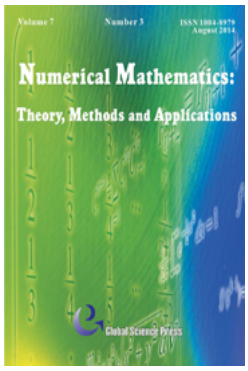


Comparison of Some Preconditioners for the Incompressible Navier-Stokes Equations

X. He and C. Vuik

Numerical Mathematics: Theory, Methods and Applications / Volume 9 / Issue 02 / May 2016, pp 239 - 261
DOI: 10.4208/nmtma.2016.m1422, Published online: 24 May 2016

Link to this article: http://journals.cambridge.org/abstract_S1004897916000118

How to cite this article:
X. He and C. Vuik (2016). Comparison of Some Preconditioners for the Incompressible Navier-Stokes Equations. Numerical Mathematics: Theory, Methods and Applications, 9, pp 239-261 doi:10.4208/nmtma.2016.m1422

Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2016.m1422

Vol. **9**, No. 2, pp. 239-261
May 2016

Comparison of Some Preconditioners for the Incompressible Navier-Stokes Equations

X. He*and C. Vuik

Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, the Netherlands.

Received 7 August 2014; Accepted 6 March 2015

Abstract. In this paper we explore the performance of the SIMPLER, augmented Lagrangian, 'grad-div' preconditioners and their new variants for the two-by-two block systems arising in the incompressible Navier-Stokes equations. The lid-driven cavity and flow over a finite flat plate are chosen as the benchmark problems. For each problem the Reynolds number varies from a low to the limiting number for a laminar flow.

AMS subject classifications: 65F10, 65F08

Key words: incompressible Navier-Stokes equations, two-by-two block systems, preconditioners.

1. Introduction

In this paper we deal with efficient solution of the stationary, laminar incompressible Navier-Stokes equations, discretized by the Finite Element (FE) method. Due to the presence of the convective term, the Navier-Stokes problem is nonlinear and a suitable linearization technique is needed, like the Picard or Newton method [14]. Both linearizations result in solving a sequence of linear systems with the two-by-two block structure. Finding the solution of the linear system is the most time-consuming part of the numerical simulations. Taking into account the high consumption of the computational time and the memory storage by using the direct solution method, the Krylov subspace methods [1, 14, 31] become feasible to solve the large scale linear systems. It is widely recognized that preconditioning is the most critical ingredient in the development of efficient and reliable Krylov subspace methods.

In the past decades a number of preconditioners are proposed for the two-by-two block systems arising in the incompressible Navier-Stokes equations. In this paper we put our eyes on the block preconditioners, that are constructed by approximating the

*Corresponding author. *Email addresses:* X.He-1@tudelft.nl (X. He), C.Vuik@tudelft.nl (C. Vuik)

block factorization of the coefficient matrix and making full use of the available information about the geometry and physics of the problem. Many state-of-the-art block preconditioners have been devised, for example the SIMPLE-type preconditioners [20,35], the Pressure-Convection-Diffusion commutator (PCD) [18], the Least Squares commutator (LSC) [12], the augmented Lagrangian preconditioner (AL) [6] and the Gradient-Divergence ('grad-div') preconditioner [9, 17]. For an overview of block preconditioners, we refer to [5, 8, 32, 34].

Nowadays the AL and 'grad-div' preconditioners gain a lot of attention. In order to overcome the bottleneck of the AL preconditioner and make it more efficient, a modified version is devised [7]. The analysis between the AL and 'grad-div' preconditioners in this paper illustrate that the strategy leading to the modified AL preconditioner is still applicable in the 'grad-div' preconditioner. In this way, a modified variant of the 'grad-div' preconditioner is realised in this paper. The SIMPLE-type preconditioner remains attractive due to its simplicity and efficiency, and are widely utilised by engineers to solve the industry applications [19]. The improvements of the SIMPLE-type preconditioner are considered in this paper and the focus is on the numerical reliability and efficiency. A comparison between the three preconditioners with their variants is carried out on some academic benchmark problems in this paper. Numerical experiments show that all the improvements advanced in this work are successful, and the modified 'grad-div' preconditioner is the most efficient in terms of the computational time and memory storage.

