Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 8(12): 30501–30510.
DOI: 10.3934/math.20231558
Received: 08 April 2023
Revised: 10 May 2023
Accepted: 23 May 2023
Published: 10 November 2023

*Research article*

# Structured backward errors analysis for generalized saddle point problems arising from the incompressible Navier-Stokes equations

**Peng Lv***

School of Mathematics and Statistics, Hubei University of Arts and Science, Xiangyang 441053, China

* **Correspondence:** Email: lvpeng@hbuas.edu.cn.

**Abstract:** Recently, a number of fast iteration methods for the solution of the structured linear system arising from the incompressible Navier-Stokes equations have been proposed by some authors. In order to evaluate the strong stability of these numerical algorithms, in this paper we deal with the structured backward error analysis for this type of structured linear system and present the explicit formula of the structured backward error. Based on the structured backward error, we perform some numerical experiments to compare the availability of some existing numerical algorithms.

**Keywords:** structured backward error; structure-preserving; saddle point problems
**Mathematics Subject Classification:** 15A06, 65F10, 65F50, 65G50

## 1. Introduction

In this paper, we consider the structured backward errors analysis for the structured linear system arising from the incompressible Navier-Stokes equations with the following form [9]:

$$\mathscr{A}\mathbf{u} = \begin{bmatrix} A_1 & 0 & B_1^T \\ 0 & A_2 & B_2^T \\ -B_1 & -B_2 & C \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ -g \end{bmatrix} = \mathbf{b}, \tag{1.1}$$

where $A_1 \in \mathbb{R}^{n_1\times n_1}, A_2 \in \mathbb{R}^{n_2\times n_2}$ are nonsymmetric positive definite matrices, $B_1 \in \mathbb{R}^{m\times n_1}, B_2 \in \mathbb{R}^{m\times n_2}$ has full row ranks, and $C \in \mathbb{SR}^{m\times m}$ is a symmetric positive semi-definite matrix; $\mathbb{R}^{m\times n}$ and $\mathbb{SR}^{m\times m}$ are the sets of $m\times n$ real matrices and $m\times m$ real symmetric matrices, respectively. These constraints guarantee the existence and uniqueness of the solution of the structured linear system (1.1).

Recently, there is a great variety of fast preconditioned Krylov subspace methods for solving the structured linear system (1.1) based on the specific block structure of coefficient matrix $\mathscr{A}$, such as dimensional splitting (DS) [1], relaxed dimensional factorization (RDF) [2], relaxed splitting

(RS) [22], modified dimensional split (MDS) [5], generalized relaxed splitting (GRS) [4], modified relaxed splitting (MRS) [10], relaxed block upper-lower triangular (RBULT) [16], relaxed upper and lower triangular splitting (RULT) [7], inexact modified relaxed splitting (IMRS) [15] preconditioned GMRES methods, and so on. In order to verify the validity and the strong stability of these numerical algorithms, one can performed the structured backward error analysis for the structured linear system (1.1). Given an approximate solution to a certain structured problem, structured backward error analysis involves finding a structure-preserving perturbation in the data of minimal size such that the approximate solution is an exact solution of the structure-preserving perturbed problem. The size of the smallest structure-preserving perturbation is called the structured backward error. In matrix computations, structured backward error analysis is useful not only to examine the structured stability (or strong stability [3]) of numerical algorithm, but also to design effective stopping criteria for the iterative solution of large sparse structured systems.

There has been substantial interest in structured backward error analysis in recent years. To our best knowledge, some scholars [6, 17, 18, 20, 23, 24] have performed the structured backward error analysis for some standard or generalized saddle point systems. Although the block structured linear system (1.1) can be viewed as a generalized 2 × 2 block saddle point problem, the aforementioned structured backward error analysis does not exactly show the case of the system (1.1) due to its special block structure. Naturally, a new and detailed analysis for the structured backward error of the linear system (1.1) with such special structure need to be performed. This paper will focus on this topic.

This paper is organized as follows. In Section 2, we first define the structured backward error of the structured linear system (1.1), and then derive its exact and computable formula. Based on the formula of structured backward error, in Section 3, we perform some numerical experiments to compare the validity of some existing numerical algorithms. Finally, in Section 4, we present some conclusions.

## 2. Structured backward error analysis

Similar to the structured backward error analysis for standard or generalized saddle point problems (see [6, 17, 18, 20, 23, 24]), we define the structured backward error for the linear system (1.1).

