INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 20, Number 1, Pages 47–66

doi: 10.4208/ijnam2023-1003

OPTIMAL BLOCK PRECONDITIONER FOR AN EFFICIENT NUMERICAL SOLUTION OF THE ELLIPTIC OPTIMAL CONTROL PROBLEMS USING GMRES SOLVER

K. MUZHINJI*

Abstract. Optimal control problems are a class of optimisation problems with partial differential equations as constraints. These problems arise in many application areas of science and engineering. The finite element method was used to transform the optimal control problems of an elliptic partial differential equation into a system of linear equations of saddle point form. The main focus of this paper is to characterise and exploit the structure of the coefficient matrix of the saddle point system to build an efficient numerical process. These systems are of large dimension, block, sparse, indefinite and ill conditioned. The numerical solution of saddle point problems is a computational task since well known numerical schemes perform poorly if they are not properly preconditioned. The main task of this paper is to construct a preconditioner the mimic the structure of the system coefficient matrix to accelerate the convergence of the generalised minimal residual method. Explicit expression of the eigenvalue and eigenvectors for the preconditioned matrix are derived. The main outcome is to achieve optimal convergence results in a small number of iterations with respect to the decreasing mesh size h and the changes in δ the regularisation problem parameters. The numerical results demonstrate the effectiveness and performance of the proposed preconditioner compared to the other existing preconditioners and confirm theoretical results.

Key words. Partial differential equations (PDEs), PDE-optimal control problems, saddle point problem, block preconditioners, preconditioned generalised minimal residual method (PGMRES).

1. Introduction

Optimal control problems associated with partial differential equations arise in a variety application areas such as social, scientific, industrial, medical and engineering applications including optimal control, optimal design and parameter identification. In particular real life applications include flow control, reaction-diffusion problem of chemical processes, shape optimization, problems in financial markets and optimal pricing. In this paper we deal with the numerical solution of the distributed optimal control of elliptic equations that arise in real life applications in the optimal stationary heat. We consider the following elliptic distributed PDE-optimal control problem

$$\min_{(\mathbf{u},\mathbf{y})} J(\mathbf{y},\mathbf{u}) := \frac{1}{2} \parallel \mathbf{y}-\mathbf{y}_d \parallel^2_{L^2(\Omega)} + \frac{\delta}{2} \parallel \mathbf{u} \parallel^2_{L^2(\Omega)}, \tag{1}$$

subject to the constraints

$$\begin{aligned} -\Delta \mathbf{y} &= \mathbf{f}+\mathbf{u} \quad \text{in} \quad \Omega, \\ \mathbf{y} &= \mathbf{g} \quad \text{on} \quad \partial\Omega. \end{aligned} \tag{2}$$

where $\Omega \subset \mathcal{R}^2$ is the domain with boundary $\partial\Omega$. These problems were theoretically introduced by [11, 22] comprise of the objective function given by Equation (1) to

Received by the editors on March 14, 2022 and, accepted on August 21, 2022.
2000 *Mathematics Subject Classification.* 49J20, 65M22, 65M55, 65M60.
*Corresponding author. Department of Mathematical and Computational Sciences, University of Venda, P. B. X5050, Thohoyandou 0950, South Africa.

be minimised and the PDE-constraints given by Equations (2). Here we want to find $\mathbf{y}$ the state variable that satisfies the PDE-constraint as close to $\mathbf{y}_d$ as possible, the desired state which is known over the domain $\bar{\Omega}$ and $\mathbf{u}$ the control variable on the right hand side. This means that $\mathbf{y}$ is the solution of Equation (2) for a given control $\mathbf{u}$ either in the whole domain or on the boundary. The control functions that are either distributed (defined on Ω) or boundary (defined on $\partial\Omega$). If control functions are defined on $\partial\Omega$ we have boundary control problem for example the optimal temperature distribution otherwise we have a distributed control problem like the optimal heat source distributed on the whole domain. The temperature distribution or state $\mathbf{y}$ inside the domain is controlled by the enforcing heating source $\mathbf{u}$. For some practical purposes, we would like to choose the optimal control which minimizes the difference between the desired stationary temperature distribution $\mathbf{y}_d$ and the achievable temperature distribution $\mathbf{y}$. Mathematically, by assuming the boundary temperature vanishes. In this paper, we develop a new fast and efficient solver for the distributed optimal control problem Equations (1-2). The parameter δ is called the regularization parameter which measures the cost of the control and is supplied and positive. We refer to [3, 8, 9] on their numerical developments of such problems.

The optimal control problem has a unique solution $(\mathbf{y}, \mathbf{u})$ characterised by the optimality system called the Karush-Kuhn-Tucker (KKT) system [4, 20]. The first order optimality system of the PDE-optimal control problem Equations (1-2) consists state equation, adjoint equation and the control equation which is a saddle point problem as given below

$$
\begin{array}{llcllll}
(3) & -\Delta \mathbf{p} & = & \mathbf{y} - \mathbf{y}_d, & \text{in } \Omega & \mathbf{p} = \mathbf{0} \text{ on } \partial\Omega & \text{adjoint equation,} \\
(4) & -\Delta \mathbf{y} & = & \mathbf{f} + \mathbf{u}, & \text{in } \Omega & \mathbf{y} = \mathbf{g} \text{ on } \partial\Omega & \text{state equation,} \\
(5) & \delta \mathbf{u} - \mathbf{p} & = & 0 & \text{in } \Omega & & \text{control equation.}
\end{array}
$$

The optimality system is achieved through the Lagrange multiplier method which partitions the model problem into three equations namely in the state $\mathbf{y}$, control $\mathbf{u}$ and the adjoint, $\mathbf{p}$. For the numerical solution of the elliptic optimal control problem we apply the finite element method to the Equations (3 - 5) to get the linear saddle point problem. The finite element method is the most popular technique for the numerical solution of the PDE-constrained optimisation problems, see [9, 17, 18]. The finite element method results in the coupled linear algebraic system which has to be solved by the appropriate solvers. The resulting discrete KKT system is

$$
\mathcal{K}\mathbf{x} = \begin{pmatrix} M & O & K \\ O & \delta M & -M \\ K & -M & O \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} M\mathbf{y}_d \\ \mathbf{0} \\ \mathbf{d} \end{pmatrix} = \mathbf{b}, \tag{6}
$$

where $K \in \mathcal{R}^{n\times n}$ is a stiffness matrix and mass matrix $M \in \mathcal{R}^{n\times n}$ both symmetric and positive definite. Both K and M are sparse, hence $\mathcal{K}$ is sparse, symmetric and indefinite. The vector $\mathbf{d} \in \mathcal{R}^n$ contains the terms arising from the boundaries of the finite element of the state $\mathbf{y}$.

The linear algebraic system of Equation (6) is large scale, indefinite and has poor spectral properties such that well known Krylov subspace methods perform poorly [3, 18] and references therein. In recent years, the efficient solvers of the algebraic system that results from the optimal control problems has attracted a lot of attention and plenty of algorithms and preconditioners are proposed. The vital requirement for optimal performance of the Krylov subspace iterative methods is that the system matrix must have good spectral properties. This has preoccupied the computational scientific community for decades to transform and achieve good distribution of eigenvalues of the system coefficient matrix through preconditioning techniques and it is still an active area of research [1, 3, 12]. The previous and the recent work has been devoted to the development of an efficient numerical solution of a class of block 3×3 linear systems of Equation (6) and the reduced system [1, 3, 18]. The main task in this paper is to develop and analyse a new effective preconditioner for the algebraic system of Equation (6) and apply the GMRES solver for its solution process. We refer to [3, 12] for an explicit detailed review of block preconditioners. There are many different preconditioners that have been developed for the linear system of saddle point form that exploit their block structure and form. In this work we present the preconditioners that fall in this class with the main task of achieving convergence of the iterative scheme in the number of iterations that is independent of the discretization mesh size parameter h and optimal with respect to the problem regularisation parameter δ. That is the optimal performance of the iterative scheme entails convergence independent of the changing and decreasing both the mesh size and the regularisation parameter. To be successful in attaining this goal the classical preconditioning techniques are divided into three broad classes, the definite Hermitian preconditioners where it is possible to retrieve a cluster of eigenvalues [2, 15] where the MINRES can be applied, indefinite Hermitian preconditioners [19] and non Hermitian preconditioners [5, 10, 12, 24]. In this paper we focus on the last approach which benefit from the spectral distribution and the block form of the coefficient matrix. There are those preconditioners which follow the same structure of the coefficient matrix but based on the Schur complement form, we refer to [13, 15, 16, 17, 19, 21] and references therein. The second class of preconditioners is based on the block structure and does not need approximations where the GMRES is applied. For more of such preconditioners, specifically, those that were developed and analysed in [2, 10, 12, 24] that follow the block structure of the system coefficient matrix. The preconditioner we propose here belongs to this class and is built with the objective of achieving a performance of the preconditioned iterative solver that is independent of the number of iterations form the mesh size h and regularisation δ. The block entries of such preconditioners can be applied exactly and inexactly in the numerical solution process. For both classes of preconditioners we need the clustering of eigenvalues whose focus and benefit is the optimal performance of the preconditioned iterative solver independent of the decreasing parameters. The preconditioner proposed in this paper does not require the Schur complement approximation and this makes it easy to construct and less costly to apply. For our choice we provide a strong clustering of eigenvalues of the preconditioned system. The goal is to obtain and show that the eigenvalues are clustered while we obtain the solver convergence in the number of iteration independent from the regularisation parameter. The main contribution of the work is to develop a block preconditioner that mimic the structure of the coefficient matrix which result in a robust and an efficient numerical GMRES scheme whose convergence is h and δ independent.