The organization of the paper is as follows. In Section 2 we briefly state the problem formulation and the Newton and Picard linearization methods. The SIMPLER, augmented Lagrangian, 'grad-div' preconditioners and their variants are introduced in Section 3. Section 4 contains numerical illustrations and some conclusions are given in Section 5.

2. Problem formulation and linearization

A mathematical model for the incompressible flows reads as follows:

$$\begin{aligned} &-\nu\Delta\mathbf{u}+(\mathbf{u}\cdot\nabla)\mathbf{u}+\nabla p=\mathbf{f} && \text{on } \Omega,\\ &\nabla\cdot\mathbf{u}=0 && \text{on } \Omega,\\ &\mathbf{u}=\mathbf{g} && \text{on } \partial\Omega_D,\\ &\nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}}-\mathbf{n}p=0 && \text{on } \partial\Omega_N. \end{aligned} \tag{2.1}$$

Here $\mathbf{u}$ is the velocity, p is the pressure and the positive coefficient ν is the kinematic viscosity, assumed here to be constant. Here Ω is a bounded and connected domain $\Omega\subset\mathbb{R}^d$ $(d=2,3)$, and $\partial\Omega=\partial\Omega_D\cup\partial\Omega_N$ is its boundary, where $\partial\Omega_D$ and $\partial\Omega_N$ denote the parts of the boundary where Dirichlet and Neumann boundary conditions for $\mathbf{u}$ are imposed, respectively. The terms $\mathbf{f}:\Omega\to\mathbb{R}^d$ and $\mathbf{g}$ are a given force field and Dirichlet

boundary data for the velocity. The term **n** denotes the outward-pointing unit normal to the boundary.

For the weak formulation of the stationary Navier-Stokes equations (2.1), we define the approximate solution and test spaces for the velocity as

$$\mathbf{H}^1_E = \Big\{\mathbf{u} \in \mathcal{H}^1(\Omega)^d | \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega_D\Big\},$$
$$\mathbf{H}^1_{E_0} = \Big\{\mathbf{v} \in \mathcal{H}^1(\Omega)^d | \mathbf{v} = \mathbf{0} \quad \text{on } \partial\Omega_D\Big\},$$
$$\mathcal{H}^1(\Omega)^d = \Big\{u_i : \Omega \to \mathbb{R}^d \mid u_i, \frac{\partial u_i}{\partial x_j} \in L_2(\Omega), i, j = 1, \cdots, d\Big\},$$

and for the pressure as

$$L_2(\Omega) = \Big\{p : \Omega \to \mathbb{R} \mid \int_\Omega p^2 < \infty\Big\}.$$

Then the weak formulation reads as follows:
Find $\mathbf{u} \in \mathbf{H}^1_E$ and $\mathbf{p} \in L_2(\Omega)$ such that

$$\nu \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} d\Omega + \int_\Omega (\mathbf{u} \cdot \nabla\mathbf{u})\mathbf{v} d\Omega - \int_\Omega p\nabla \cdot \mathbf{v} d\Omega = \int_\Omega \mathbf{f}\mathbf{v} d\Omega, \tag{2.2a}$$
$$\int_\Omega q\nabla \cdot \mathbf{u} d\Omega = 0, \tag{2.2b}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and all $q \in L_2(\Omega)$. The pressure is uniquely defined only up to a constant term. To make it unique, one usually imposes an additional constraint $\int_\Omega p\, d\Omega = 0$. We also assume that the discretization is done using a stable FE pair, satisfying the LBB condition [14].