Let $\mathbf{\tilde{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ be the computed solution of the system (1.1), its **parameterized structured backward error** $\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p})$ can be defined as

$$\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p}) = \min_{\begin{pmatrix}\Delta A_1,\Delta A_2,\Delta B_1,\\ \Delta B_2,\Delta C,\Delta f_1,\\ \Delta f_2,\Delta g\end{pmatrix}\in\mathscr{F}} \left\| \begin{bmatrix} \Delta A_1 & 0 & \theta_3\Delta B_1^T & \lambda_1\Delta f_1 \\ 0 & \theta_1\Delta A_2 & \theta_4\Delta B_2^T & \lambda_2\Delta f_2 \\ \theta_3\Delta B_1 & \theta_4\Delta B_2 & \theta_2\Delta C & \lambda_3\Delta g \end{bmatrix} \right\|_F, \tag{2.1}$$

where the set $\mathscr{F}$ is defined by

$$\mathscr{F} = \left\{ \begin{pmatrix}\Delta A_1,\Delta A_2,\Delta B_1,\\ \Delta B_2,\Delta C,\Delta f_1,\\ \Delta f_2,\Delta g\end{pmatrix} : \begin{array}{c} \begin{bmatrix} A_1+\Delta A_1 & 0 & (B_1+\Delta B_1)^T \\ 0 & A_2+\Delta A_2 & (B_2+\Delta B_2)^T \\ -(B_1+\Delta B_1) & -(B_2+\Delta B_2) & C+\Delta C \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{p} \end{bmatrix} = \begin{bmatrix} f_1+\Delta f_1 \\ f_2+\Delta f_2 \\ -g-\Delta g \end{bmatrix}, \\ \Delta C = \Delta C^T \end{array} \right\} \tag{2.2}$$

and $\theta_1, \theta_2, \theta_3, \theta_4, \lambda_1, \lambda_2, \lambda_3$ are positive parameters that can be adjusted to emphasize the requisite perturbations more than others. A special set of selections is

$$\tilde{\theta}_1 \equiv \frac{\|A_1\|_F}{\|A_2\|_F},\ \tilde{\theta}_2 \equiv \frac{\|A_1\|_F}{\|C\|_F},\ \tilde{\theta}_3 \equiv \frac{\|A_1\|_F}{\|B_1\|_F},\ \tilde{\theta}_4 \equiv \frac{\|A_1\|_F}{\|B_2\|_F}, \tilde{\lambda}_1 \equiv \frac{\|A_1\|_F}{\|f_1\|_F},\ \tilde{\lambda}_2 \equiv \frac{\|A_1\|_F}{\|f_2\|_2},\ \tilde{\lambda}_3 \equiv \frac{\|A_1\|_F}{\|g\|_2} \tag{2.3}$$

which yields the **relative structured backward error**

$$\eta_S(\tilde{\mathbf{u}}) = \eta^{\left(\tilde{\theta}_1,\tilde{\theta}_2,\tilde{\theta}_3,\tilde{\theta}_4,\tilde{\lambda}_1,\tilde{\lambda}_2,\tilde{\lambda}_3\right)}(\tilde{u}_1,\tilde{u}_2,\tilde{p}) \,/\, \|A_1\|_F\,, \tag{2.4}$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius norm and Euclidean norm, respectively.

It can be seen from the above definition that a small $\eta_S(\tilde{\mathbf{u}})$ means the computed solution $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ is the exact solution of a slightly perturbed structure-preserving linear system. We call $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ a structured backward stable solution to (1.1) and the corresponding numerical algorithm is structured backward stable if $\eta_S(\tilde{\mathbf{u}})$ is a small multiple of the machine precision. Consequently, finding the exact and computable formula of the structured backward errors $\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1,\tilde{u}_2,\tilde{p})$ will be useful for testing the strong stability of a practical algorithm.

In the following, we give the explicit expression of parameterized structured backward error $\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1,\tilde{u}_2,\tilde{p})$.

**Theorem 2.1.** *Let* $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ *with* $\tilde{p} \neq 0$ *be a computed solution to the structured linear system* (1.1). *Then*

$$\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1,\tilde{u}_2,\tilde{p}) = \left\|P_K^{\perp} d\right\|_2, \tag{2.5}$$

*where*

$$K = \begin{bmatrix} \theta_3 I_{mn_1} & 0 & 0 \\ 0 & \theta_4 I_{mn_2} & 0 \\ 0 & 0 & \lambda_3 I_m \\ -\frac{I_{n_1}\otimes\tilde{p}^T}{\|\hat{u}_1\|_2} & 0 & 0 \\ 0 & -\frac{I_{n_2}\otimes\tilde{p}^T}{\|\hat{u}_2\|_2} & 0 \\ \frac{\theta_2\left(\tilde{u}_1^T\otimes I_m\right)}{\|\tilde{p}\|_2} & \frac{\theta_2\left(\tilde{u}_2^T\otimes I_m\right)}{\|\tilde{p}\|_2} & -\frac{\theta_2 I_m}{\|\tilde{p}\|_2} \\ \frac{\theta_2\left(\tilde{u}_1^T\otimes\left(I_m-\tilde{p}\tilde{p}^{\dagger}\right)\right)}{\|\tilde{p}\|_2} & \frac{\theta_2\left(\tilde{u}_2^T\otimes\left(I_m-\tilde{p}\tilde{p}^{\dagger}\right)\right)}{\|\tilde{p}\|_2} & -\frac{\theta_2\left(I_m-\tilde{p}\tilde{p}^{\dagger}\right)}{\|\tilde{p}\|_2} \end{bmatrix} \in \mathbb{R}^{s\times t}, d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{r_{f_1}}{\|\hat{u}_1\|_2} \\ \frac{r_{f_2}}{\|\hat{u}_2\|_2} \\ \frac{\theta_2 r_g}{\|\tilde{p}\|_2} \\ \frac{\theta_2\left(I_m-\tilde{p}\tilde{p}^{\dagger}\right)r_g}{\|\tilde{p}\|_2} \end{bmatrix} \in \mathbb{R}^s, \tag{2.6}$$

