\|\mathbf{v}_h\|_{V_h} \|q_h\|_{W_h}} \geq \beta > 0 \tag{13}$$

holds. Then the associated (discretized, steady state) Stokes problem has a unique solution (\mathbf{u}_h, p_h) , and there exists a constant β_1 such that

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_V + \|p - p_h\|_W \\ \leq \beta_1 \left(\inf_{\mathbf{v} \in V_h} \|\mathbf{u} - \mathbf{v}\|_V \right) + \left(\inf_{q \in W_h} \|p - q\|_W \right). \end{aligned} \tag{14}$$

If the basis of W_h is given by $\{\psi_1, \dots, \psi_m\}$ and of V_h is given by $\{\varphi_1, \dots, \varphi_n\}$, then

$$\begin{aligned} \mathbf{u}_h &= \sum_{i=1}^{n_i} \mathbf{u}_i \cdot \varphi_i + \sum_{i=n_i+1}^{n_i+n_\partial} \mathbf{u}_i \cdot \varphi_i, \\ p_h &= \sum_{k=1}^m p_k \psi_k, \end{aligned} \tag{15}$$

where n_i is the number of inner nodes and n_∂ is the number of boundary nodes so the coefficients $\mathbf{u}_i : i = n_i + 1, \dots, n_i + n_\partial$ interpolate the boundary data and $n = n_i + n_\partial$. The mixed finite element entails partitioning of the solution domain Ω into quadrilaterals; in our case that is $\Omega = \cup_i \tau_i$ we denote a set of rectangular (square) elements by $T_h = \{\tau_1, \tau_2, \tau_3, \dots\}$ and on each element τ_i and we denote the space $P_k(\tau_i)$ of degree less than or equal to k . There are a variety of finite element pairs whose effectiveness is through stabilization [26]. In this work we are going to use Hoods-Taylor Q_2 - Q_1 square finite elements which are known to be stable.

We specify

$$\begin{aligned} V_h &:= \{ \mathbf{u}_h \in \mathbf{V} | \mathbf{u}_h|_{\tau_i} \in P_2(\tau_i), \forall \text{ elements } \tau_i \}, \\ W_h &:= \{ p_h \in W | p_h|_{\tau_i} \in P_1(\tau_i), \forall \text{ elements } \tau_i \}. \end{aligned} \tag{16}$$

An element $(\mathbf{u}_h, p_h) \in W_h \times V_h$ is uniquely determined by specifying d components of \mathbf{u}_h on the nodes and on the midpoints of the edges of the elements and the values of p_h on the nodes of the elements. The mixed finite element method results in the coupled linear algebraic system which has to be solved by the appropriate solvers. The resulting system is

$$\begin{bmatrix} A_h & B_h^T \\ B_h & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h \\ g_h \end{bmatrix}; \tag{17}$$

with A_h being a block Laplacian matrix and B_h being the divergence matrix whose entries are given by

$$\begin{aligned} A &= [a_{ij}], \quad a_{ij} = \int_{\Omega} (\nabla \phi_i : \nabla \phi_j)_{i,j=1,\dots,n}, \\ B &= [b_{ki}], \quad b_{ki} = - \int_{\Omega} (\psi_k \nabla \cdot \phi_i)_{i=1,\dots,n; k=1,\dots,m}. \end{aligned} \tag{18}$$

The entries of the right hand side vector are

$$\mathbf{f} = [\mathbf{f}_i], \quad \mathbf{f}_i = \int_{\partial\Omega} \mathbf{f} \cdot \phi_i - \sum_{i=n+1}^{n+n_\partial} \mathbf{u}_i \int_{\Omega} (\nabla \phi_i : \nabla \phi_j), \tag{19}$$

$$g = [g_k], \quad g_k = \sum_{i=n+1}^{n+n_\partial} \mathbf{u}_i \int_{\Omega} (\psi_k \nabla \cdot \phi_i).$$

The linear algebraic system can be represented as

$$\mathcal{M}x = b, \tag{20}$$

where $\mathcal{M} := \begin{bmatrix} A_h & B_h^T \\ B_h & O \end{bmatrix}$, $x := \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix}$, and $b := \begin{bmatrix} \mathbf{f}_h \\ g_h \end{bmatrix}$.

The solution vectors (\mathbf{u}_h, p_h) from (15) are the mixed finite element solution. The system (17)–(19) is called the discrete Stokes problem.

The discretization and assembling of matrices are done by the MATLAB implementation of the mixed finite element method [8]. A_h is stiffness matrix resulting from the discretization of the Laplacian. The resultant coefficient matrix is large, sparse, indefinite and the system must be solved iteratively, in this case by multigrid solvers. The multigrid solver is a well known fast solver for the elliptic partial differential equations [2, 5].

3. Multigrid Method

The main focus of this section is the construction of the multigrid solver to find the approximate solution of (20) at the finest mesh/discretization. Let $(V_l \times W_l)$ be a sequence of subspaces of the finite dimensional subspaces W_h and V_h defined on sequence of grids $l \in \{0, 1, 2, 3, \dots, l_{\max}\}$ with mesh sizes $h_0, h_1, h_2, \dots, h_l$ with $h_{l+1} := (1/2)h_l$. We define a hierarchy/family of nested finite element subspaces for the velocity and pressure:

$$\begin{aligned} (V_l \times W_l) &\subset (V_{l+1} \times W_{l+1}) \subset (V_h \times W_h) \subset (V \times W) \\ &= H_0^1(\Omega) \times L^2(\Omega), \end{aligned} \tag{21}$$

where $(V_{l+1} \times W_{l+1})$ subspace which corresponds to Ω_{l+1} is the refinement of Ω_l with subspace $(V_l \times W_l)$ such that

$\Omega_l \subset \Omega_{l+1} \subset \Omega$. At the discrete level with the defined discrete spaces and bases, the linear algebraic system is defined by

$$\mathcal{M}_l x_l = b_l, \tag{22}$$

where $\mathcal{M}_l := \begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix}$, $x_l := [\mathbf{u}_l \ p_l]$, and $b_l := \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix}$.

The main goal is to find the pair $x_l = (\mathbf{u}_l, p_l)$ of the discrete velocity and the discrete pressure variables at the finest level l .

Now we introduce the multigrid iteration for solving the discretized equation (22) on grid l . We define the multigrid algorithm at level l as $\text{MGM}_l(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}}, \mathbf{u}_l^{\text{old}}, p_l^{\text{old}}, \mathbf{f}_l, g_l, m_1, m_2)$, where

- (i) $(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}})$ is the output of velocity and pressure after one step of the multigrid algorithm at level l ;
- (ii) $\mathbf{u}_l^{\text{old}}$ is the input velocity at level l ;
- (iii) p_l^{old} is the input pressure at level l ;
- (iv) $R_{\mathbf{u},l,l-1}$ and $R_{p,l,l-1}$ are restriction operators for velocity and pressure, respectively, from level l to level $l-1$;
- (v) $P_{\mathbf{u},l-1,l}$ and $P_{p,l-1,l}$ are prolongation operators for velocity and pressure, respectively, from level $l-1$ to level l .