This paper is organized as follows. In Section 2 we preview the existing classes of block preconditioners based on the Schur complement form and the block form of the coefficient matrix. In Section 3, we discuss our proposed new block preconditioner that are in the structure of the coefficient matrix of Equation (6) and give the explicit eigenvalue expressions of the preconditioned systems. In Section 4 we present the numerical experiments to demonstrate how well the new proposed preconditioner works and finally we draw the conclusion in Section 5.

2. Existing block preconditioners

In this section we outline the several existing preconditioners for the linear algebraic system of Equation (6). For this system, the following Schur complement based preconditioners were developed. In [17] the block diagonal preconditioner

$$\mathcal{P}_{BD}(S_1) = \begin{pmatrix} M & O & O \\ O & \delta M & O \\ O & O & S_1 \end{pmatrix},$$

was developed and applied with the MINRES solver where $S_1 = KM^{-1}K$ is the Schur complement approximation and in [15] the block diagonal preconditioner

$$\mathcal{P}_{BD}(S_2) := \begin{pmatrix} M & O & O \\ O & \delta M & O \\ O & O & S_2 \end{pmatrix},$$

and block triangular preconditioner

$$\mathcal{P}_{BT}(S_2) := \begin{pmatrix} M & O & O \\ O & \delta M & O \\ K & -M & S_2 \end{pmatrix},$$

with Schur complement approximation $S_2 = (K + \frac{1}{\sqrt{\delta}}M)M^{-1}(K + \frac{1}{\sqrt{\delta}}M)$. In [13] the block diagonal preconditioner was presented with the new approximation of the Schur complement approximation form $S_3 = (\sqrt{\delta}K + M)(\delta M)^{-1}(\sqrt{\delta}K + M)$

$$\mathcal{Q}_{S_3} := \begin{pmatrix} M & O & O \\ O & \delta M & O \\ O & O & S_3 \end{pmatrix}.$$

The preconditioners that follow the structure of the system matrix of the Equation (6) were developed in [5] and were used to precondition the GMRES method

$$\mathcal{P}_N := \begin{pmatrix} O & \delta K & O \\ O & \delta M & -M \\ K & -M & O \end{pmatrix},$$

and

$$\mathcal{P}_{BCT} := \begin{pmatrix} O & O & K \\ O & \delta M & -M \\ K & -M & O \end{pmatrix}.$$

Below we have the preconditioners developed for system involving the coefficient matrix of the form

$$\mathcal{K}_1 = \begin{pmatrix} 2\delta M & O & -M \\ O & M & K \\ -M & K & O \end{pmatrix}, \tag{7}$$

which emanates form the finite element discretization of the similar elliptic optimal control problem. For the system coefficient matrix in Equation (7) the following preconditioners were presented. In [18] the Schur complement based block triangular preconditioner was developed

$$\mathcal{P}_{BT} := \begin{pmatrix} 2\delta M & O & O \\ O & M & O \\ -M & K & KM^{-1}K \end{pmatrix}.$$

Bai in [2] developed the block counter diagonal preconditioner for the GMRES

$$\mathcal{P}_{BCD} := \begin{pmatrix} O & O & -M \\ O & M & O \\ -M & O & O \end{pmatrix},$$

and block tridiagonal preconditioner

$$\mathcal{P}_{BCT} := \begin{pmatrix} O & O & -M \\ O & M & K \\ -M & K & O \end{pmatrix},$$

The other pair of preconditioners, block symmetric and block lower triangular were developed in [24]

$$\mathcal{P}_{BS} := \begin{pmatrix} 2\delta M & O & -M \\ O & M & O \\ -M & O & O \end{pmatrix},$$

and

$$\mathcal{P}_{BLT} := \begin{pmatrix} 2\delta M & O & -M \\ O & M & O \\ -M & O & -\frac{1}{2\delta} M \end{pmatrix}.$$

Ke and Ma in [10] proposed the following four preconditioners

$$\mathcal{P}_1 := \begin{pmatrix} 2\delta M & O & -M \\ O & O & K \\ -M & K & O \end{pmatrix}, \qquad \mathcal{P}_2 := \begin{pmatrix} 2\delta M & O & -M \\ O & M & K \\ O & K & O \end{pmatrix},$$

$$\mathcal{P}_3 := \begin{pmatrix} 2\delta M & O & -M \\ O & M & O \\ -M & K & O \end{pmatrix}, \qquad \mathcal{P}_4 := \begin{pmatrix} 2\delta M & O & -M \\ O & M & K \\ M & O & O \end{pmatrix},$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ are well suited for $\delta \geq 10^{-6}$ while $\mathcal{P}_3$ and $\mathcal{P}_4$ are well suited for $\delta \leq 10^{-6}$ and recently Mirchi and Salkuyeh in [12] developed a new preconditioner

$$\mathcal{P} := \begin{pmatrix} O & K & O \\ O & M & K \\ -M & K & O \end{pmatrix}.$$

The main observation from the application of some preconditioners like $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{P}_N, \mathcal{P}_3, \mathcal{P}_4$ is that the computational performance of the numerical schemes is independent of the regularisation parameter δ. In some cases the numerical scheme fails to converge for the decreasing regularisation parameter and performs well for large values of the regularisation parameter. The computational performance of numerical schemes incorporated with the preconditioners like $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BCT}, \mathcal{P}_{BLT}, \mathcal{P}_1, \mathcal{P}_2$ strongly relies on the regularisation parameter. The trade in between the size of the regularisation parameter and the convergence of the iterative solvers inspired this work. In this paper we present efficient preconditioners which enhance the optimal performance of the iterative solvers. We demonstrate this by deriving the explicit eigenvalues expressions of the preconditioned system that clearly exhibit the clustering of the eigenvalues. We solve the algebraic system (6) and apply preconditioners designed for such systems.

3. Proposed preconditioner

In this section we present our proposed preconditioner $\mathcal{Q}_1$ and investigate the spectral distribution of the preconditioned system $\mathcal{Q}_1^{-1}\mathcal{K}$. We propose the following preconditioner

$$\mathcal{Q}_1 := \begin{pmatrix} M & O & K \\ O & O & -M \\ K & -M & O \end{pmatrix}, \tag{8}$$

for the system coefficient matrix of Equation (6). The main task here is to find the numerical solution of the system $\mathcal{Q}_1^{-1}\mathcal{K}\mathbf{x} = \mathcal{Q}_1^{-1}\mathbf{b}$ using the GMRES solver. We want to achieve good convergence properties for the GMRES iterative solver by clustering of most of the eigenvalues of $\mathcal{Q}_1^{-1}\mathcal{K}$ around a unit disc and away from zero [3].

We present the following results for preconditioner $\mathcal{Q}_1$. The mass matrix M and the stiffness matrix K in equation (6) from the $\mathbf{Q}_1$ finite element discretization of the equations (3-5) are symmetric and positive definite. This means that they are nonsingular and it follows that the preconditioner $\mathcal{Q}_1$ is also nonsingular

$$\mathcal{Q}_1^{-1} = \begin{pmatrix} M^{-1} & M^{-1}KM^{-1} & O \\ M^{-1}KM^{-1} & M^{-1}KM^{-1}KM^{-1} & -M^{-1} \\ O & -M^{-1} & O \end{pmatrix}. \tag{9}$$

Theorem 3.1. *Let the coefficient matrix $\mathcal{K}$ defined in (6) with matrices M and K nonsingular and the preconditioner $\mathcal{Q}_1$. Assume that λ is an eigenvalue of the preconditioned coefficient matrix $\mathcal{Q}_1^{-1}\mathcal{K}$ and $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T)^T \in \mathcal{C}^{3n}$ is the corresponding eigenvector. Then the eigenvalues of the preconditioned system matrix $\mathcal{Q}_1^{-1}\mathcal{K}$ are 1 of multiplicity $2n$ with corresponding eigenvectors of the form $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ 0 \\ \mathbf{x}_3 \end{pmatrix}$ for $\forall \mathbf{x}_1,\ \mathbf{x}_3 \neq 0$ and the remaining eigenvalues are of the form*

$$\lambda = 1 + \delta \frac{\mathbf{x}_1^T K M^{-1} K \mathbf{x}_1}{\mathbf{x}_1^T M \mathbf{x}_1}. \tag{10}$$

and the corresponding eigenvector $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ M^{-1}K\mathbf{x}_1 \\ K^{-1}M\mathbf{x}_1 \end{pmatrix}$, $\mathbf{x}_1 \neq 0$