$$r_{f_1} = f_1 - A_1\tilde{u}_1 - B_1^T\tilde{p},\ r_{f_2} = f_2 - A_2\tilde{u}_2 - B_2^T\tilde{p},\ r_g = -g + B_1\tilde{u}_1 + B_2\tilde{u}_2 - C\tilde{p}, \tag{2.7}$$

$$\hat{u}_1 = \left(\tilde{u}_1^T, 1/\lambda_1\right)^T,\ \hat{u}_2 = \left(\tilde{u}_2^T/\theta_1, 1/\lambda_2\right)^T, \tag{2.8}$$

*and*

$$s = mn_1 + mn_2 + 3m + n_1 + n_2, t = mn_1 + mn_2 + m.$$

*Proof.* It is seen from (2.2) that $\begin{pmatrix} \Delta A_1, \Delta A_2, \Delta B_1, \\ \Delta B_2, \Delta C, \Delta f_1, \\ \Delta f_2, \Delta g \end{pmatrix} \in \mathscr{F}$ if and only if $\Delta A_1$, $\Delta A_2$, $\Delta B_1$, $\Delta B_2$, $\Delta C$, $\Delta f_1$, $\Delta f_2$, $\Delta g$ satisfy

$$\Delta A_1\tilde{u}_1 - \Delta f_1 = r_{f_1} - \Delta B_1^T\tilde{p},\ \Delta A_2\tilde{u}_2 - \Delta f_2 = r_{f_2} - \Delta B_2^T\tilde{p},\ \Delta C\tilde{p} = w \text{ and } \Delta C = \Delta C^T, \tag{2.9}$$

i.e.,

$$(\Delta A_1, -\lambda_1\Delta f_1)\begin{pmatrix} \tilde{u}_1 \\ \frac{1}{\lambda_1} \end{pmatrix} = r_{f_1} - \Delta B_1^T\tilde{p},\ (\theta_1\Delta A_2, -\lambda_2\Delta f_2)\begin{pmatrix} \frac{1}{\theta_1}\tilde{u}_2 \\ \frac{1}{\lambda_2} \end{pmatrix} = r_{f_2} - \Delta B_2^T\tilde{p},$$

and

$$(\theta_2\Delta C)\left(\frac{1}{\theta_2}\tilde{p}\right) = w,\ \Delta C = \Delta C^T,$$

where

$$w := r_g - \Delta g + \Delta B_1 \tilde{u}_1 + \Delta B_2 \tilde{u}_2.$$

From the above equations and (2.8), and using the well-known conclusions of Lemmas 2.1 and 2.2 in [20] or in [21], we have

$$(\Delta A_1, -\lambda_1 \Delta f_1) = \left(r_{f_1} - \Delta B_1^T \tilde{p}\right) \hat{u}_1^{\dagger} + Z_1 \left(I_{n_1+1} - \hat{u}_1 \hat{u}_1^{\dagger}\right),$$

$$(\theta_1 \Delta A_2, -\lambda_2 \Delta f_2) = \left(r_{f_2} - \Delta B_2^T \tilde{p}\right) \hat{u}_2^{\dagger} + Z_2 \left(I_{n_2+1} - \hat{u}_2 \hat{u}_2^{\dagger}\right),$$

and

$$\theta_2 \Delta C = \theta_2 w \tilde{p}^{\dagger} + \theta_2 \left(w \tilde{p}^{\dagger}\right)^T \left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) + \left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) T \left(I_m - \tilde{p}\tilde{p}^{\dagger}\right),$$

where $Z_1 \in \mathbb{R}^{n_1 \times (n_1+1)}$, $Z_2 \in \mathbb{R}^{n_2 \times (n_2+1)}$, $T \in \mathbb{R}^{m \times m}$. Due to the fact that

$$\hat{u}_1^{\dagger} \left(I_{n_1+1} - \hat{u}_1 \hat{u}_1^{\dagger}\right) = 0,\ \hat{u}_2^{\dagger} \left(I_{n_2+1} - \hat{u}_2 \hat{u}_2^{\dagger}\right) = 0$$