Algorithm 3 (multigrid algorithm).

$$\text{MGM}_l(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}}, \mathbf{u}_l^{\text{old}}, p_l^{\text{old}}, \mathbf{f}_l, g_l, m_1, m_2) \tag{23}$$

if $l = 0$ (coarsest grid)

$$\begin{aligned} & \begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_l^{\text{new}} \\ p_l^{\text{new}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix} \\ &= \text{MGM}_l(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}}, \mathbf{u}_l^{\text{old}}, p_l^{\text{old}}, \mathbf{f}_l, g_l, m_1, m_2) \end{aligned} \tag{24}$$

else $l > 0$ define $\text{MGM}_l(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}}, \mathbf{u}_l^{\text{old}}, p_l^{\text{old}}, g_l, \mathbf{f}_l, m_1, m_2)$

- (1) Pre-Smoothing: Smoothing operator \mathcal{S} starting with, $(\mathbf{u}_l^{\text{old}}, p_l^{\text{old}})$ with m_1 smoothing steps, producing $(\tilde{\mathbf{u}}_l, \tilde{p}_l)$,

$$\begin{bmatrix} \tilde{\mathbf{u}}_l \\ \tilde{p}_l \end{bmatrix} = \begin{bmatrix} \mathbf{u}_l^{\text{old}} \\ p_l^{\text{old}} \end{bmatrix} - \mathcal{S}^{-1} \left(\begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_l^{\text{old}} \\ p_l^{\text{old}} \end{bmatrix} - \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix} \right) \tag{25}$$

- (a) defect/residual

$$\begin{aligned} \mathbf{r}_l &= \mathbf{f}_l - (A_l \tilde{\mathbf{u}}_l + B_l^T \tilde{p}_l), \\ d_l &= g_l - B_l \tilde{\mathbf{u}}_l \end{aligned} \tag{26}$$

- (2) restrict the defect

$$\begin{aligned} \mathbf{r}_{l-1} &= R_{\mathbf{u},l,l-1} \mathbf{r}_l, \\ d_{l-1} &= R_{p,l,l-1} d_l, \end{aligned} \tag{27}$$

- (3) approximate solution

$$\begin{bmatrix} A_{l-1} & B_{l-1}^T \\ B_{l-1} & O \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}}_{l-1} \\ \tilde{q}_{l-1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{l-1} \\ d_{l-1} \end{bmatrix} \tag{28}$$

- (4) applying one/two iterations of MGM_{l-1} at the recursive call:

- (a) apply μ steps of MGM_{l-1}
- (b) Set $\mathbf{v}_{l-1}^0 = 0$
- (c) Set $q_{l-1}^0 = 0$
- (d) compute
for $\mu = 1 : 2$

$$\begin{aligned} & (\tilde{\mathbf{v}}_{l-1}, \tilde{q}_{l-1}) \\ &= \text{MGM}_{l-1}(\tilde{\mathbf{v}}_{l-1}, \tilde{q}_{l-1}, \mathbf{v}_{l-1}^0, q_{l-1}^0, \mathbf{r}_{l-1}, d_{l-1}, m_1, m_2); \end{aligned} \tag{29}$$

end

- (5) *Correction Step* define the new iterate:

$$\begin{aligned} \mathbf{u}_l^* &:= \tilde{\mathbf{u}}_l - P_{\mathbf{u},l-1,l} \tilde{\mathbf{v}}_{l-1}, \\ p_l^* &:= \tilde{p}_l - P_{p,l-1,l} \tilde{q}_{l-1}. \end{aligned} \tag{30}$$

Postsmoothing. Starting with (\mathbf{u}_l^*, p_l^*) perform m_2 smoothing steps using smoothing operator \mathcal{S} to produce $(\mathbf{u}_l^{\text{new}}, p_l^{\text{new}})$:

$$\begin{bmatrix} \mathbf{u}_l^{\text{new}} \\ p_l^{\text{new}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{u}}_l^* \\ \tilde{p}_l^* \end{bmatrix} - \mathcal{S}^{-1} \left(\begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_l^* \\ \tilde{p}_l^* \end{bmatrix} - \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix} \right). \tag{31}$$

The multigrid method described above belongs to a class of optimal order methods for solving linear systems emanating from the discretization techniques like the finite element method. Its known convergence speed does not deteriorate when the discretization is refined whereas classical iterative solvers slow down for the decreasing mesh size [1, 2, 5, 6]. The starting point of the multigrid concept is the observation that classical iteration methods have some smoothing properties. The operator \mathcal{S} represents such methods; in this study it represents Braess-Sarazin, inexact Uzawa, distributed Gauss Seidel, and the preconditioned minimum residual method. These methods are characterized by poor/slow global convergence rates and for errors whose length scales are comparable to mesh sizes, they provide rapid damping leaving behind smooth, longer wave length errors. These smooth parts of the error are responsible for the poor convergence. A geometric multigrid method involves a hierarchy of meshes and related discretization. A quantity that is smooth on a certain grid can also be approximated on a coarser grid. Low frequency error components can be effectively reduced by a coarse grid correction procedure. Since the action of a smoothing iteration leaves only smooth error components, it is possible to represent them as the solution of an appropriate coarser system. Once this coarser problem is solved, the solution is interpolated back to the fine grid to correct the fine approximation for its low frequency errors.

The most essential ingredients of the multigrid method are the smoothing operator, for which using a wrong smoother will destroy the efficiency of the entire multigrid method, and the coarse grid correction which involves the prolongation and the restriction operators. In multigrid methods we have to transform information from one grid to another and for that purpose we use prolongations and restrictions operators. Restriction transfers values from fine grid to the next coarse grid. Prolongation transfers values from the coarse grid to the next fine grid.

Next we discuss the key components of the multigrid method.