Proof. *Let $(\lambda, \mathbf{x})$ be an eigenpair of the matrix $\mathcal{Q}_1^{-1}\mathcal{K}$ where $\mathbf{x} \neq 0$. The eigenvalue problem $\mathcal{K}\mathbf{x} = \lambda\mathcal{Q}_1\mathbf{x}$ is given by*

$$\begin{pmatrix} M & O & K \\ O & \delta M & -M \\ K & -M & O \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix} = \lambda \begin{pmatrix} M & O & K \\ O & O & -M \\ K & -M & O \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix}. \tag{11}$$

The equation 11 reduces to

$$(1-\lambda)M\mathbf{x}_1 = (\lambda - 1)K\mathbf{x}_3, \tag{12}$$

$$\delta M\mathbf{x}_2 = (1-\lambda)M\mathbf{x}_3, \tag{13}$$

$$(1-\lambda)K\mathbf{x}_1 = (1-\lambda)M\mathbf{x}_2. \tag{14}$$

From the above equations we note that if $\lambda = 1$ then equations (12) and (14) are always satisfied and equation (13) takes the form $\mathbf{x}_2 = \mathbf{0}$. This gives an eigenvector

$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0} \\ \mathbf{x}_3 \end{pmatrix}$, *with non zero vectors* $\mathbf{x}_1$ *and* $\mathbf{x}_3$ *corresponding to* $\lambda = 1$ *with algebraic multiplicity* $2n$.
Now assume that $\lambda \neq 1$, *then equations (12 - 14) reduce to*

$$\mathbf{x}_3 = -K^{-1}M\mathbf{x}_1, \tag{15}$$

$$\delta\mathbf{x}_2 = (1-\lambda)\mathbf{x}_3, \tag{16}$$

$$\mathbf{x}_2 = M^{-1}K\mathbf{x}_1. \tag{17}$$

This entails that $\mathbf{x}_1 \neq \mathbf{0}$ *because from equations (15) and (17) if we have* $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_3 = \mathbf{0}$ *which gives* $\mathbf{x} = \mathbf{0}$ *which is a contradiction for an eigenvector. Substituting* $\mathbf{x}_2$ *and* $\mathbf{x}_3$ *from equations (15 and 17) into equation (16) we get*

$$\delta KM^{-1}K\mathbf{x}_1 + M\mathbf{x}_1 = \lambda M\mathbf{x}_1. \tag{18}$$

Multiplying by $\mathbf{x}_1^T$ *and dividing by* $\mathbf{x}_1^T M\mathbf{x}_1 \neq 0$ *both sides gives*

$$\delta\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1 + \mathbf{x}_1^T M\mathbf{x}_1 = \lambda\mathbf{x}_1^T M\mathbf{x}_1, \tag{19}$$

$$\lambda = 1 + \delta\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{20}$$

with the corresponding eigenvector
$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ M^{-1}K\mathbf{x}_1 \\ -K^{-1}M\mathbf{x}_1 \end{pmatrix}$ *with* $\mathbf{x}_1 \neq 0$.

We now give the clear eigenvalue bounds which are dependent on PDE problem and the finite element used. In this paper we used the bilinear $\mathbf{Q}_1$ finite element approximations to discretise the system of Equations (3-5). We use the following results of Theorems (3.4) and (3.5) in [17] and which were also used by [10, 12] which are frequently used to derive the eigenvalue bounds of the preconditioned system.

Theorem 3.2. [17] *For the problem Equation (6) in* $\Omega \in \mathcal{R}^2$ *with the degree of approximation* $\mathbf{Q_m}$ *or* $\mathbf{P_m}$ *with* $\mathbf{m} \geq 1$ *the following bounds hold*

$$\alpha_1(m)h^2 \leq \frac{\mathbf{v}^T M\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq \alpha_2(m), h^2 \tag{21}$$

where α_1 *and* α_2 *are real constants independent of* h *and* δ *but dependent on* $\mathbf{m}$.

Theorem 3.3. [17] *For the problem Equation (6) in* $\Omega \in \mathcal{R}^2$ *with the degree of approximation* $\mathbf{Q_m}$ *or* $\mathbf{P_m}$ *with* $\mathbf{m} \geq 1$ *the following bounds hold*

$$\theta_1(m)h^2 \leq \frac{\mathbf{v}^T K\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq \theta_2(m), \tag{22}$$

where θ_1 *and* θ_2 *are real constants independent of* h *and* δ *but dependent on* $\mathbf{m}$.

Here m=1. Since the mass matrix M and the stiffness matrix K are symmetric and positive definite then from Equation (21) we have the following equations also used in [10, 12, 17]

$$\frac{1}{\alpha_2 h^2} \leq \frac{\mathbf{v}^T\mathbf{v}}{\mathbf{v}^T M\mathbf{v}} \leq \frac{1}{\alpha_1 h^2}, \tag{23}$$

$$\frac{1}{\alpha_2 h^2} \leq \frac{\mathbf{v}^T M^{-1}\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq \frac{1}{\alpha_1 h^2}, \tag{24}$$

$$\alpha_1 h^2 \leq \frac{\mathbf{v}^T\mathbf{v}}{\mathbf{v}^T M^{-1}\mathbf{v}} \leq \alpha_2 h^2, \tag{25}$$

and from (22) we have

$$\frac{1}{\theta_2} \leq \frac{\mathbf{v}^T\mathbf{v}}{\mathbf{v}^T K\mathbf{v}} \leq \frac{1}{\theta_1 h^2}, \tag{26}$$

$$\frac{1}{\theta_2} \leq \frac{\mathbf{v}^T K^{-1}\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq \frac{1}{\theta_1 h^2}, \tag{27}$$

$$\theta_1 h^2 \leq \frac{\mathbf{v}^T\mathbf{v}}{\mathbf{v}^T K^{-1}\mathbf{v}} \leq \theta_2. \tag{28}$$

With the above techniques, we obtain the following theorem.

Theorem 3.4. [10, 12, 24] *For $\mathbf{Q_1}$ approximation, let the preconditioner $\mathcal{Q}_1$ and assume that λ is an eigenvalue of the preconditioned system matrix $\mathcal{Q}_1^{-1}\mathcal{K}$. Then $\lambda = 1$ with multiplicity 2n or λ satisfies the following bound*

$$\lambda = 1 + \delta\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{29}$$

Proof. *Let λ be an eigenvalue of $\mathcal{Q}_1^{-1}\mathcal{K}$ corresponding to the eigenvector $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T)^T$. From Theorem 3.1 we have $\lambda = 1$ with algebraic multiplicity 2n and the remaining n eigenvalues satisfy*

$$\delta KM^{-1}Kx_1 + Mx_1 = \lambda Mx_1. \tag{30}$$

Since K and M are symmetric and positive definite and $\mathbf{x}_1 \neq 0$, rearranging equation (30) we get

$$\lambda = 1 + \delta\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{31}$$

Further we consider the expression $\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}$

$$\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T\mathbf{x}_1} = \frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T\mathbf{x}_1} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{32}$$

Let $\mathbf{y} = K\mathbf{x}_1$ then $\mathbf{y}^T = \mathbf{x}_1^T K$ and $\mathbf{x}_1 = K^{-1}\mathbf{y}$ then $\mathbf{x}_1^T = \mathbf{y}^T K^{-1}$. Substituting in equation (32)

$$\frac{\mathbf{x}_1^T KM^{-1}K\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1} = \frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T K^{-2}\mathbf{y}} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1 M\mathbf{x}_1}, \tag{33}$$

$$= \frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \cdot \frac{\mathbf{y}^T\mathbf{y}}{\mathbf{y}^T K^{-2}\mathbf{y}} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{34}$$

Let $\mathbf{z}_1 = K^{\frac{-1}{2}}\mathbf{y}$ and $\mathbf{z}_1^T = \mathbf{y}^T K^{\frac{-1}{2}}, \mathbf{y} = K^{\frac{1}{2}}\mathbf{z}_1$ and $\mathbf{y}^T = \mathbf{z}_1^T K^{\frac{1}{2}}$ and substituting in equation (34)

$$\frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \cdot \frac{\mathbf{y}^T\mathbf{y}}{\mathbf{y}^T K^{-2}\mathbf{y}} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1} = \frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \cdot \frac{\mathbf{z}_1^T K^{\frac{1}{2}}K^{\frac{1}{2}}\mathbf{z}_1}{\mathbf{z}_1^T K^{\frac{1}{2}}K^{-2}K^{\frac{1}{2}}\mathbf{z}_1} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}, \tag{35}$$

$$= \frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \cdot \frac{\mathbf{z}_1^T K\mathbf{z}_1}{\mathbf{z}_1^T K^{-1}\mathbf{z}_1} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}, \tag{36}$$

$$= \frac{\mathbf{y}^T M^{-1}\mathbf{y}}{\mathbf{y}^T\mathbf{y}} \cdot \frac{\mathbf{z}_1^T K\mathbf{z}_1}{\mathbf{z}_1^T\mathbf{z}_1} \cdot \frac{\mathbf{z}_1^T\mathbf{z}_1}{\mathbf{z}_1^T K^{-1}\mathbf{z}_1} \cdot \frac{\mathbf{x}_1^T\mathbf{x}_1}{\mathbf{x}_1^T M\mathbf{x}_1}. \tag{37}$$

Using the equations (21-22) and (23-28) we get

$$\frac{\theta_1^2}{\alpha_2} \le \frac{\mathbf{x}_1^T K M^{-1} K \mathbf{x}_1}{\mathbf{x}_1^T M \mathbf{x}_1} \le \frac{\theta_2^2}{\alpha_1^2 h^4}. \tag{38}$$

Hence the result

$$1 + \delta \frac{\theta_1^2}{\alpha_2} \le \lambda \le 1 + \delta \frac{\theta_2^2}{\alpha_1^2 h^4}. \tag{39}$$

This completes the proof.