and $\tilde{p}^{\dagger} \left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) = 0$, we have

$$\|\Delta A_1\|_F^2 + \lambda_1^2 \|\Delta f_1\|_2^2 = \frac{\left\|r_{f_1} - \Delta B_1^T \tilde{p}\right\|_2^2}{\|\hat{u}_1\|_2^2} + \left\|Z_1 \left(I_{n_1+1} - \hat{u}_1 \hat{u}_1^{\dagger}\right)\right\|_F^2, \tag{2.10}$$

$$\theta_1^2 \|\Delta A_2\|_F^2 + \lambda_2^2 \|\Delta f_2\|_2^2 = \frac{\left\|r_{f_2} - \Delta B_2^T \tilde{p}\right\|_2^2}{\|\hat{u}_2\|_2^2} + \left\|Z_2 \left(I_{n_2+1} - \hat{u}_2 \hat{u}_2^{\dagger}\right)\right\|_F^2, \tag{2.11}$$

and

$$\theta_2^2 \|\Delta C\|_F^2 = \frac{\theta_2^2 \|w\|_2^2}{\|\tilde{p}\|_2^2} + \frac{\theta_2^2 \left\|\left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) w\right\|_2^2}{\|\tilde{p}\|_2^2} + \left\|\left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) T \left(I_m - \tilde{p}\tilde{p}^{\dagger}\right)\right\|_F^2. \tag{2.12}$$

It follows from the definition (2.1) of $\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p})$ and (2.10)–(2.12) that

$$\begin{aligned}
&\left(\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p})\right)^2 \\
&= \min_{\substack{Z_1 \in \mathbb{R}^{(n_1+1)\times(n_1+1)}, Z_2 \in \mathbb{R}^{(n_2+1)\times(n_2+1)}, \\ T \in \mathbb{R}^{m \times m}, \Delta B_1 \in \mathbb{R}^{m \times n_1}, \Delta B_2 \in \mathbb{R}^{m \times n_2}, \Delta g \in \mathbb{R}^m}} \left\{ \begin{array}{l} \|\Delta A_1\|_F^2 + \theta_1^2 \|\Delta A_2\|_F^2 + \theta_2^2 \|\Delta C\|_F^2 + \theta_3^2 \|\Delta B_1\|_F^2 \\ + \theta_4^2 \|\Delta B_2\|_F^2 + \lambda_1^2 \|\Delta f_1\|_2^2 + \lambda_2^2 \|\Delta f_2\|_2^2 + \lambda_3^2 \|\Delta g\|_2^2 \end{array} \right\} \\
&= \min_{\Delta B_1 \in \mathbb{R}^{m \times n_1}, \Delta B_2 \in \mathbb{R}^{m \times n_2}, \Delta g \in \mathbb{R}^m} p(\Delta B_1, \Delta B_2, \Delta g),
\end{aligned}$$

where

$$\begin{aligned}
p(\Delta B_1, \Delta B_2, \Delta g) = {} & \frac{\left\|r_{f_1} - \Delta B_1^T \tilde{p}\right\|_2^2}{\|\hat{u}_1\|_2^2} + \frac{\left\|r_{f_2} - \Delta B_2^T \tilde{p}\right\|_2^2}{\|\hat{u}_2\|_2^2} + \frac{\theta_2^2 \|w\|_2^2}{\|\tilde{p}\|_2^2} + \frac{\theta_2^2 \left\|\left(I_m - \tilde{p}\tilde{p}^{\dagger}\right) w\right\|_2^2}{\|\tilde{p}\|_2^2} \\
& + \theta_3^2 \|\Delta B_1\|_F^2 + \theta_4^2 \|\Delta B_2\|_F^2 + \lambda_3^2 \|\Delta g\|_2^2 .
\end{aligned}$$

Using the Kronecker product [11, 12], we have

$$\operatorname{vec}\left(r_{f_1} - \Delta B_1^T \tilde{p}\right) = r_{f_1} - \left(I_{n_1} \otimes \tilde{p}^T\right) \operatorname{vec}(\Delta B_1),\ \operatorname{vec}\left(r_{f_2} - \Delta B_2^T \tilde{p}\right) = r_{f_2} - \left(I_{n_2} \otimes \tilde{p}^T\right) \operatorname{vec}(\Delta B_2),$$

$$\operatorname{vec}(w) = r_g - \Delta g + \left(\tilde{u}_1^T \otimes I_m\right)\operatorname{vec}(\Delta B_1) + \left(\tilde{u}_2^T \otimes I_m\right)\operatorname{vec}(\Delta B_2),$$

and

$$\begin{aligned}\operatorname{vec}\left((I_m - \tilde{p}\tilde{p}^\dagger)w\right) = \left(I_m - \tilde{p}\tilde{p}^\dagger\right) r_g - \left(I_m - \tilde{p}\tilde{p}^\dagger\right)\Delta g + \left(\tilde{u}_1^T \otimes (I_m - \tilde{p}\tilde{p}^\dagger)\right)\operatorname{vec}(\Delta B_1)\\ + \left(\tilde{u}_2^T \otimes (I_m - \tilde{p}\tilde{p}^\dagger)\right)\operatorname{vec}(\Delta B_2).\end{aligned}$$