- (a) Intergrid transfer operators: the intergrid transfer operators are the restriction and prolongation between different grid levels. The restriction operator maps the residual from the finer grid to a coarser grid while the prolongation operator transfers vectors from coarse grid to fine grid. The restriction between levels l and $l - 1$ is defined by

$$R_{(l,l-1)} := \begin{pmatrix} R_{(u,l,l-1)} & 0 \\ 0 & R_{(p,l,l-1)} \end{pmatrix}, \quad (32)$$

where the restriction operators $R_{(u,l,l-1)} : \mathcal{R}^{n_l} \rightarrow \mathcal{R}^{n_{l-1}}$ and $R_{(p,l,l-1)} : \mathcal{R}^{m_l} \rightarrow \mathcal{R}^{m_{l-1}}$ for velocity and pressure, respectively. The prolongation between levels $l - 1$ and l is defined again as

$$P_{(l-1,l)} := \begin{pmatrix} P_{(u,l-1,l)} & 0 \\ 0 & P_{(p,l-1,l)} \end{pmatrix}, \quad (33)$$

where the prolongation operators $P_{(u,l-1,l)} : \mathcal{R}^{n_{l-1}} \rightarrow \mathcal{R}^{n_l}$ and $P_{(p,l-1,l)} : \mathcal{R}^{m_{l-1}} \rightarrow \mathcal{R}^{m_l}$ are representations of the following relations $\mathbf{V}_{l-1} \subset \mathbf{V}_l$ for the quadratic interpolation of the velocity (Q_2) and $W_{l-1} \subset W_l$ for the linear interpolation of the pressure (Q_1).

- (b) Coarse grid correction: the other key ingredient of the multigrid method is the coarse grid correction. In the multigrid solution process we need to solve the problem at the finest define level $l = l_{\max}$. The problem is defined on the coarser grid levels and on the coarsest grid level the problem is solved exactly. There are very few situations in which a grid can be coarsened to the extent that it is not practical to solve the problem using a direct method but iteratively. In this work the iterative solver used as a smoother is applied to solve the problem at the coarsest level.

3.1. The Smoothers. The most crucial part is the proper choice of a smoothing technique. Usually, the well-known smoothing iterations for the scalar problems (damped Jacobi or Gauss-Seidel relaxation) are not appropriate for saddle point problems or are even not defined, for example, in saddle point systems like (22). There are natural ways to generalize scalar smoothing schemes to systems of PDEs. The smoothing process is the main ingredient of the multigrid method. The convergence of the multigrid method is influenced by the smoothing process [11, 14, 18]. We perform a number of

iterations of an iterative solver to smooth the residual. The main goal is to compare the effectiveness of different iterative schemes as smoothers of the multigrid methods. On each level of a multigrid method, a system involving operator \mathcal{S} has to be solved approximately. The smoother dumps out highly oscillating error modes of the systems. In this paper we consider the following smoothing process:

$$\begin{pmatrix} \mathbf{u}_l^{i+1} \\ p_l^{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_l^i \\ p_l^i \end{pmatrix} - \mathcal{S}_l^{-1} \left[\begin{pmatrix} A_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \mathbf{u}_l^i \\ p_l^i \end{pmatrix} - \begin{pmatrix} \mathbf{f}_l \\ 0 \end{pmatrix} \right]. \quad (34)$$

Several smoothers have been proposed and applied in literature. Brandt [4] advocates for the use of the distributed Gauss Seidel smoothing. The Vanka-type smoother is widely used with a coupled Gauss Seidel scheme [13, 14] that introduces the idea of transforming smoothers and combines with incomplete factorization to develop an efficient smoothing. John and Tobska [14] and Pernice [15] used the Braess-Sarazin-type smoother with the Schur complement schemes as smoothers which exhibit wonderful smoothing properties. The following algorithms describe the iterative schemes that are used as smoothers in this study.

3.1.1. Braess-Sarazin-Type Smoother. The Braess-Sarazin smoothers proposed in [17] and used in [13, 18] solve a large saddle point problem in each smoothing step. This Braess-Sarazin or SIMPLE-type iteration uses $\begin{pmatrix} \widehat{A} & B^T \\ B & O \end{pmatrix}$ as a smoother for the saddle point problem (22). The smoother as presented in [17] and generalised in [18] consisted of constant application of the smoothing iteration:

$$\begin{pmatrix} \mathbf{u}_l^{i+1} \\ p_l^{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_l^i \\ p_l^i \end{pmatrix} - \begin{pmatrix} \widehat{A}_l & B_l^T \\ B_l & O \end{pmatrix}^{-1} \left[\begin{pmatrix} A_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \mathbf{u}_l^i \\ p_l^i \end{pmatrix} - \begin{pmatrix} \mathbf{f}_l \\ g_l \end{pmatrix} \right] \quad (35)$$

with $\widehat{A}_l = \alpha \text{diag}(A_l)$ and $\alpha = 2$ given. The smoothing Braess-Sarazin iteration (35) solves the auxiliary problem

$$\begin{pmatrix} \alpha \widehat{A}_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \widehat{\mathbf{u}}_l^i \\ \widehat{p}_l^i \end{pmatrix} = \begin{pmatrix} \mathbf{r}_l^i \\ s_l^i \end{pmatrix} \quad (36)$$

with $\mathbf{r}_l^i = A_l \mathbf{u}_l^i + B_l^T p_l^i - \mathbf{f}_l$ and $s_l^i = B_l \mathbf{u}_l^i - g_l$. Inherent in the system system (36) is the problem of the auxiliary pressure variable \widehat{p}_l

$$\widehat{S}_l \widehat{p}_l = B_l \widehat{A}_l^{-1} \mathbf{r}_l^i - \alpha B_l \mathbf{u}_l^i. \quad (37)$$

This system is solved approximatively by an iterative solver. From the system we get \widehat{p}_l approximatively which can be used to approximatively determine $\widehat{\mathbf{u}}_l$ from

$$\alpha \widehat{A}_l \widehat{\mathbf{u}}_l = \mathbf{r}_l^i - B_l^T \widehat{p}_l. \quad (38)$$

3.1.2. Inexact Uzawa Type Smoothers. The variant of the inexact Uzawa iteration used as a smoother is outlined.

Algorithm 4. (1) For $i = 1$: smoothing steps.

(2) Compute the residual $\mathbf{r}_i = \mathbf{f} - \mathbf{A}\mathbf{u}_i - B^T p_i$.

- (3) Compute the residual $s_i = g - B\mathbf{u}_i$.
- (4) Solve $\widehat{A}\mathbf{w}_i = \mathbf{r}_i$.
- (5) Solve $\widehat{S}\mathbf{d}_i = B^T\mathbf{w}_i - s_i$.
- (6) Solve $\widehat{A}\mathbf{w}_i = \mathbf{r}_i - B\mathbf{d}_i$.
- (7) Update the velocity and pressure

$$\begin{pmatrix} \mathbf{u}_{i+1} \\ p_{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_i \\ p_i \end{pmatrix} + \begin{pmatrix} \mathbf{w}_i \\ \mathbf{d}_i \end{pmatrix}; \tag{39}$$

End.