The nonlinearity of the considered problem is handled by a linearization method. Two well-known and most often used linearization methods are the Newton and Picard methods [14], briefly introduced below.

Let $(\mathbf{u}_0, p_0)$ be an initial guess and let $(\mathbf{u}_k, p_k)$ be the approximate solutions at the kth nonlinear step. Then we update the velocity and the pressure on the $(k+1)$ step as $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta\mathbf{u}_k$, $p_{k+1} = p_k + \delta p_k$ for $k = 0, 1, \cdots$ until convergence, where $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ (provided $\mathbf{u}_k \in \mathbf{H}^1_E$ and $p_k \in L_2(\Omega)$). Substituting $\mathbf{u}_{k+1}$ and p_{k+1} into the weak formulation (**??**), the correction $(\delta\mathbf{u}_k, \delta p_k)$ should satisfy the following problem: Find $\delta\mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\nu \int_\Omega \nabla\delta\mathbf{u}_k : \nabla\mathbf{v} d\Omega + \int_\Omega (\mathbf{u}_k \cdot \nabla\delta\mathbf{u}_k) \cdot \mathbf{v} d\Omega + \int_\Omega (\delta\mathbf{u}_k \cdot \nabla\mathbf{u}_k) \cdot \mathbf{v} d\Omega$$
$$- \int_\Omega \delta p_k\, (\nabla \cdot \mathbf{v}) d\Omega = R_k \tag{2.3a}$$
$$\int_\Omega q\, (\nabla \cdot \delta\mathbf{u}_k) d\Omega = P_k, \tag{2.3b}$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and $q \in L_2(\Omega)$. The residual terms are obtained as

$$R_k = \int_\Omega \mathbf{f} \cdot \mathbf{v} d\Omega - \nu \int_\Omega \nabla \mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega - \int_\Omega (\mathbf{u}_k \cdot \nabla \mathbf{u}_k) \cdot \mathbf{v} d\Omega + \int_\Omega p_k (\nabla \cdot \mathbf{v}) d\Omega, \quad (2.4a)$$

$$P_k = -\int_\Omega q \, (\nabla \cdot \mathbf{u}_k) d\Omega. \quad (2.4b)$$

This procedure is refereed to as the so-called the Newton linearization method.

The Picard linearization is obtained in a similar way as the Newton method, except that the term $\int_\Omega (\delta \mathbf{u}_k \cdot \nabla \mathbf{u}_k) \cdot \mathbf{v} d\Omega$ is dropped. Thus, the linear problem in the Picard method reads as follows: Find $\delta \mathbf{u}_k \in \mathbf{H}^1_{E_0}$ and $\delta p_k \in L_2(\Omega)$ such that

$$\nu \int_\Omega \nabla \delta \mathbf{u}_k : \nabla \mathbf{v} d\Omega + \int_\Omega (\mathbf{u}_k \cdot \nabla \delta \mathbf{u}_k) \cdot \mathbf{v} d\Omega - \int_\Omega \delta p_k \, (\nabla \cdot \mathbf{v}) d\Omega = R_k, \quad (2.5a)$$

$$\int_\Omega q \, (\nabla \cdot \delta \mathbf{u}_k) d\Omega = P_k, \quad (2.5b)$$

for all $\mathbf{v} \in \mathbf{H}^1_{E_0}$ and $q \in L_2(\Omega)$. Similarly, we update the approximations as $\mathbf{u}_{k+1} = \mathbf{u}_k + \delta \mathbf{u}_k$ and $p_{k+1} = p_k + \delta p_k$ for $k = 0, 1, \cdots$ until convergence.

The Newton iteration method gains its popularity due to its locally quadratical convergence. This quadratical convergence is guaranteed if the initial estimation of the unknown is sufficiently close to the solution [14]. In order to achieve the fast convergence of the Newton method, the *continuation* method is utilised in this paper. On a fixed mesh, we carry out numerical simulations by using the computed solution from a lower Reynolds number as an initial guess for the next simulation with a higher Reynolds number. The *continuation* method is also used in the Picard method.