Then

$$\begin{aligned}\left(\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p})\right)^2 &= \min_{\Delta B_1 \in \mathbb{R}^{m\times n_1}, \Delta B_2 \in \mathbb{R}^{m\times n_2}, \Delta g \in \mathbb{R}^m} \left\| K \begin{pmatrix}\operatorname{vec}(\Delta B_1)\\ \operatorname{vec}(\Delta B_2)\\ \Delta g\end{pmatrix} + d \right\|_2^2\\ &= \left\| \left(I_s - KK^\dagger\right) d \right\|_2^2,\end{aligned}$$

in which $K$ and $d$ are defined as those in (2.6), and here $K^\dagger$ stands for the Moore-Penrose inverse [11] of $K$. □

Reconsidering the structured backward error $\eta^{(\theta_1,\theta_2,\theta_3,\theta_4,\lambda_1,\lambda_2,\lambda_3)}(\tilde{u}_1, \tilde{u}_2, \tilde{p})$, by broadening the structure-preserving constraint in (2.2) to unstructured constraint, we can get the relative unstructured backward error $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ which defined by [13, 19]

$$\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}}) = \min_{\Delta\mathscr{A},\Delta\mathbf{b}} \left\{ \left\| \left( \frac{\|\Delta\mathscr{A}\|_F}{\|\mathscr{A}\|_F}, \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \right) \right\|_F : (\mathscr{A} + \Delta\mathscr{A})\,\tilde{\mathbf{u}} = \mathbf{b} + \Delta\mathbf{b} \right\} = \frac{\|\mathbf{b} - \mathscr{A}\tilde{\mathbf{u}}\|_2}{\sqrt{\|\mathscr{A}\|_F^2 \|\tilde{\mathbf{u}}\|_2^2 + \|\mathbf{b}\|_2^2}}.$$

In addition, if only the right-side is perturbed, yields

$$\eta_{\mathbf{b}}(\tilde{\mathbf{u}}) = \min_{\Delta\mathbf{b}} \left\{ \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} : \mathscr{A}\tilde{\mathbf{u}} = \mathbf{b} + \Delta\mathbf{b} \right\} = \frac{\|\mathbf{b} - \mathscr{A}\tilde{\mathbf{u}}\|_2}{\|\mathbf{b}\|_2},$$

which often used as the stopping criterion for the iterative methods.

We note that a small $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ means the computed solution $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ is the exact solution of a slightly perturbed linear system. We call $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ a backward stable solution to (1.1) and the corresponding numerical algorithm is backward stable if $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ is a small multiple of the machine precision. It is worth noting that a backward stable solution may be not the exact solution of a slightly perturbed structure-preserving linear system (1.1). In other words, the relative structured backward error $\eta_S(\tilde{\mathbf{u}})$ may be much large than the relative unstructured backward error $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$. Next, we given an example to illustrate it.

**Example 2.1.** *Consider the structured linear system* (1.1) *with*

$$A_1 = M(1:3, 1:3),\ A_1 = M(4:6, 4:6),\ B_1 = B_2 = \begin{bmatrix} 0 & 0 & 1\\ 0 & 1 & 0\\ 10^4 & 0 & 0\end{bmatrix},\ C = 10^{-14} \times \begin{bmatrix} 1 & -2 & 1\\ -2 & 6 & 0\\ 1 & 0 & 0\end{bmatrix}$$

*and*

$$f_1 = \left(10^8, 10, 0\right)^T,\ f_2 = \left(10^8, 1, 0\right)^T,\ g = \left(10^{-8}, 0, 0\right)^T,$$

*where* $M = D_1PD_2$, $D_1 = \operatorname{diag}\left(1, 5, 10, 50, 100, 10^4\right)$, $D_2 = \operatorname{diag}(1, 5, 100, 1, 5, 10)$ *and* $P$ *is the Pascal matrix of order 6. Using Gaussian elimination with partial pivoting, we obtain a computed solution* $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ *with*

$$\tilde{u}_1 = \begin{pmatrix} -4.5172\times 10^{-1} \\ 9.8272\times 10^{-2} \\ -1.4527\times 10^{-2} \end{pmatrix},\ \tilde{u}_2 = \begin{pmatrix} 4.5172\times 10^{-1} \\ -9.8272\times 10^{-2} \\ 1.4527\times 10^{-2} \end{pmatrix},\ \tilde{p} = \begin{pmatrix} 7.6937\times 10^{1} \\ 2.9135\times 10^{1} \\ 1.0000\times 10^{4} \end{pmatrix}.$$