Step (6) in the outline may be rearranged as $\mathbf{w}_i := \mathbf{w}_i - \widehat{A}^{-1}(B\mathbf{d}_i)$ with $\widehat{A}^{-1}(B\mathbf{d}_i)$ obtained as a by-product of step (5). This saves the application of \widehat{A}^{-1} at the end of every outer iteration and hence improves the efficiency of the algorithm. The other variants of the inexact Uzawa method are analysed in [26–28].

3.1.3. The Distributed Gauss Seidel Type Smoothers (DGS). The standard smoothing iteration schemes like Jacobi and Gauss Seidel smoothers are not applicable to the system (22) because of the nature of the coefficient matrix; particularly the zero block in the diagonal hampers the smoothing process. The distributive smoother transforms the vital operators to the main diagonal and applied as a decoupled smoother. The DGS was introduced in [4] is related to a successive application of standard Gauss Seidel applied to the matrix operator \mathcal{M} (22) and $\mathcal{G} = \begin{pmatrix} I_l & B_l^T \\ O & -B_l B_l^T \end{pmatrix}$ with $\mathcal{M}\mathcal{G} = \begin{pmatrix} A_l & B_l^T \\ B_l & B_l B_l^T \end{pmatrix}$. We solve the transformed residual equation

$$\begin{pmatrix} \widehat{A}_l & B_l^T \\ B_l & \widehat{A}_p \end{pmatrix} \begin{pmatrix} \mathbf{w}_i \\ q_i \end{pmatrix} = \begin{pmatrix} \mathbf{r}_u \\ r_p \end{pmatrix} \tag{40}$$

with \widehat{A} and \widehat{A}_p being invertible approximations of A and $A_p := BB^T$, respectively. A single iteration with the update through a distributive matrix \mathcal{G} is performed by the following algorithm.

Algorithm 5 ($[\mathbf{u}^{i+1}, p^{i+1}] \leftarrow \text{DGS}(\mathbf{u}^i, p^i)$). (1) Smooth momentum equations

$$\mathbf{w} = \mathbf{u}^i + \widehat{A}^{-1}(\mathbf{f} - A\mathbf{u}^i - B^T p^i). \tag{41}$$

- (2) Smooth the transformed continuity equation

$$q = \widehat{A}_p^{-1}(g - B\mathbf{w}). \tag{42}$$

- (3) Transform the correction back to the original variables

$$\begin{aligned} \mathbf{u}^{i+1} &= \mathbf{w} + B^T q, \\ p^{i+1} &= p^i - BB^T q. \end{aligned} \tag{43}$$

The DGS has been widely used as a smoother for the finite difference discretization. In this paper the DGS type smoothers are used for finite element discretization of the Stokes problem.

3.1.4. The Preconditioned Minimum Residual Smoother. The preconditioned minimum residual method is a Krylov subspace method for solving symmetric indefinite systems and uses popular block preconditioners. This method is used as a smoother for the multigrid method of the Stokes problem in this paper. For the Stokes equations, the classical block-diagonal preconditioner for MINRES method [8] is

$$P = \begin{pmatrix} \widehat{A} & 0 \\ 0 & \widehat{S} \end{pmatrix} \tag{44}$$

with $\widehat{S} = B\widehat{A}^{-1}B^T$. The block preconditioning requires the solution of two systems of equations with matrices \widehat{A} and \widehat{S} at each MINRES iteration. If P^{-1} is computed exactly, the preconditioned Krylov methods converge in two or three steps [10]. For practical implementations, the Schur complement \widehat{S} is replaced by the mass matrix M_p of the pressure space. For discontinuous pressure space, M_p is block diagonal and easy to invert. For continuous pressure space, say Q_1 , the mass matrix M_p can be further replaced by its diagonal matrix [8].

3.2. Multigrid Convergence. The convergence analysis of the multigrid method relies on the two properties, namely, the approximation and the smoothing. The general convergence rates are independent of h (the mesh size), l is the level of discretization, and m_1 and m_2 are the number of pre- and postsmoothing iterations [1, 12, 18]. The results for the convergence of the multigrid method for the scalar elliptic problems cannot apply to the Stokes equations. We provide a snapshot of the available convergence results of the multigrid method for Stokes equations. The ideas presented in this paper are based on the work in [12, 16, 18]. An iteration of single multigrid step consists of a combination of smoothing step and a coarse grid correction step. We will consider the multigrid convergence with the Braess-Sarazin smoother with \mathcal{S}_l being the iteration matrix of the smoother (34) and the \mathcal{M}_l being the Stokes stiffness matrix in (22). The operator P_l and its adjoint R_l are intergrid transfer operators, prolongation, and the restriction, respectively. The convergence analysis of the multigrid method begins with the analysis of the two-grid method, with m_1 and m_2 the pre- and post-smoothing steps, respectively, applied to (22) results in the iteration matrix

$$L_l = \mathcal{S}_l^{m_1} (I_l - P_l \mathcal{M}_{l-1}^{-1} R_l \mathcal{M}_l) \mathcal{S}_l^{m_2}. \tag{45}$$

The key point on the analysis of the multigrid method is that the error can be split into two components. That is the one produced by the smoothing process and the one produced by the coarse grid correction. The coarse grid error consists of the low frequency components and the smoothing consists of the high frequency components of error. The ability to cope with the low frequency components is called the approximation property and with the second is called the smoothing property. For the analysis of the multigrid convergence [2, 29] used the framework based on the smoothing and approximation property. For analysis we

define the following norms, Euclidean norm by $\|\cdot\|$, and on $\mathcal{R}^{n_1+m_1}$ the following norm is applied:

$$\left\| \begin{pmatrix} \mathbf{u}_l \\ p_l \end{pmatrix} \right\|_h^2 := \|\mathbf{u}_l\|^2 + h_l^2 \|p_l\|^2 = \left\| \Theta_l \begin{pmatrix} \mathbf{u}_l \\ p_l \end{pmatrix} \right\|^2 \quad (46)$$

with $\Theta_l := \begin{pmatrix} I_{n_l} & O \\ O & I_{m_l} \end{pmatrix}$.

Furthermore we introduce

$$\widehat{\mathcal{M}}_l := \Theta_l^{-1} \mathcal{M}_l \Theta_l^{-1}, \quad \widehat{\mathcal{S}}_l := \Theta_l^{-1} \mathcal{S}_l \Theta_l^{-1}. \quad (47)$$

Using the norms defined above and taking $m_1 = m$ and $m_2 = 0$ above we obtain

$$\begin{aligned} \|L_l\|_h &= \left\| \Theta_l (\mathcal{M}_l^{-1} - P_l \mathcal{M}_{l-1}^{-1} R_l) \Theta_l \Theta_l^{-1} L_l \mathcal{S}_l^m \Theta_l^{-1} \right\| \\ &\leq \left\| \Theta_l (\mathcal{M}_l^{-1} - P_l \mathcal{M}_{l-1}^{-1} R_l) \Theta_l \right\| \left\| \widehat{\mathcal{M}}_l \widehat{\mathcal{S}}_l \right\|. \end{aligned} \quad (48)$$

The theorems below state the two properties and the multi-grid convergence. For detailed proof we refer to [12, 18].