Remark 3.5. *The theoretical results for Theorems 3.4 show that if the regularization parameter δ is small the eigenvalues of the preconditioned matrices $\mathcal{Q}_1^{-1}\mathcal{K}$ are clustered around 1..*

Algorithm 3.6. ***Application of Preconditioner*** $\mathcal{Q}_1$
Let $\mathbf{w} = (\mathbf{w}_1^T, \mathbf{w}_2^T, \mathbf{w}_3^T)^T$ be any given vector, we can compute the residual vector $\mathbf{v} = (\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T)^T$ and $\mathcal{Q}_1\mathbf{v} = \mathbf{w}$ using the following procedures:

(1) $M\mathbf{v}_3 = -\mathbf{w}_2$ *solve for* $\mathbf{v}_3$.
(2) $M\mathbf{v}_1 = \mathbf{w}_1 - K\mathbf{v}_3$ *solve for* $\mathbf{v}_1$.
(3) $M\mathbf{v}_2 = K\mathbf{v}_1 - \mathbf{w}_3$ *solve for* $\mathbf{v}_2$.

Remark 3.7. *We note form the Algorithm 3.6 that the construction of the preconditioner $\mathcal{Q}_1$ is simplified since it does not require approximation of the possible Schur complements of the coefficient matrix. Moreover we have shown that this preconditioner provides a strong cluster around 1 for the eigenvalues of the preconditioned coefficient matrix independent from the regularisation parameter δ. We consider the numerical experiments with the following preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2), \mathcal{Q}_{S_3}$, $\mathcal{P}_{BCT}, \mathcal{P}_N$ and compare their effectiveness in accelerating of the right preconditioned GMRES with our new proposed preconditioner $\mathcal{Q}_1$. Below are the subsystems that are involved in their application.*

Algorithm 3.8. ***Application of Preconditioners*** $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2), \mathcal{P}_{BCT}, \mathcal{P}_N$, $\mathcal{Q}_{S_3}$.
Let $\mathbf{w} = (\mathbf{w}_1^T, \mathbf{w}_2^T, \mathbf{w}_3^T)^T$ be any given vector, we can compute the residual vector $\mathbf{v} = (\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T)^T$ and $\mathcal{P}\mathbf{v} = \mathbf{w}$ using the following procedures:

(1) $\mathcal{P}_{BD}(S_1)\mathbf{v} = \mathbf{w}$.
  (a) $M\mathbf{v}_1 = \mathbf{w}_1$.
  (b) $\delta M\mathbf{v}_2 = \mathbf{w}_2$.
  (c) $KM^{-1}M\mathbf{v}_3 = \mathbf{w}_3$.
    (i) $K\mathbf{u} = \mathbf{w}_3$.
    (ii) $K\mathbf{v}_3 = M\mathbf{u}$.
(2) $\mathcal{P}_{BCT}\mathbf{v} = \mathbf{w}$.
  (a) $K\mathbf{v}_1 = \mathbf{w}_1$.
  (b) $\delta M\mathbf{v}_2 = Kv_1 - w_3$.
  (c) $M\mathbf{v}_3 = \delta M\mathbf{v}_2 - \mathbf{w}_2$.
(3) $\mathcal{P}_N\mathbf{v} = \mathbf{w}$.
  (a) $\delta K\mathbf{v}_2 = \mathbf{w}_1$.
  (b) $M\mathbf{v}_3 = \delta M\mathbf{v}_2 - \mathbf{w}_2$.
  (c) $K\mathbf{v}_1 = \mathbf{w}_3 + M\mathbf{w}_2$.
(4) $\mathcal{P}_{BD}(S_2)\mathbf{v} = \mathbf{w}$.
  (a) $M\mathbf{v}_1 = \mathbf{w}_1$.
  (b) $\delta M\mathbf{v}_2 = \mathbf{w}_2$.
  (c) $(K + \frac{1}{\sqrt{\delta}}M)M^{-1}(K + \frac{1}{\sqrt{\delta}}M)\mathbf{v}_3 = \mathbf{w}_3$.
    (i) $(K + \frac{1}{\sqrt{\delta}}M)\mathbf{u} = \mathbf{w}_3$.
    (ii) $(K + \frac{1}{\sqrt{\delta}}M)\mathbf{v}_3 = M\mathbf{u}$.
(5) $\mathcal{Q}_{S_3}\mathbf{v} = \mathbf{w}$.
  (a) $M\mathbf{v}_1 = \mathbf{w}_1$.
  (b) $\delta M\mathbf{v}_2 = \mathbf{w}_2$.
  (c) $(\sqrt{\delta}K + M)(\delta M)^{-1}(\sqrt{\delta}K + M)\mathbf{v}_3 = \mathbf{w}_3$.
    (i) $(\sqrt{\delta}K + M)\mathbf{u} = \mathbf{w}_3$.
    (ii) $(\sqrt{\delta}K + M)\mathbf{v}_3 = \delta M\mathbf{u}$.

Remark 3.9. *In the application of the preconditioners we propose further the approximation of the mass matrix M with $\tilde{M}$ and K with $\tilde{K}$. For example the approximation of our proposed preconditioner becomes*

$$\tilde{\mathcal{Q}}_1 := \begin{pmatrix} \tilde{M} & O & \tilde{K} \\ O & O & -\tilde{M} \\ \tilde{K} & -\tilde{M} & O \end{pmatrix}.$$

When applying the exact preconditioner for GMRES we approximate $\tilde{M}$ and $\tilde{K}$ by Cholesky factorisation and for the inexact preconditioner for the GMRES we approximate $\tilde{M}$ by 10 Chebyshev semi iterations and $\tilde{K}$ can be produced by 2 algebraic multigrid (AMG) cycles with 2 smoothing steps [2, 15, 17, 24].

Theorem 3.1 above clearly shows that the decreasing mesh size h and the regularisation parameter δ leads to the clustering of eigenvalues of the preconditioned coefficient matrix $\mathcal{Q}_1^{-1}\mathcal{K}$ around 1. The iterative solver is expected to converge with changes in the problem parameter and discretization parameter. In the next section we carry out numerical tests to verify the theoretical findings and compare the performance of the GMRES solver with the block preconditioners discussed above.

4. Numerical results

In this section we test the application of some of the preconditioners outlined in Section 2 that were formulated to accelerate the GMRES iterative solvers applied the saddle point coefficient system of Equation (6) and our proposed preconditioner analysed in Section 3. To achieve this, we consider the following problem

$$\min_{(\mathbf{y},\mathbf{u})} J(\mathbf{y},\mathbf{u}) := \frac{1}{2} \parallel \mathbf{y} - \mathbf{y}_d \parallel^2_{L^2(\Omega)} + \frac{\delta}{2} \parallel u \parallel^2_{L^2(\Omega)}, \tag{40}$$

subject to the constraints

$$\begin{array}{rcccc} -\Delta\mathbf{y} & = & \mathbf{u} & \text{in} & \Omega, \\ \mathbf{y} & = & \mathbf{0} & \text{on} & \partial\Omega. \end{array} \tag{41}$$

where $\Omega = (0,1)\times(0,1)$, $\mathbf{y}_d \in L^2(\Omega)$. Our methods are illustrated on the distributed control problems, which is

Problem 4.1. $\mathbf{y}_d = \sin(\pi x_1)\sin(\pi x_2)$ *defined on the domain Ω. For more information, we refer to Example 5.1.3 in [7] and Example 2 in [10].*

and [2] and Example 5.1 in [17, 18] with the following information

Problem 4.2. $\mathbf{y}_d = \begin{cases} (2x-1)^2(2y-1)^2 & \textit{if } (x,y) \in [0,\frac{1}{2}]^2 \\ 0 & \textit{if elsewhere} \end{cases}$