*Then, in view of* (2.4)*, we have the relative structured backward error*

$$\eta_S(\tilde{\mathbf{u}}) = 7.7318\times 10^{-06},$$

*and the relative unstructured backward error*

$$\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}}) = 5.8850\times 10^{-20},\ \eta_{\mathbf{b}}(\tilde{\mathbf{u}}) = 1.0812\times 10^{-16}.$$

*It is seen from the above results that Gaussian elimination with partial pivoting for solving this problem is backward stable but not structured backward stable, and the relative structured backward error* $\eta_S(\tilde{\mathbf{u}})$ *can indeed be much larger than the unstructured backward error* $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$*. This implies that the structured backward error provides a more reliable measure for assessing accuracy of a computed solution of the structured linear system* (1.1).

## 3. Numerical examples

In this section, we will present some test examples to examine the stability and effectiveness of some existing preconditioners for the generalized saddle point problem (1.1) by the structured backward error analysis. These problems arises from the discretization of the 2D linearized steady-state Navier-Stokes equation, i.e., the steady Oseen equation of the form:

$$\begin{cases} -v\Delta u + (\omega\cdot\nabla)\,u + \nabla p = f, \\ \nabla\cdot u = 0, \end{cases} \quad \text{in } \Omega, \tag{3.1}$$

where $\Omega$ is a bounded domain, $v > 0$ is the viscosity, and $\omega$ is the viscosity field. The vector field $u$ stands for the velocity, and $p$ represents the pressure. We use the IFISS software package developed by Elman et al. [8] to generate discretizations of the "regularized" two-dimensional lid-driven cavity problem for the Oseen equation (3.1). The mixed finite element used here is the bilinear pressure Q1-P0 pair with local stabilization. In addition, we use the uniform grids of increasing size and the known viscosity scalar and others are follows the default setting.

We apply the GMRES method in conjunction with the preconditioners RBULT [16], MRS [10], GRS [4] and MDS [5] to solve the generalized saddle point problem (1.1). All runs are started from the initial zero vector and terminated if the current iterations satisfy $RES = ||\mathbf{b} - \mathscr{A}\tilde{\mathbf{u}}||_2 \,/\, ||\mathbf{b}||_2 < 10^{-14}$. In actual computations, the subsystems of linear equations arising in the applications of the preconditioners are solved by the Cholesky or the LU factorization in combination with AMD or column AMD reordering. We choose the parameters in the preconditioned GMRES methods by using the algebraic estimation technique [14]. The symbols "IT" and "CPU" stand for the iteration counts and total CPU time respectively. All experiments were run on a PC with 3.30 GHz central processing unit (Intel(R) Core(TM) i7-11370H), 16 GB memory and Windows 10 operating system using MATLAB 2014a with machine precision $2.2204 \times 10^{-16}$.

For different grids and viscosities, the iteration counts and elapsed CPU times of GMRES with four preconditioners, the residual, the unstructured backward errors $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ and the structured backward errors $\eta_S(\tilde{\mathbf{u}})$ with respect to the final iteration solutions $\tilde{\mathbf{u}} = \left(\tilde{u}_1^T, \tilde{u}_2^T, \tilde{p}^T\right)^T$ are listed in Tables 1–4. It is seen from Tables 1–4 that the structured backward errors are about one order of magnitude larger than the unstructured one for each test problem and they are both of the order of unit round-off, which indicates that the preconditioned GMRES methods are backward stable and strongly stable for solving these test problems. In addition, we see from the iteration numbers, the elapsed CPU times and the structured backward errors that the MRS preconditioned GMRES method is more accuracy (strongly stable) and effective than those of the other preconditioned GMRES methods.

**Table 1.** Preconditioned GMRES methods numerical results for the Oseen problem with $v = 0.5$.

| Grids | | RBULT | MRS | GRS | MDS |
|---|---|---|---|---|---|
| $4 \times 4$ | IT | 13 | 15 | 16 | 21 |
| | CPU | 0.0526 | 0.0097 | 0.0058 | 0.0111 |
| | RES | 2.7696e-15 | 4.3654e-16 | 3.8489e-15 | 2.9127e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.3540e-16 | 4.8642e-17 | 2.8253e-16 | 2.4661e-16 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 4.5049e-16 | 2.7545e-16 | 1.3018e-15 | 8.4573e-16 |
| $8 \times 8$ | IT | 31 | 22 | 32 | 29 |
| | CPU | 0.0727 | 0.0240 | 0.0362 | 0.0384 |
| | RES | 8.1142e-15 | 2.6728e-15 | 7.0315e-15 | 8.3449e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.2244e-16 | 4.6301e-17 | 1.0205e-16 | 9.5598e-17 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 2.6232e-15 | 3.9443e-16 | 2.1595e-15 | 9.2548e-16 |