Theorem 6 (approximation property). *Assume that Ω is such that the problem (5) is H^2 -regular. Let \mathcal{M}_l be the coefficient stiffness matrix and R_l and P_l the prolongation and the restriction operators. Then there exists a constant $C_{\mathcal{M}}$ independent of l and using h -scaling induced by \mathcal{M}_l then*

$$\left\| \Theta_l (\mathcal{M}_l^{-1} - P_l \mathcal{M}_{l-1}^{-1} R_l) \Theta_l \right\|_h \leq C_{\mathcal{M}} \|\mathcal{M}_l^{-1}\|_2, \quad (49)$$

where $C_{\mathcal{M}} = Ch^2$.

The smoothing property is dependent on the smoother used. It varies from one smoother to another. In this work we used the Braess-Sarazin in which we solve the system (37) exactly and sufficiently accurate inexact inner solver.

Theorem 7 (smoothing property). *Let \mathcal{M}_l be the coefficient stiffness matrix and the smoothing operator \mathcal{S}_l . Then*

$$\left\| \widehat{\mathcal{M}}_l \widehat{\mathcal{S}}_l^m \right\| \leq g(m) \|\mathcal{M}_l\|, \quad (50)$$

where $g(m) = ch_l^2/(m-1)$ for $m \geq 2$ and $g(m)$ is a decreasing function with $\lim_{m \rightarrow \infty} g(m) = 0$.

Combining the approximation property Theorem 6 with the smoothing property Theorem 7 produces a two-grid convergence result.

Theorem 8. *Assume that $m_2 = 0$ and that Ω is such that the problem (5) is H^2 -regular. Then for the two-grid method the following holds:*

$$\| \mathcal{M}_l \|_h \leq \frac{C_{\mathcal{M}}}{m-1}, \quad m \geq 2 \quad (51)$$

with a constant $C_{\mathcal{M}}$ independent of l and m .

Using this two-grid contraction number bound the multi-grid W -cycle method convergence results can be derived using ideas in [1, 2].

4. Numerical Results

In this section we present the numerical solution of classical Stokes problem (1)–(3) using the solver presented above. The solver is denoted by MGM (Algorithm 3). We present the results of this method as outlined above to run the traditional test problem, the driven cavity flow problem [11, 12, 27, 28]. It is a model of the flow in a square cavity (the domain is Ω_{\square}) with the top lid moving from left to right in our case the regularized cavity model $\{y = 1 : -1 \leq x \leq 1 \mid u_x = 1 - x^4\}$ [11]. The Dirichlet no-slip boundary condition is applied on the side and bottom boundaries. The mixed finite element method was used to discretize the cavity domain $\Omega = (-1, 1)^2$.

We pay particular attention to the computational performance of the multigrid method on the system (22) at different grid levels. We compare the effectiveness of different smoothing/relaxation methods in the performance of the multigrid method and different approximations for the preconditioners \widehat{A} and \widehat{S} of the smoothers. The following setup of the smoothers listed is considered.

- (i) Distributed Gauss Seidel (DGS) smoother: we use one Gauss Seidel iteration for the evaluations of \widehat{A} and one Gauss Seidel iteration for the computation of \widehat{A}_p . The method becomes DGSMG.
- (ii) Inexact Uzawa smoother (IUzawa): the two cases are considered for the evaluation of the preconditioners. Firstly, the approximation $\widehat{A} = \text{diag}(A)$ and one $\nu(1, 1)$ -cycle is used to approximate Schur complement matrix $B\widehat{A}B^T$. The second case is to use one $\nu(1, 1)$ -cycle for both evaluations of \widehat{A} and $B\widehat{A}B^T$. The method becomes IUZAWAMG.
- (iii) Braess-Sarazin smoother (B-S): the two cases are considered for the evaluation of the preconditioners. Firstly, the approximation $\widehat{A} = \text{diag}(A)$ and one $\nu(1, 1)$ -cycle is used to solve the approximate Schur complement matrix $B\widehat{A}B^T$. The second case is to use one $\nu(1, 1)$ -cycle for both evaluations of \widehat{A} and $B\widehat{A}B^T$. The method becomes B-SMG.
- (iv) PMINRES smoother: the first case is to use diagonal preconditioner for \widehat{A} and \widehat{S} and the second case is one $\nu(1, 1)$ -cycle for committing the inversion of the Laplacian operator for velocity as one $\nu(1, 1)$ cycle is used to approximate the Schur complement using the pressure mass matrix to accelerate the MINRES. The method becomes PMINRESMG.

The comparison is made on the performance of the multigrid schemes with different smoothers (i)–(iv) and cases involving different approximations of preconditioners in terms of iterative counts and CPU time. The numerical treatment is given to the discrete Stokes problem which resulted from the mixed finite Hood-Taylor stable elements consisting of biquadratic elements for the velocities and bilinear elements for the pressure, on a uniform grid. Implementation of our algorithms was performed on a Windows 7 platform with 2.13 GHz speed intel dual core processor by using MATLAB 7.14

TABLE 1: Refinement levels and number of nodes (n_v : number of velocity unknowns ($\times 2$) and m_p : number of pressure unknowns).

Refinement level (l)	1	2	3	4	5	6	7	8
Mesh size (h_l)	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{256}$
Velocity nodes (n_v)	9	25	81	289	1089	4425	16641	66049
Pressure nodes (m_p)	4	9	25	81	289	1089	4425	16641

TABLE 2: Number of iterations and CPU time for Braess-Sarazin (2diag(A), v -cycle(1, 1)) multigrid V -cycle at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- V -cycle			
	$v(1, 1)$ Iter(cpu time (sec))	$v(2, 2)$ Iter(cpu time (sec))	$v(3, 3)$ Iter(cpu time (sec))	$v(4, 4)$ Iter(cpu time (sec))
$\frac{1}{16}$	21 ($7.1e - 02$)	15 ($7.3e - 02$)	9 ($2.1e - 02$)	6 ($8.6e - 02$)
$\frac{1}{32}$	22 ($5.4e - 01$)	12 ($6.4e - 01$)	10 ($1.6e - 01$)	7 ($7.7e - 01$)
$\frac{1}{64}$	22 ($1.6e - 01$)	16 ($5.14e - 01$)	11 (1.6)	7 ($5.3e - 01$)
$\frac{1}{128}$	22 (1.64)	16 (2.58)	11 (2.17)	7 (4.45)