The results of the numerical experiments are presented to demonstrate the effectiveness of our proposed new preconditioner for the elliptic PDE-constrained optimisation problem. All numerical simulations and implementations were performed on a Windows 10 platform with Intel(R)Core(TM)i5-3230M CPU @2.6 GHz 6.00 GB speed intel(R) using Matlab 7 programming language. We use the IFISS matlab package developed in [6] to generate a discrete block linear algebraic system. The block matrix entries M, K and system coefficient matrix $\mathcal{K}$ sizes are shown in Table 1 below

We compute the solution using the right preconditioned restarted GMRES solver with a zero vector as initial guess to achieve a tolerance of 10^{-6} on the residual to

TABLE 1. Mesh sizes and corresponding sizes of the matrices.

mesh($\boldsymbol{h}$)	size of $\boldsymbol{M}$	size of $\boldsymbol{K}$	size of $\boldsymbol{\mathcal{K}}$
2^{-3}	81×81	81×81	243×243
2^{-4}	289×289	289×289	867×867
2^{-5}	1089×1089	1089×1089	3267×3267
2^{-6}	4225×4225	4225×4225	12675×12675
2^{-7}	16641×16641	16641×16641	49923×49923
2^{-8}	66049×66049	66049×66049	198147×198147

measure the number iterations and the CPU time in seconds. The goal is to check the effectiveness of the preconditioned GMRES numerical scheme with the changes in mesh size and problem regularisation parameter, that is to achieve parameter independent convergence. Table 2 below gives the costs for different preconditioners during the preconditioning process at each iteration step for computing the residual vector. We get the solution by solving the preconditioner subsystems in the

TABLE 2. The costs of different preconditioners.

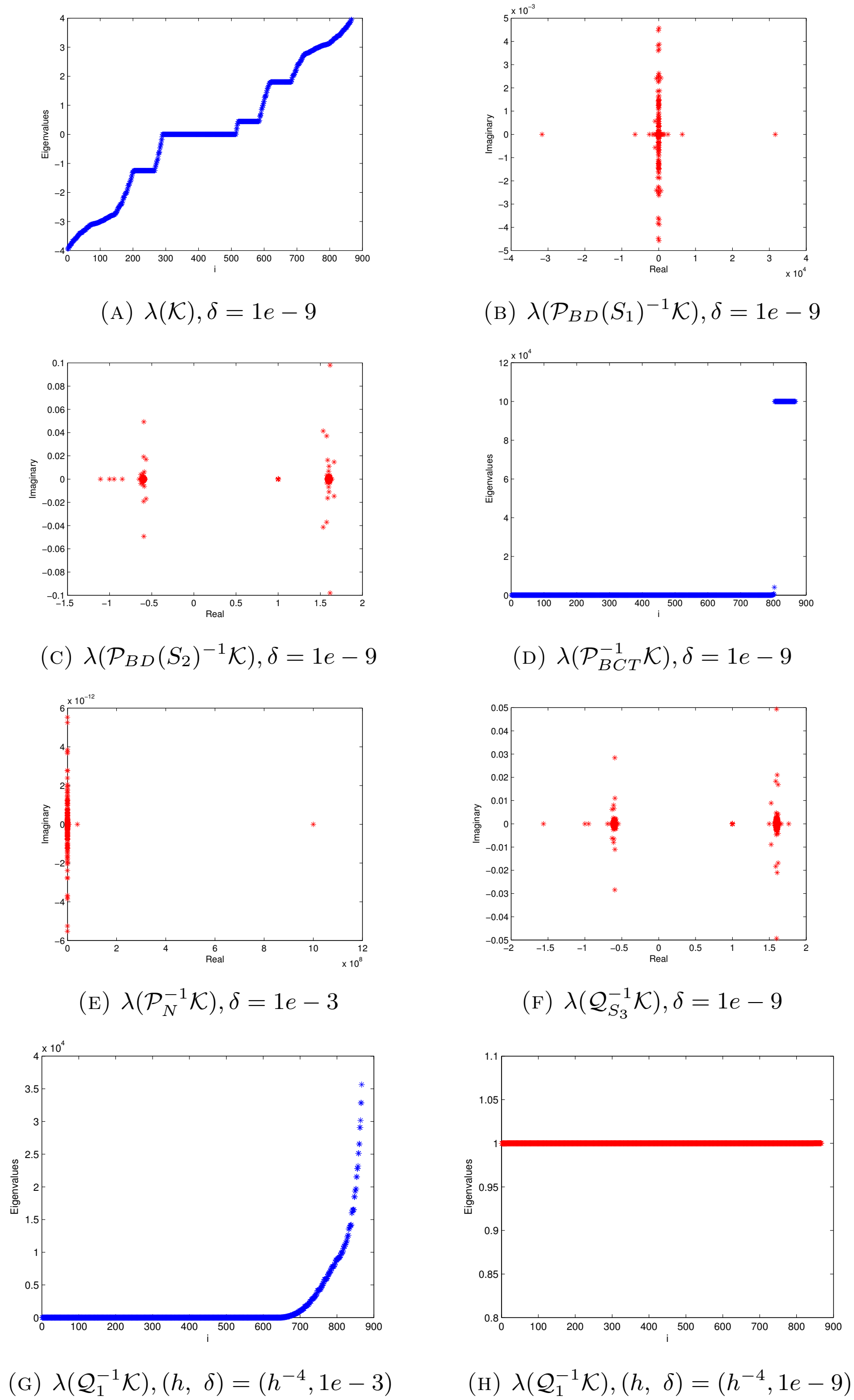
	$\boldsymbol{\mathcal{P}_{BD}(S_1)}$	$\boldsymbol{\mathcal{P}_{BD}(S_2)}$	$\mathbf{Q}_{S_3}$	$\boldsymbol{\mathcal{P}_{BCT}}$	$\boldsymbol{\mathcal{P}_{N}}$	$\boldsymbol{\mathcal{Q}_1}$
Matrix vector multiplications	1	1	1	2	2	2
Number of subsystems	4	4	4	3	3	3

Algorithms (3.6 and 3.8) in Section 3. Table 2 gives the cost of the application of each preconditioner in the form of the subsystems and matrix vector multiplication. The application of the preconditioners $\mathcal{P}_{BD}(S_1)$, $\mathcal{P}_{BD}(S_2)$ and $\mathcal{Q}_{S_3}$ are involved in the solution of 4 subsystems and the other preconditioners involve the solution of 3 subsystems in accelerating the GMRES solver.

We first display the distribution of eigenvalues of the coefficient matrix and preconditioned coefficient matrix with different preconditioners for $(h,\ \delta) = (h^{-4}, 1e - 9)$ and $(h^{-4}, 1e - 1)$ for $\mathcal{Q}_1$. To do this, we display (i, λ_i) for $i = 1, 2, 3, ..., n$, where λ_i and for complex (imaginary, real) ordered eigenvalues for these matrices in Figure 1.

Figure 1 presents the eigenvalue distribution of $\mathcal{K}$, $P_{BD}(S_1)^{-1}\mathcal{K}$, $\mathcal{P}_{BD}(S_2)^{-1}\mathcal{K}$, $\mathcal{P}_{BCT}^{-1}\mathcal{K}$, $\mathcal{P}_N^{-1}\mathcal{K}$, $\mathcal{Q}_1^{-1}\mathcal{K}$ and $\mathcal{Q}_{S_3}^{-1}\mathcal{K}$ for regularisation parameter $\delta = 1e - 9$ at $h = 2^{-4}$. For $\delta = 1e - 3$, the preconditioned matrix $\mathcal{Q}_1^{-1}\mathcal{K}$ has eigenvalues outside the spectrum. The most important observation is that the eigenvalue distribution of $P_D(S_2)^{-1}\mathcal{K}$, $\mathcal{Q}_{S_3}^{-1}\mathcal{K}$ and $\mathcal{Q}_1^{-1}\mathcal{K}$ are clustered around 1 for smaller regularisation parameter. This agrees with the theoretical findings in [13, 15] and the Theorem 3.1 for the eigenvalues of our proposed preconditioner. It is expected that for the decreasing regularisation parameter the inexact preconditioned GMRES by the preconditioners performs extremely well. The preconditioned GMRES will perform well for large values of δ with other preconditioners since there is no clustering of eigenvalues for small values of δ. The eigenvalue distribution of $\mathcal{K}$ and for the preconditioned systems $\mathcal{P}_{BD}(S_1)^{-1}\mathcal{K}$, $\mathcal{P}_{BCT}^{-1}\mathcal{K}$ and $P_N^{-1}\mathcal{K}$ we note that most of the eigenvalues are not clustered around 1 for smaller regularization parameter. This means that the inexact GMRES with these preconditioners will not perform well.