**Table 2.** Preconditioned GMRES methods numerical results for the Oseen problem with $v = 0.1$.

| Grids | | RBULT | MRS | GRS | MDS |
|---|---|---|---|---|---|
| $4 \times 4$ | IT | 13 | 15 | 16 | 21 |
| | CPU | 0.0402 | 0.0099 | 0.0072 | 0.0123 |
| | RES | 1.3354e-15 | 6.3830e-16 | 4.3890e-15 | 2.9338e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.0812e-16 | 5.6560e-17 | 3.5954e-16 | 2.5398e-16 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 3.4183e-16 | 2.9241e-16 | 1.9758e-15 | 8.9378e-16 |
| $8 \times 8$ | IT | 31 | 22 | 32 | 29 |
| | CPU | 0.0760 | 0.0264 | 0.0357 | 0.0373 |
| | RES | 9.6628e-15 | 2.4086e-15 | 9.2769e-15 | 7.6370e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.5059e-16 | 4.1375e-17 | 1.3495e-16 | 8.9264e-17 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 3.1619e-15 | 3.6270e-16 | 2.8529e-15 | 8.6730e-16 |

**Table 3.** Preconditioned GMRES methods numerical results for the Oseen problem with $v = 0.05$.

| Grids | | RBULT | MRS | GRS | MDS |
|---|---|---|---|---|---|
| $4 \times 4$ | IT | 13 | 15 | 16 | 21 |
| | CPU | 0.0341 | 0.0080 | 0.0060 | 0.0099 |
| | RES | 3.1126e-15 | 6.6461e-16 | 3.7661e-15 | 2.9099e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.5922e-16 | 4.9428e-17 | 3.2641e-16 | 2.5155e-16 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 4.8855e-16 | 2.5989e-16 | 1.8577e-15 | 8.8886e-16 |
| $8 \times 8$ | IT | 31 | 22 | 32 | 29 |
| | CPU | 0.0718 | 0.0268 | 0.0367 | 0.0345 |
| | RES | 9.7311e-15 | 2.2431e-15 | 9.5202e-15 | 7.3532e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.5364e-16 | 4.5875e-17 | 1.3954e-16 | 8.6810e-17 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 3.2157e-15 | 3.7135e-16 | 2.9603e-15 | 8.4309e-16 |

**Table 4.** Preconditioned GMRES methods numerical results for the Oseen problem with $v = 0.01$.

| Grids | | RBULT | MRS | GRS | MDS |
|---|---|---|---|---|---|
| $4 \times 4$ | IT | 13 | 15 | 16 | 21 |
| | CPU | 0.0337 | 0.0109 | 0.0067 | 0.0119 |
| | RES | 5.0086e-15 | 6.8143e-16 | 4.5317e-15 | 2.8925e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 2.6048e-16 | 6.5806e-17 | 3.6459e-16 | 2.5462e-16 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 7.4883e-16 | 3.1987e-16 | 2.0524e-15 | 9.1407e-16 |
| $8 \times 8$ | IT | 31 | 22 | 32 | 29 |
| | CPU | 0.0737 | 0.0237 | 0.0315 | 0.0344 |
| | RES | 9.7263e-15 | 2.0932e-15 | 9.6089e-15 | 7.1474e-15 |
| | $\eta_{\mathscr{A},\mathbf{b}}(\tilde{\mathbf{u}})$ | 1.5504e-16 | 3.8421e-17 | 1.4355e-16 | 9.0864e-17 |
| | $\eta_S(\tilde{\mathbf{u}})$ | 3.2365e-15 | 3.1490e-16 | 2.9856e-15 | 8.6299e-16 |

## 4. Concluding remarks

In this paper, we discussed the structured backward error analysis for the generalized saddle point problems arising from the incompressible Navier-Stokes equations and obtained the explicit expressions of the structured backward error. The structured backward error may be much large than the relative unstructured backward error which make it more suitable for assessing the validity and practicability of the numerical algorithms for solving the structured linear system (1.1). In addition, we also presented some test examples to examine the accuracy (strongly stability) and effectiveness of some existing preconditioned GMRES methods by the structured backward error.

## Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

The authors would like to express their gratitude to the anonymous referees for their detailed and helpful suggestions that substantially improved the manuscript. This work was partially supported by Research ability cultivation fund of HUAS(Hubei University of Arts and Science) (Nos. 2021kpgpzk04, 2020kypytd006) and Excellent young and middle-aged science and Technology Innovation team project of the Education Department of Hubei Province (T2022029).

## Conflict of interest

The authors disclosed no conflicts of interest in publishing this paper.