TABLE 3: Number of iterations and CPU time for Braess-Sarazin (2diag(A), $v(1, 1)$) multigrid W -cycle at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- W -cycle			
	$v(1, 1)$ Iter(cpu time (sec))	$v(2, 2)$ Iter(cpu time (sec))	$v(3, 3)$ Iter(cpu time (sec))	$v(4, 4)$ Iter(cpu time (sec))
$\frac{1}{16}$	16 ($8.3e - 02$)	11 ($8.4e - 02$)	7 ($3.5e - 02$)	5 ($9e - 02$)
$\frac{1}{32}$	17 ($6e - 01$)	12 ($6e - 01$)	7 ($1.2e - 01$)	5 ($7.1e - 01$)
$\frac{1}{64}$	17 ($5e - 01$)	13 ($5.1e - 01$)	8 (1.89)	6 ($5.6e - 01$)
$\frac{1}{128}$	16 (1.98)	13 (3.24)	8 (2.13)	6 (5.78)

programming language and the MATLAB built-in Minres functions are used for the smoother. For the discretization we start with a uniform square grid with $h_0 = 1/2$ and we apply regular refinements to this starting discretization to obtain the finest grid level. The discretized equations are solved using the multigrid iteration with the W -cycle and V -cycle and m_1 and m_2 being presmoothing and postsmoothing steps, respectively. The smoothers are determined by specifying approximations \hat{A} and \hat{S} as highlighted in (i)–(vi) and in all cases where one v -cycle inner multigrid iteration is used with n_1 and n_2 being Gauss Seidel iteration steps of the presmoothing and the postsmoothing, respectively.

In this work we use the structured mesh and regular refinements. The finite element matrices on the rectangular grids are assembled and the meshes are generated by the MATLAB IFISS toolbox [3] in a hierarchy of grids which are produced by successive regular refinements. We need to choose the coarse mesh (the starting mesh), the finest mesh which corresponds to the maximum level of refinement on which the final approximate solution is considered.

The assembled matrices are stored for each refinement level for the system (22). Table 1 shows an example of the refinement levels, we use the coarsest (starting) level to have 9 nodes for velocity and 4 nodes for pressure variables (level 1) but we start the computation at level 2.

Table 1 shows the refinement levels and the number of grid points (nodes) for each level.

The zero initial guess is chosen for all the tests. In all the tests the iterations are repeated until the tolerance $\|R_i\|/\|R_0\| < 10^{-6}$, where $R_i = \begin{pmatrix} f_i \\ g_i \end{pmatrix} - \begin{pmatrix} A_i & B_i^T \\ B_i & O \end{pmatrix} \begin{pmatrix} u_i \\ p_i \end{pmatrix}$ is satisfied. The schemes converge if the stopping criteria are satisfied. The results show the first case of the evaluation, the preconditioners of the smoothers \hat{A} and \hat{S} with $\hat{A} = 2 \text{diag}(A)$, and for all cases evaluation of \hat{S} by one v -cycle inner multigrid iteration with n_1 and n_2 being Gauss Seidel iteration of the presmoothing and postsmoothing steps, respectively.

Tables 2 and 3 show the number of iterations and computing time to demonstrate the effects of different V -cycle (1, 2, 3, and 4) and W -cycle (1, 2, 3, and 4) being

TABLE 4: Number of iterations and CPU time for multigrid (V -cycle) with different smoothers and smoother preconditioner approximations at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- V -cycle(3, 3)			
	DGS Iter(cpu time (sec))	IUzawa Iter(cpu time (sec))	Braess-Sarazin Iter(cpu time (sec))	PMINRES Iter(cpu time (sec))
$\frac{1}{16}$	18 ($8.6e - 02$)	13 ($6.3e - 02$)	9 ($2.1e - 02$)	14 ($7.1e - 02$)
$\frac{1}{32}$	19 ($7.4e - 01$)	14 ($2.0e - 01$)	10 ($1.6e - 01$)	16 ($6.5e - 01$)
$\frac{1}{64}$	19 (3.12)	14 (1.64)	11 (1.6)	16 (4.898)
$\frac{1}{128}$	19 (8.66)	14 (3.2)	11 (2.17)	16 (7.97)

TABLE 5: Number of iterations and CPU time for multigrid (W -cycle) with different smoothers and smoother preconditioner approximations at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- W -cycle(3, 3)			
	DGS Iter(cpu time (sec))	IUzawa Iter(cpu time (sec))	Braess-Sarazin Iter(cpu time (sec))	PMINRES Iter(cpu time (sec))
$\frac{1}{16}$	15 ($9.8e - 02$)	9 ($7.1e - 02$)	7 ($3.5e - 02$)	15 ($8.5e - 02$)
$\frac{1}{32}$	16 ($8.3e - 01$)	10 ($3.4e - 01$)	8 ($1.2e - 01$)	14 ($5.1e - 01$)
$\frac{1}{64}$	17 (5.76)	9 (2.56)	8 (1.89)	14 (6.43)
$\frac{1}{128}$	17 (9.87)	9 (4.32)	8 (2.13)	14 (8.55)

pre- and postsmoothing steps with Braess-Sarazin (B-S) smoother ($\text{diag}(A), \nu(1, 1)$). We compare the performance of the V -cycle and W -cycle multigrid iterations with various smoothing steps at different grid levels using one of the smoothers, Braess-Sarazin.

From Tables 2 and 3 we observe that the number of iterations decreases when the smoothing steps decrease and the CPU time increases as expected with the increase in smoothing steps.

Tables 4 and 5 show the numerical results obtained of the multigrid solver at different grid levels. The number of V -cycle and W -cycle multigrid iterations and CPU time are shown, respectively. All the results presented underline the efficiency of the multigrid solver to indefinite systems of equations. In both tables we present results of the four studied smoothers of the multigrid solver. In Tables 4 and 5 we choose the approximation of the smoother preconditioners as $\hat{A} = 2 \text{diag}(A)$ and $\hat{S} = \nu\text{-cycle}(1, 1)$ for Braess-Sarazin, IUzawa, and PMINRES. For the DGS we use the one Gauss Seidel for both A and A_p . We fix the number of smoothing steps to (3,3) for all the results in the tables.

Comparing the performance of the V -cycle and W -cycle multigrid solver, we observe that the smoothers have different effects on the performance of the multigrid solver. The multigrid solver is optimal and the iterations are bound for all the grid levels. In Tables 4 and 5 we compare all smoothers and we observe that the Braess-Sarazin smoother leads to faster convergence of the multigrid with fewer iterations and

less CPU ahead of other smoothers. The inexact Uzawa did not disappoint in relaxing the error but the DGS and the PMINRES lead to more iterations and computing times. The other observation in Tables 4 and 5 is that the W -cycle converges in fewer iterations than the V -cycle though it has more computing times for all smoothers.