We compare the numerical results of the proposed preconditioner $\mathcal{Q}_1$ with the those presented in Figure 1 for the Problems 4.1 and 4.2. The preconditioners in Figure 1 have been considered because they were developed for that coefficient



(A) $\lambda(\mathcal{K}), \delta = 1e-9$

(B) $\lambda(\mathcal{P}_{BD}(S_1)^{-1}\mathcal{K}), \delta = 1e-9$

(C) $\lambda(\mathcal{P}_{BD}(S_2)^{-1}\mathcal{K}), \delta = 1e-9$

(D) $\lambda(\mathcal{P}_{BCT}^{-1}\mathcal{K}), \delta = 1e-9$

(E) $\lambda(\mathcal{P}_{N}^{-1}\mathcal{K}), \delta = 1e-3$

(F) $\lambda(\mathcal{Q}_{S_3}^{-1}\mathcal{K}), \delta = 1e-9$

(G) $\lambda(\mathcal{Q}_{1}^{-1}\mathcal{K}), (h,\ \delta) = (h^{-4}, 1e-3)$

(H) $\lambda(\mathcal{Q}_{1}^{-1}\mathcal{K}), (h,\ \delta) = (h^{-4}, 1e-9)$

FIGURE 1. Eigenvalues distribution of the coefficient matrix and preconditioned coefficient matrix with different preconditioners, $\delta = 1e-9$ with $h = 2^{-4}$.

TABLE 3. Number of iterations taken by MINRES solver with the block diagonal preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2), \mathcal{Q}_{S_3}$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

h	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$				$\mathcal{Q}_{S_3}$			
	δ				δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
2^{-4}	9	23	35	35	13	5	3	3	13	5	3	3
2^{-5}	11	29	71	99	13	9	3	3	13	9	3	3
2^{-6}	11	33	111	101	13	11	5	3	13	10	5	3
2^{-7}	13	33	133	107	15	11	7	5	15	10	5	3
2^{-8}	17	33	139	103	15	11	7	5	15	10	5	3

TABLE 4. CPU times(sec) taken by MINRES solver with the block diagonal preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2), \mathcal{Q}_{S_3}$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

h	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$				$\mathcal{Q}_{S_3}$			
	δ				δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
2^{-4}	0.14	0.266	0.297	0.166	0.188	0.109	0.078	0.078	0.081	0.063	0.078	0.047
2^{-5}	0.283	0.581	1.317	1.457	0.312	0.25	0.141	0.156	0.297	0.325	0.11	0.109
2^{-6}	0.594	1.297	4.168	3.579	0.625	0.530	0.500	0.556	0.510	0.381	0.272	0.532
2^{-7}	2.169	5.171	17.27	15.59	2.54	2.00	1.48	1.64	2.42	2.05	1.45	1.531
2^{-8}	10.58	21.79	78.92	60.50	10.59	8.43	5.97	5.54	8.64	7.47	4.95	4.02

matric Equation 6. We start by extracting the results from [13] where the MINRES solver applied with the block diagonal preconditioners based on the Schur complement forms $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2), \mathcal{Q}_{S_3}$ for comparison purposes.

We now give the numerical experiment results from the GMRES iterative solver preconditioned with the block preconditioners $\mathcal{P}_{BD}(S_1)$, $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{P}_{BCT}$, $\mathcal{P}_N$, $\mathcal{Q}_1$ for problems 4.1 and 4.2 to demonstrate that our proposed preconditioner $\mathcal{Q}_1$ is applicable, competitive, robust and cost effective. We concentrate on the performance in terms of the number of iterations and the CPU time in seconds for comparison purposes. In a sequel we present two sets of numerical experiment results. We first present the set of results of the numerical experiments obtained from using the preconditioned GMRES solver for solving Equation (6) with inexact approximations of the preconditioners. In this case all the subsystems are solved using a fixed 10 Chebyshev semi iterations [13, 14, 17, 23] for the mass matrix M and two algebraic multigrid iterations with two pre-and post-smoothing steps of the Jacobi method for the matrix involving K.

In Tables (5, 6, 7, 8, 9, 10), we report the number of iterations and computing times (in braces) with respect to preconditioners $\mathcal{P}_{BD}(S_1)$, $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{P}_{BCT}$, $\mathcal{P}_N$, $\mathcal{Q}_1$ employed to precondition the GMRES solvers for the system 6. The subsystems associated with the application of preconditioners are solved by iterative solvers. In Tables (5, 6, 8, 9), we observe that the computing efficiency of the preconditioned iterative solvers associated with $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BCT}, \mathcal{P}_N$ are dependent on the regularisation parameter. The dash (-) means the solver did not converge for the maximum number of iterations set at 1000. It is clear from the numerical results that for large values of δ the preconditioners are efficient and for decreasing δ they are not efficient. We also observe in Tables (7, 10) that the preconditioners $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{Q}_1$ are very efficient and their application in the numerical process with the GMRES resulted in both mesh and regularisation parameter independent convergence. The inexact GMRES with the preconditioners $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{Q}_1$

TABLE 5. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BCT}, \mathcal{P}_N$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{P}_{BCT}$				$\mathcal{P}_N$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	7(0.141)	7(0.188)	8(0.125)	8(0.125)	4(0.093)	8(0.156)	8(0.156)	8(0.156)
2^{-4}	18(0.281)	12(0.1870)	17(0.235)	17(0.125)	5(0.125)	16(0.265)	40(0.093)	53(0.859)
2^{-5}	40(0.833)	24(0.630)	17(0.235)	34(0.687)	6(0.281)	19(0.453)	69(1.756)	126(2.789)
2^{-6}	84(2.845)	44(1.640)	33(1.235)	62(2.072)	6(0.437)	22(1.022)	86(3.631)	212(9.415)
2^{-7}	170(21.02)	70(8.327)	40(4.565)	79(9.178)	6(1.244)	22(0.954)	104(15.264)	283(46.29)
2^{-8}	344(28.15)	132(38.21)	63(33.53)	87(78.03)	7(5.575)	23(19.2)	107(72.57)	-

TABLE 6. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	9(0.141)	9(0.171)	13(0.172)	13(0.187)	13(0.172)	9(0.250)	3(0.156)	3(0.078)
2^{-4}	13(0.250)	39(0.609)	86(1.312)	95(1.537)	13(0.266)	18(0.312)	3(0.060)	3(0.094)
2^{-5}	15(0.460)	43(1.172)	151(3.630)	189(4.736)	15(0.481)	19(0.50)	3(0.090)	3(0.265)
2^{-6}	17(0.891)	47(2.163)	186(8.860)	310(16.43)	17(0.875)	20(1.061)	17(0.223)	3(0.563)
2^{-7}	19(3.26)	49(7.755)	186(30.80)	310(15.98)	19(3.565)	22(3.928)	21(1.472)	10(5.370)
2^{-8}	24(16.90)	56(39.20)	195(161.0)	-	21(16.335)	26(19.288)	26(20.717)	10(17.53)

TABLE 7. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{Q}_1$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{Q}_{S_3}$				$\mathcal{Q}_1$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	11(0.120)	9(0.194)	3(0.0156)	3(0.125)	3(0.360)	2(0.110)	2(0.125)	2(0.109)
2^{-4}	11(0.178)	11(0.272)	3(0.026)	3(0.203)	3(0.156)	2(0.125)	2(0.109)	2(0.125)
2^{-5}	15(0.366)	17(0.535)	3(0.190)	3(0.235)	6(0.266)	3(0.175)	2(0.125)	2(0.114)
2^{-6}	15(0.622)	17(0.538)	11(0.406)	3(0.422)	9(0.422)	3(0.438)	2(0.281)	2(0.247)
2^{-7}	17(4.856)	19(4.840)	11(4.634)	5(3.519)	15(5.560)	5(0.594)	2(0.340)	2(0.319)
2^{-8}	17(11.936)	19(12.834)	15(11.830)	6(11.639)	33(7.536)	7(0.534)	2(0.480)	2(0.453)

yield good computing results in the form of very small iteration steps and computing times and the number of iterations are δ and h independent. Comparing the performance of the new preconditioner $\mathcal{Q}_1$ and $\mathcal{Q}_{S_3}$ from the Tables (3 and 4) we observe that the new preconditioner is superior in terms of the number of iterations and the CPU times. To sum up, the GMRES method applied with he new preconditioner converges in a small number of iterations and CPU times for different mesh sizes h and regularisation parameter δ. This is a different case with other preconditioners where the number of iterations and CPU time increase with the changes

TABLE 8. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BCT}, \mathcal{P}_N$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{P}_{BCT}$				$\mathcal{P}_N$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	69(2.223)	64(2.398)	58(2.112)	53(2.123)	14(0.239)	30(0.0.520)	54(0.578)	59(0.997)
2^{-4}	250(7.853)	202(9.115)	510(25.110)	-	14(0.298)	36(0.723)	96(1.250)	146(3.942)
2^{-5}	692(30.923)	-	-	-	13(0.385)	37(1.030)	119(3.298)	271(2.789)
2^{-6}	-	-	-	-	13(0.797)	37(4.723)	142(6.950)	517(8.021)
2^{-7}	-	-	-	-	13(2.335)	32(9.500)	160(28.530)	-
2^{-8}	-	-	-	-	13(15.657)	30(19.2)	166(13957)	-