## References

1. M. Benzi, X. P. Guo, A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations, *Appl. Numer. Math.*, **61** (2011), 66–76.
2. M. Benzi, M. K. Ng, Q. Niu, Z. Wang, A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **230** (2011), 6185–6202.
3. J. R. Bunch, W. James Demmel, C. F. Van Loan, The strong stability of algorithms for solving symmetric linear systems, *SIAM J. Matrix Anal. Appl.*, **10** (1989), 494–499. https://doi.org/10.1137/0610035
4. Y. Cao, S. X. Miao, Y. S. Cui, A relaxed splitting preconditioner for genenralized saddle point problems, *Comput. Appl. Math.*, **34** (2015), 865–879. https://doi.org/10.1007/s40314-014-0150-y
5. Y. Cao, L. Q. Yao, M. Q. Jiang, A modified dimensional split preconditioner for generalized saddle point problems, *J. Comput. Appl. Math.*, **250** (2013), 70–82. https://doi.org/10.1016/j.cam.2013.02.017
6. X. S. Chen, W. Li, X. Chen, J. Liu, Structured backward errors for generalized saddle point systems, *Linear Algebra Appl.*, **436** (2012), 3109–3119. https://doi.org/10.1016/j.laa.2011.10.012
7. G. Cheng, J. C. Li, A relaxed upper and lower triangular splitting preconditioner for the linearized Navier–Stokes equation, *Comput. Math. Appl.*, **80** (2020), 43–60. https://doi.org/10.1016/j.camwa.2020.02.025
8. H. C. Elman, A. Ramage, D. J. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow, *ACM Trans. Math. Software*, **33** (2007), 14. https://doi.org/10.1145/1236463.1236469
9. H. C. Elman, D. J. Silvester, A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford: Oxford University Press, 2014.
10. H. T. Fan, X. Y. Zhu, A modified relaxed splitting preconditioner for generalized saddle point problems from the incompressible Navier-Stokes equations, *Appl. Math. Lett.*, **55** (2016), 18–26. https://doi.org/10.1016/j.aml.2015.11.011
11. G. H. Golub, C. F. Van Loan, *Matrix Computations*, Baltimore: The Johns Hopkins University Press, 2013.

12. A. Graham, *Kronecker Products and Matrix Calculus with Application*, New York: Wiley, 1981.

13. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Philadelphia: SIAM, 2002. https://doi.org/10.1137/1.9780898718027

14. Y. M. Huang, A practical formula for computing optimal parameters in the HSS iteration methods, *J. Comput. Appl. Math.*, **255** (2014), 142–149. https://doi.org/10.1016/j.cam.2013.01.023

15. Y. F. Ke, C. F. Ma, An inexact modified relaxed splitting preconditioner for the generalized saddle point problems from the incompressible Navier-Stokes equation, *Numer. Algor.*, **75** (2017), 1103–1121. https://doi.org/10.1007/s11075-016-0233-5

16. Y. J. Li, X. Y. Zhu, H. T. Fan, Relaxed block upper–lower triangular preconditioner for generalized saddle point problems from the incompressible Navier-Stokes equations, *J. Comput. Appl. Math.*, **364** (2020), 112329. https://doi.org/10.1016/j.cam.2019.06.045

17. P. Lv, B. Zheng, Structured backward error analysis for a class of block three-by-three saddle point problems, *Numer. Algor.*, **90** (2022), 59–78. https://doi.org/10.1007/s11075-021-01179-6

18. L. S. Meng, Y. W. He, S. X. Miao, Structured backward errors for two kinds of generalized saddle point systems, *Linear Multilinear Algebra*, **70** (2022), 1345–1355. https://doi.org/10.1080/03081087.2020.1760193

19. J. L. Rigal, J. Gaches, On the compatibility of a given solution with the data of a linear system, *J. Assoc. Comput. Mach.*, **14** (1967), 543–548. https://doi.org/10.1145/321406.321416

20. J. G. Sun, Structured backward errors for KKT systems, *Linear Algebra Appl.*, **288** (1999), 75–88. https://doi.org/10.1016/S0024-3795(98)10184-2

21. J. G. Sun, *Matrix Perturbation Analysis*, Beijing: Science Press, 2001.

22. N. B. Tan, T. Z. Huang, Z. J. Hu, A relaxed splitting preconditioner for the incompressible Navier–Stokes equations, *J. Appl. Math.*, **2012** (2012), 402490. https://doi.org/10.1155/2012/402490

23. H. Xiang, Y. M. Wei, On normwise structured backward errors for saddle point systems, *SIAM J. Matrix Anal. Appl.*, **29** (2007), 838–849. https://doi.org/10.1137/060663684

24. B. Zheng, P. Lv, Structured backward error analysis for generalized saddle point problems, *Adv. Comput. Math.*, **46** (2020), 34. https://doi.org/10.1007/s10444-020-09787-x