In Tables 6 and 7 we use different approximations for the preconditioner of the smoothers of the multigrid V -cycle and W -cycle, respectively. In applying the preconditioners, we approximate the preconditioner \hat{A} of the Laplacian stiffness and sparse matrix A and \hat{S} by a geometric multigrid $\nu(1, 1)$ -cycle method (A_{mg}). The multigrid is a well-known fast solver for such systems. The multigrid solver is an inner iteration of the smoothers. The results in Tables 6 and 7 also show that the one iteration of the multigrid ν -cycle is a suitable approximation of the smoothers since the multigrid solver has improved iterations from the ones in Tables 4 and 5. In both tables the multigrid method is optimal in solving the indefinite systems and the number of iterations is bounded for all smoothers independent of the mesh size or grid level.

Table 8 shows the changes in the estimated a posteriori errors for regularized driven cavity flow using Q_2 - Q_1 approximation for the flow: using the strategy built in IFISS [3, 8] that, for every element error, the local error estimation is given by the combination of the energy norm of the velocity error and the L_2 norm of the divergence error; that is,

$$\eta_T^2 := \|\nabla \mathbf{e}_T\|_T^2 + \|R_T\|_T^2, \tag{52}$$

TABLE 6: Number of iterations and CPU time of iterations for multigrid (V -cycle) with different smoothers and using one ν -cycle multigrid preconditioner approximation (for both \widehat{A} and \widehat{S}) at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- V -cycle(3, 3)			
	DGS Iter(cpu time (sec))	IUzawa Iter(cpu time (sec))	Braess-Sarazin Iter(cpu time (sec))	PMINRES Iter(cpu time (sec))
$\frac{1}{16}$	14 ($4e - 01$)	10 ($3.4e - 02$)	6 ($1.3e - 02$)	13 ($4.4e - 02$)
$\frac{1}{32}$	15 ($2.1e - 01$)	10 ($2.3e - 01$)	7 ($1.5e - 01$)	14 ($5.4e - 01$)
$\frac{1}{64}$	16 (2.11)	11 (2.01)	8 (1.01)	14 ()
$\frac{1}{128}$	16 (4.56)	11 (3.67)	8 (2.02)	14 (9.01)

TABLE 7: Number of iterations and CPU time for multigrid (W -cycle) with different smoothers and using one ν -cycle multigrid preconditioner approximation (for both \widehat{A} and \widehat{S}) at different levels of refinement, tolerance = 10^{-6} .

Levels	MG- W -cycle(3, 3)			
	DGS Iter(cpu time (sec))	IUzawa Iter(cpu time (sec))	Braess-Sarazin Iter(cpu time (sec))	PMINRES Iter(cpu time (sec))
$\frac{1}{16}$	11 ($6.7e - 01$)	8 ($5.6e - 02$)	5 ($3.2e - 02$)	13 ($5.8e - 02$)
$\frac{1}{32}$	12 ($2.4e - 01$)	9 ($3.2e - 01$)	6 ($2.2e - 01$)	14 ($7.3 - 01$)
$\frac{1}{64}$	13 (4.62)	9 (4.31)	6 (1.01)	15 (3.21)
$\frac{1}{128}$	13 (7.009)	9 (3.35)	6 (2.98)	15 (6.23)

TABLE 8: Changes in the $\|\nabla \cdot \mathbf{u}\|_{\Omega}$ estimated velocity divergence error using multigrid V -cycle. η : the global error estimator using different smoothers from one level to the other.

Levels	$\ \nabla \cdot \mathbf{u}\ _{\Omega}$		η	
	IUzawa	Braess-Sarazin	IUzawa	Braess-Sarazin
$\frac{1}{16}$	$3.2e - 001$	$2.7e - 002$	$7.97e - 002$	$1.81e - 001$
$\frac{1}{32}$	$1.24e - 002$	$1.2e - 002$	$3.08e - 002$	$1.44e - 001$
$\frac{1}{64}$	$5.7e - 003$	$5.93e - 003$	$1.15e - 003$	$1.12e - 002$
$\frac{1}{128}$	$3.18e - 003$	$3.93e - 003$	$4.8e - 003$	$1.12e - 002$

where \mathbf{e}_T is the velocity error estimate and $R_T = \|\nabla \cdot \mathbf{u}\|_T$ and $\eta := (\sum_{T \in T_h} \eta_T^2)^{1/2}$ are the global error estimator, using different smoothers from one level to the other.

From Table 8 we note that the velocity divergence is clearly converging at a faster rate to $O(h^3)$, which means that the estimated global error η is increasingly dominated by the velocity error component as $h \rightarrow 0$.

Figure 1 shows the sample grid output at the levels $1/64$ and $1/128$, the sample velocity solution (exponential streamlines), and the pressure plot at the same level with the same smoother.

5. Conclusion

The purpose of this study was to explore the multigrid solver for the Stokes equations. We have introduced four smoother iterative methods for both multigrids V -cycle and W -cycle to solve the indefinite systems emanating for the mixed finite element discretization of the Stokes problem. We analyse the construction of the multigrid solver, construction of the smoothers, computation costs, and CPU time as an indicator of the performance of each smoother at all grid levels. Numerical experimental results are given for both V -cycle and W -cycle for the smoothers at different grid levels.