TABLE 9. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	15(0.062)	31(0.110)	52(3.950)	70(0.186)	15(0.219)	14(0.265)	15(0.204)	10(0.141)
2^{-4}	15(1.25)	36(0.250)	96(4.971)	369(5.055)	15(0.250)	15(0.252)	16(0.283)	12(0.211)
2^{-5}	15(0.260)	43(0.725)	116(2.220)	670(116.7)	13(0.318)	15(0.568)	16(0.531)	12(0.554)
2^{-6}	15(1.640)	43(1.910)	134(5.430)	-	13(0.766)	16(0.856)	16(0.872)	13(0.334)
2^{-7}	17(7.368)	44(7.800)	153(24,20)	-	13(2.921)	16(0.919)	15(1.548)	13(1.598)
2^{-8}	17(15.904)	44(40.543)	165(112.5)	-	13(3.675)	15(2.843)	15(2.830)	13(2.354)

TABLE 10. Number of iterations and CPU times(sec) taken by inexact preconditioned GMRES solver with the block preconditioners $\mathcal{Q}_1$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{Q}_{S_3}$				$\mathcal{Q}_1$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	13(0146)	15(0.262)	14(0.162)	10(0.125)	29(0.072)	10(0.063)	2(0.047)	2(0.031)
2^{-4}	15(0.250)	16(0.212)	16(0.295)	11(0.214)	162(1.343)	15(0.218)	3(0.063)	2(0.047)
2^{-5}	13(0.525)	16(0.525)	16(0.666)	11(0.619)	771(14.94)	25(0.693)	3(0.143)	2(0.079)
2^{-6}	15(0.810)	15(0.628)	15(0.969)	13(0.886)	-	32(0.995)	3(0.642)	2(0.141)
2^{-7}	15(1.368)	15(0.866)	15(1.461)	13(1.458)	-	35(1.392)	3(1.109)	2(0.336)
2^{-8}	15(2.264)	15(1.982)	15(2.675)	13(2.639)	-	35(2.357)	3(2.157)	2(0.725)

in parameters. This agrees with the results in Figure 1. From the results stated in Tables (5, 6, 7, 8, 9, 10) show that we should choose $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{Q}_1$ for inexact preconditioning of the GMRES solver. We now present the second set of the numerical experiment results exact preconditioned GMRES, that is approximating the preconditioners subsystems in Algorithms (3.6 and 3.8) using incomplete Cholesky factorisation.

In Tables (12, 11, 13, 14, 15, 16) we list the number of iterations and computing times with respect to the preconditioners $\mathcal{P}_{BD}(S_1)$, $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{P}_{BCT}$, $\mathcal{P}_N$, $\mathcal{Q}_1$

TABLE 11. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $(\mathcal{P}_{BCT}, \mathcal{P}_N)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{P}_{BCT}$				$\mathcal{P}_N$			
	δ				δ			
h	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	2(0.079)	2(0.063)	2(0.062)	2(0.062)	2(0.063)	2(0.078)	2(0.079)	2(0.094)
2^{-4}	2(0.078)	2(0.078)	2(0.062)	2(0.094)	2(0.078)	2(0.0.78)	2(0.078)	2(0.078)
2^{-5}	2(0.094)	3(0.094)	3(0.094)	2(0.078)	2(0.125)	2(0.410)	2(0.099)	2(0.188)
2^{-6}	8(0.422)	8(0.406)	8(0.922)	9(0.466)	2(0.359)	2(0.328)	2(0.328)	2(0.360)
2^{-7}	20(5.588)	20(5.632)	19(5.310)	23(6.356)	2(1.410)	2(1.587)	2(1.179)	2(1.333)
2^{-8}	20(10.888)	20(10.72)	19(10.60)	23(10.56)	2(6.20)	2(5.687)	2(5.709)	2(5.324)

TABLE 12. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$			
	δ				δ			
h	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	3(0.078)	3(0.063)	3(0.063)	3(0.078)	3(0.063)	3(0.063)	3(0.046)	3(0.047)
2^{-4}	3(0.109)	3(0.094)	3(1.312)	3(1.534)	3(0.063)	3(0.078)	3(0.078)	3(0.078)
2^{-5}	3(0.124)	3(0.110)	3(3.633)	3(4.737)	3(0.078)	3(0.094)	3(0.094)	3(0.109)
2^{-6}	3(0.266)	3(0.218)	3(8.866)	3(16.41)	3(0.243)	3(0.250)	3(0.265)	3(0.281)
2^{-7}	3(0.354)	3(1.410)	3(30.82)	3(15.98)	3(1.402)	3(1.410)	3(1.387)	3(1.374)
2^{-8}	3(1.544)	3(4.410)	3(35.28)	3(37.28)	3(2.423)	3(2.341)	3(3.743)	3(3.647)

TABLE 13. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $\mathcal{Q}_{S_3}$ and $\mathcal{Q}_1$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.1.

	$\mathcal{Q}_{S_3}$				$\mathcal{Q}_1$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	3(0.056)	3(0.056)	3(0.039)	3(0.033)	2(0.062)	2(0.078)	2(0.078)	1(0.063)
2^{-4}	3(0.056)	3(0.064)	3(0.059)	3(0.068)	2(0.062)	2(0.079)	2(0.078)	1(0.062)
2^{-5}	3(0.064)	3(0.078)	3(0.079)	3(0.065)	1(0.140)	2(0.158)	1(0.078)	1(0.094)
2^{-6}	3(0.159)	3(0.228)	3(0.240)	3(0.236)	1(0.406)	1(0.406)	1(0.406)	1(0.266)
2^{-7}	3(0.720)	3(0.825)	3(0.823)	3(0.847)	1(0.574)	1(0.388)	1(0.398)	1(0.241)
2^{-8}	3(0.910)	3(0.921)	3(0.973)	3(0.978)	1(0.673)	1(0.390)	1(0.388)	1(0.255)

which are applied to precondition the GMRES solver for different δ and h for Problems (4.1 and 4.2). The subsystems in the application of the preconditioners are solved exactly using the incomplete Cholesky factorisation. We can observe from the results in the tables above that the computing efficiency of the preconditioners approximated exactly are independent of the h and δ for Problem 4.1 because it has a smooth data. For Problem 4.2 the exact application of preconditioners $\mathcal{P}_{BD}(S_1)$, $\mathcal{P}_{BCT}$, $\mathcal{P}_N$, the GMRES failed to converge for the small values of the

TABLE 14. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $(\mathcal{P}_{BCT}, \mathcal{P}_N)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{P}_{BCT}$				$\mathcal{P}_N$			
	δ				δ			
h	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	60(2.223)	64(2.398)	58(2.112)	53(2.113)	14(0.239)	30(0.578)	54(0.909)	59(0.994)
2^{-4}	172(7.853)	202(9.115)	510(25.11)	-	14(0.278)	36(0.728)	146(3.033)	146(2.942)
2^{-5}	448(30.924)	-	-	-	13(0.385)	37(1.030)	129(3.299)	264(8.021)
2^{-6}	-	-	-	-	35(1.859)	35(4.328)	123(16.32)	517(77.360)
2^{-7}	-	-	-	-	34(9.335)	30(93.587)	125(93.53)	-
2^{-8}	-	-	-	-	34(19.20)	32(120.687)	125(121.79)	-

TABLE 15. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $\mathcal{P}_{BD}(S_1), \mathcal{P}_{BD}(S_2)$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{P}_{BD}(S_1)$				$\mathcal{P}_{BD}(S_2)$			
	δ				δ			
h	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	15(0.062)	31(0.110)	84(3.95)	70(0.178)	11(0.063)	13(0.078)	7(0.047)	7(0.054)
2^{-4}	15(0.125)	36(0.250)	354(4.912)	369(5.534)	13(0.125)	13(0.125)	13(0.111)	9(0.111)
2^{-5}	15(0.264)	40(0.750)	-	-	13(0.318)	13(0.268)	13(0.354)	11(0.354)
2^{-6}	15(1.646)	37(3.918)	-	-	13(1.243)	13(0.750)	12(0.865)	11(0.781)
2^{-7}	15(2.354)	34(21.410)	-	-	11(9.402)	11(1.410)	11(1.387)	11(1.374)
2^{-8}	17(7.544)	34(46.410)	-	-	11(12.423)	11(9.341)	11(9.543)	11(3.647)

TABLE 16. Number of iterations and CPU times(sec) taken by exact preconditioned GMRES solver with the block preconditioners $\mathcal{Q}_{S_3}$ and $\mathcal{Q}_1$ for different values of h and δ, tolerance $= 10^{-6}$ for Problem 4.2.