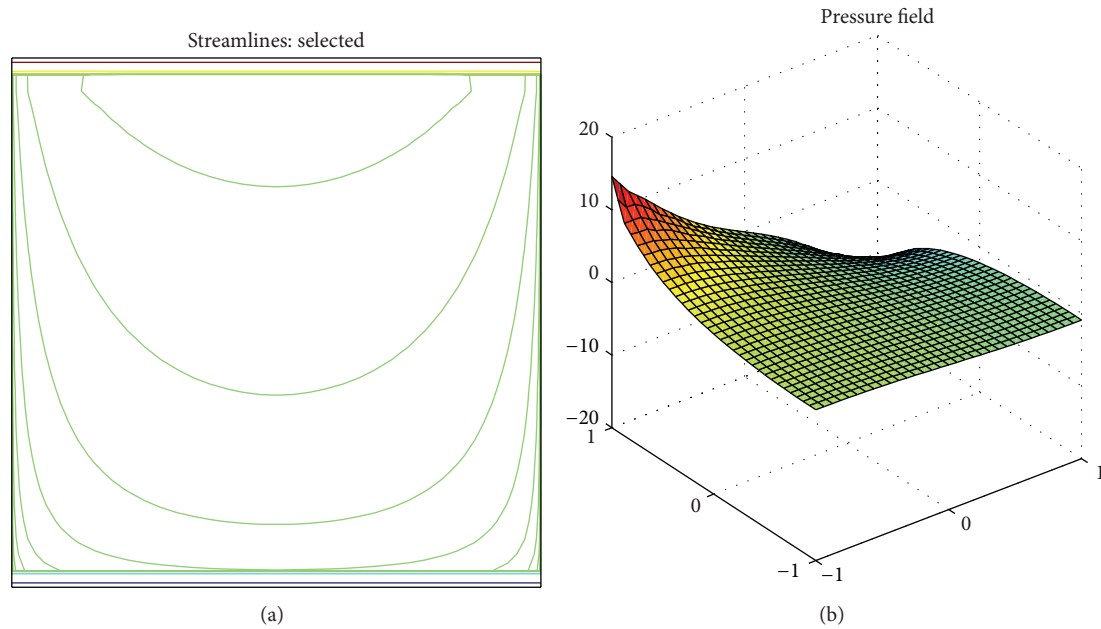


FIGURE 1: Velocity streamlines (a) and pressure plot (b) of the Stokes equation at level 5.

We have found out that for both cases and for all smoothers used in this study the multigrid solver is optimal and the number of iterations is bounded for all the grid levels. For the steady Stokes equations and the choices of the smoothers used the Braess-Sarazin like smoother became the best iteration to relax the error of the multigrid solver. All the numerical results show that the one v -cycle multigrid iteration is also a suitable preconditioner for the smoothers used.

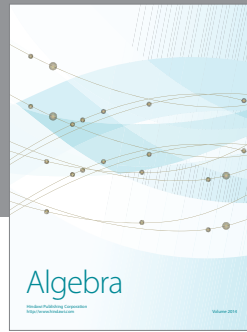
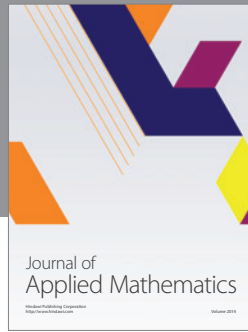
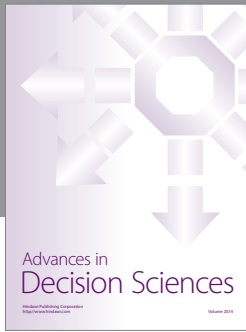
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] W. Hackbusch, *Multigrid Methods and Applications*, Springer, New York, NY, USA, 1985.
- [2] W. Hackbusch, "Fast solution of elliptic control problems," *Journal of Optimization Theory and Applications*, vol. 31, no. 4, pp. 565–581, 1980.
- [3] H. C. Elman, A. Ramage, and D. J. Silvester, "Algorithm 886: IFISS, a Matlab toolbox for modelling incompressible flow," *ACM Transactions on Mathematical Software*, vol. 33, no. 2, article 14, 18 pages, 2007.
- [4] A. Brandt, *Multigrid Techniques: Guide with Applications to Fluid Dynamics*, vol. 85 of GMD-studien, 1984.
- [5] W. Hackbusch, "On the fast solving of parabolic boundary control problems," *SIAM Journal on Control and Optimization*, vol. 17, no. 2, pp. 231–244, 1979.
- [6] U. Trottenberg and C. S. Oosterleer, *Multigrid*, Academic Press, London, UK, 2001.
- [7] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, New York, NY, USA, 2008.
- [8] H. C. Elman, D. J. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, UK, 2005.
- [9] A. Reusken, "Introduction to multigrid methods for elliptic boundary value problems," *Multiscale Simulation Methods in Molecular Sciences*, vol. 42, pp. 467–506, 2009.
- [10] H. C. Elman and G. H. Golub, "Inexact and preconditioned Uzawa algorithms for saddle point problems," *SIAM Journal on Numerical Analysis*, vol. 31, no. 6, pp. 1645–1661, 1994.
- [11] D. Silvester, H. C. Elman, D. Kay, and A. Wathen, "Efficient preconditioning of linearised Navier Stokes Equations," *Journal of Computational and Applied Mathematics*, 1999.
- [12] M. Larin and A. Reusken, "A comparative study of efficient iterative solvers for generalized Stokes equations," *Numerical Linear Algebra with Applications*, vol. 15, no. 1, pp. 13–34, 2008.
- [13] V. John, "A comparison of parallel solvers for the incompressible Navier-Stokes equations," *Computing and Visualization in Science*, vol. 1, no. 4, pp. 193–200, 1999.
- [14] V. John and L. Tobska, "Numerical performance of smoothers in coupled multigrid method for parallel solution of incompressible Stokes Equations," *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 453–473, 2000.
- [15] M. Pernice, "A hybrid multigrid method for the steady-state incompressible Navier-Stokes equations," *Electronic Transactions on Numerical Analysis*, vol. 10, pp. 74–91, 2000.
- [16] D. Braess, *Finite Elements*, Cambridge University Press, Cambridge, UK, 2007.
- [17] D. Braess and R. Sarazin, "An efficient smoother for the Stokes problem," *Applied Numerical Mathematics*, vol. 23, no. 1, pp. 3–19, 1997.
- [18] W. Zulehner, "A class of smoothers for saddle point problems," *Computing*, vol. 65, no. 3, pp. 227–246, 2000.
- [19] P. G. Ciarlet, *The Finite Element Methods for Elliptic Problems*, North-Holland Publishing Company, 1978.

- [20] M. S. Gockenbach, *Understanding and Implementing Finite Elements Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2006.
- [21] A. Elakkad, A. Elkhalfi, and N. Guessous, "A mixed finite element method for Navier-Stokes equations," *Journal of Applied Mathematics & Informatics*, vol. 28, no. 5-6, pp. 1331-1345, 2010.
- [22] V. Girault and P. Raviart, *A Finite Elements Approximation of the Navier-Stokes Equations*, Springer, Berlin, Germany, 1986.
- [23] M. Gunzburger, *Finite Element Method for Viscous Incompressible Flows*, Academic Press, New York, NY, USA, 1989.
- [24] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, 1991.
- [25] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*, John Wiley and Sons, West Sussex, UK, 2003.
- [26] C. H. Elman, "Multigrid and Krylov subspace methods for the discrete Stokes equations," UMIACS TR 74, 1994.
- [27] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, "Analysis of the inexact Uzawa algorithm for saddle point problems," *SIAM Journal on Numerical Analysis*, vol. 34, no. 3, pp. 1072-1092, 1997.
- [28] H. J. Bramble and J. E. Pasciak, "A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems," *Mathematics of Computation*, vol. 69, pp. 667-689, 1999.
- [29] W. Zulehner, "Analysis of iterative methods for saddle point problems: a unified approach," *Mathematics of Computation*, vol. 71, no. 238, pp. 479-505, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