	$\mathcal{Q}_{S_3}$				$\mathcal{Q}_1$			
	δ				δ			
	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-3}	10^{-5}	10^{-7}	10^{-9}
h	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)	Iter(cpu)
2^{-3}	13(0.046)	13(0.062)	11(0.062)	7(0.047)	29(0.672)	2(0.063)	1(0.047)	1(0.031)
2^{-4}	13(0.125)	13(0.112)	13(0.095)	9(0.094)	49(5.92)	1(0.063)	1(0.047)	1(0.037)
2^{-5}	13(0.125)	13(0.125)	13(0.279)	11(0.219)	61(0.0340)	1(0.093)	1(0.043)	1(0.022)
2^{-6}	13(0.559)	13(0.228)	13(0.540)	11(0.636)	-	1(0.395)	1(0.170)	1(0.022)
2^{-7}	11(0.620)	11(0.625)	11(0.823)	11(0.847)	-	1(0.392)	1(0.042)	1(0.041)
2^{-8}	11(8.710)	11(7.721)	11(7.491)	11(7.458)	-	1(0.390)	1(0.053)	(0.055)

parameters δ and h. The GMRES with the preconditioners $\mathcal{P}_{BD}(S_2)$, $\mathcal{Q}_{S_3}$, $\mathcal{Q}_1$ show excellent results for both problems. We see that the GMRES method incorporated with these preconditioners is computationally more efficient in terms of both iteration counts and computing times but $\mathcal{Q}_{S_3}$, $\mathcal{Q}_{S_2}$ and $\mathcal{Q}_1$ produced best performance results. The preconditioner $\mathcal{Q}_1$ produced the best results in terms of number of iterations and CP time. This shows the superiority of our new proposed preconditioner to the other preconditioners.

Table 17. Numerical results using preconditioner with approximation $\mathcal{Q}_1$ with $h = 2^{-6}$.

	Preconditioner $\mathcal{Q}_1$				
δ	$\parallel u_h \parallel_2$	$\parallel u_h - u \parallel_2$	$\parallel y_h - y \parallel_2$	$\parallel y - y_d \parallel_2$	$J(u_h, y_h)$
10^{-1}	1.58e+2	7.63e-4	1.56e-3	3.13e+1	4.07e+2
10^{-2}	1.29e+2	4.75e-4	1.28e-3	2.55e+1	4.07e+2
10^{-3}	4.56e+2	3.53e-4	4.50e-4	8.98	1.44e+2
10^{-4}	6.08e+2	7.43e-4	6.03e-5	1.20	1.92e+1
10^{-5}	6.29e+2	7.98e-4	6.30e-6	1.24e-1	1.98e-1
10^{-6}	6.32e+2	8.03e-4	6.29e-7	1.25e-2	2.0e-2
10^{-7}	6.32e+2	8.03e-4	6.26e-8	1.24e-3	2.0e-3
10^{-8}	6.32e-2	8.03e-4	6.32e-9	1.24e-4	2.0e-4
10^{-9}	6.32e+2	8.03e-4	6.32e-10	1.24e-5	2.0e-5

The results in Table 17 show the behaviour of the cost functional for different values of the regularisation parameter δ. It is well known that the δ determines how close the state approaches the desired state y_d. The results provide an interesting observations that $\parallel u \parallel_2$ stops increasing at 10^{-6} with $J(y, u)$ and $\parallel y - y_d \parallel_2$ decreases with the decrease by a constant factor of the decrease of δ. We observe that the state variable become very close to the desired state and also that the control variable $\parallel u \parallel_2$ increases as δ decreases. This is a clear indication that the cost functional will be insensitive to the control variable as δ decreases.

5. Conclusion

The PDE-constrained optimisation problem discretised by the finite element method results in a large scale linear algebraic system of saddle point form. The robust and efficient numerical solution of such problems strongly depends on the preconditioning strategies. The main task of this paper was to contribute to the long list of block preconditioners by developing a robust and efficient preconditioners for solving the saddle point systems. The preconditioners are built on the block structure of the coefficient matrix and can be employed to precondition the GMRES method. The theoretical analysis of the proposed preconditioner $\mathcal{Q}_1$ indicate that the spectral properties of the preconditioned matrix are clustered around 1 and the numerical experimental results confirm this with the preconditioned GMRES solver, when both the mesh size h and regularisation parameter δ decrease. We have compared the numerical results of the proposed preconditioner with those several preconditioners presented in literature both applied exactly and inexactly. For both inexact and exact application of the preconditioners applied numerically, the proposed preconditioner $\mathcal{Q}_1$ is the most efficient and effective compared to the other tested. The numerical results show that our new preconditioner can be an alternative choice for preconditioning Krylov subspace solvers and must be considered as a suitable and a better preconditioner for the considered and similar problems. The techniques and the numerical results highlighted in this paper can be extended to handle many practical real life problems which results in similar algebraic structure. Furthermore, the numerical results have demonstrated that the proposed preconditioner can be used practically and can be considered as a viable preconditioner for other classes of problems that occur in various fields such as finance, shape optimization, atmospheric and oceanic sciences, optimal heat control and among others.

Acknowledgments

The authors wish to acknowledge the financial support from University of Venda.

Conflict of Interests

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

References

[1] O. Axelsson and K. Salkuyeh. A new iteration and preconditioning method for elliptic pde-constrained optimization problem. Numer. Math. Theor. Meth. Appl., 13:1098–1122, 2020.
[2] Z. Z. Bai. Block preconditioners for elliptic pde-constrained optimization problems. Computing, 91:379–395, 2011.
[3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. Acta Numerica, 14:1–137, 2005.
[4] A. Borzi and V. Schulz. Multigrid methods for pde optimization. SIAM Review, 51:361–395, 2009.
[5] F. Durastante and I. Furci. Spectral analysis of saddle-point matrices from optimization problems with elliptic PDE constraints. The Electronic Journal of Linear Algebra, 36: 773-798, 2020
[6] C. H. Elman, A. Ramage, and J. D. Silvester. Algorithm 866. IFISS a matlab toolbox for modeling incompressible flow. ACM Transactions on Mathematical Software, 33(2): article 14, 2007.
[7] H. C. Elman, D. J. Silvester, and A. J. Wathen. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford University Press, Oxford/New York, 2014.
[8] W. Hackbusch. Fast solution of elliptic control problems. Journal of Optimisation Theory and Application, 31:565–581, 1980.
[9] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. Optimization with PDE constraints. Springer, New York, USA, 2009.
[10] Y. F. Ke and C. F. Ma. Some preconditioners for elliptic pde-constrained optimization problems. Computers and Mathematics with Applications, 75:2795–2813, 2018.
[11] J. L. Lions. Optimal control systems governed by partial differential equations. Springer, Berlin, Germany, 1971.
[12] H. Mirchi and D. K. Salkkuyeh. A new preconditioner for elliptic pde-constrained optimization problems. Numerical Algorithms, 83:663–668, 2020.
[13] K. Muzhinji and S. Shateyi. A robust approximation of the schur complement preconditioner for an efficient numerical solution of the elliptic optimal control problems. Computation, 8:68, 2020.
[14] J. W. Pearson, M. Stoll, and A. J. Wathen. Robust iterative solution of a class of time-dependent optimal control problems. Proceedings in Applied Mathathematics and Mechanics, 12:3–6, 2012.
[15] J. W. Pearson and A. J. Wathen. A new approximation of the schur complement in preconditioners for pde-constrained optimization. Numerical Linear Algebra with Applications, 19:816–829, 2012.
[16] J. W. Pestana and A. J. Wathen. Combination preconditioning of saddle point systems for positive definiteness. Numerical Linear Algebra with Applications, 20(5):785–808, 2013.
[17] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for pde-constrained optimization. SIAM Journal on Scientific Computing, 32:271–298, 2010.
[18] T. Rees and M. Stoll. Block-triangular preconditioners for PDE-constrained optimization. Numerical Linear Algebra with Applications, 17(6):977–996, 2010.
[19] J. Schöberl and W. Zulehner. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimisation problems. Journal on Matrix Analysis and Applications, 29:752–773, 2007.

[20] J. Schöberl, W. Zulehner, and R. Simon. A robust multigrid method for elliptic control problems. SIAM Journal on Numerical Analysis, 49:1482–1503, 2011.

[21] M. Stoll and A. J. Wathen. Combination preconditioning and the $Bramble - Pasciak^+$ preconditioner. Journal on Matrix Analysis and Applications, 30:582–608, 2008.

[22] F. Tröltzsch. Optimal control of partial differential equations. Theory, methods and applications. American Mathematical Society, Berlin, Germany, 2010.

[23] A. J. Wathen and T. Rees. Chebyshev semi-iteration in preconditioning for problems including the mass matrix. Electronic Transactions on Numerical Ananlysis, 34:125–135, 2009.

[24] G. Zhang and Z. Zheng. Block-symmetric and block-lower-triangular preconditioners for pde-constrained optimization problems. Journal of Computational Mathematics, 31:370–381, 2013.

Department of Mathematical and Computational Sciences, University of Venda, P. B. X5050, Thohoyandou 0950, South Africa.

E-mail: kizito.muzhinji@univen.ac.za

URL: https://orcid.org/ 0000-0002-7767-7453