3. Preconditioning techniques

Let $\mathbf{X}^h_{E_0}$ and P^h be finite dimensional subspaces of $\mathbf{H}^1_{E_0}$ and $L_2(\Omega)$, and let $\{\vec{\varphi}_i\}_{1 \le i \le n_u}$ be the nodal basis of $\mathbf{X}^h_{E_0}$ and $\{\phi_i\}_{1 \le i \le n_p}$ be the nodal basis of P^h. According to the Galerkin framework, the discrete corrections of the velocity and pressure are represented as

$$\delta \mathbf{u}_h = \sum_{i=1}^{n_u} \delta \mathbf{u}_i \vec{\varphi}_i, \quad \delta \mathbf{p}_h = \sum_{i=1}^{n_p} \delta p_i \phi_i,$$

where n_u and n_p are the total number of degrees of freedom for the velocity and pressure. The linear systems arising in the Newton and Picard linearizations are of the form

$$\begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_h \\ \delta \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{A}\mathbf{x} = \mathbf{b}, \quad (3.1)$$

where the system matrix

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & O \end{bmatrix}$$

is nonsymmetric due to the non-symmetry of A and of a two-by-two block form. The matrix $B \in \mathbb{R}^{n_p \times n_u}$ corresponds to the (negative) divergence operator and B^T corresponds to the gradient operator (e.g., [14]). Here we assume that the LBB condition is satisfied, otherwise, the $(2,2)$ block becomes nonzero since some stabilization is required. When comparing the Newton and Picard linearization methods, the difference appears in the pivot block $A \in \mathbb{R}^{n_u \times n_u}$, which is of the form $A = A_\nu + N + \delta_1 \widehat{N}$. Given the approximation $\mathbf{u}_h$, the entries of A_ν, N and $\widehat{N}$ are

$$A_\nu \in \mathbb{R}^{n_u \times n_u}, \quad [A_\nu]_{i,j} = \nu \int_\Omega \nabla \vec{\varphi}_i : \nabla \vec{\varphi}_j, \tag{3.2a}$$

$$N \in \mathbb{R}^{n_u \times n_u}, \quad [N]_{i,j} = \int_\Omega (\mathbf{u}_h \cdot \nabla \vec{\varphi}_j) \vec{\varphi}_i, \tag{3.2b}$$

$$\widehat{N} \in \mathbb{R}^{n_u \times n_u}, \quad [\widehat{N}]_{i,j} = \int_\Omega (\vec{\varphi}_j \cdot \nabla \mathbf{u}_h) \vec{\varphi}_i. \tag{3.2c}$$

The Newton method corresponds to $\delta_1 = 1$, while the Picard method corresponds to $\delta_1 = 0$. Also, in the Picard linearization, the velocity pivot block A is of a block diagonal form. However, this block diagonal structure does not hold in the Newton linearization due to the presence of $\widehat{N}$. The linear system (3.1) arising in the Newton (**??**) or Picard method (**??**) is often referred to as the Oseen problem.

Finding the solution of the linear system (3.1) is the kernel and most time-consuming part in the numerical simulations. Therefore, fast and reliable solution techniques are critical. In this paper Krylov subspace iterative solution methods [14, 31] accelerated by numerically and computationally efficient preconditioners are utilised.

Linear systems of the form (3.1) are in a two-by-two block form, and how to precondition such systems have been intensively studied. In this work we limit ourselves to preconditioners, based on approximate block factorizations of the original matrix. The literature on this class of preconditioners is huge. We refer for more details to the articles [2–4, 21, 26], the surveys [5, 8, 32, 34] and the books [14, 31], with numerous references therein. In general, the exact factorization of a two-by-two block matrix is

$$\begin{aligned} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} &= \begin{bmatrix} A_{11} & O \\ A_{21} & S \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1} A_{12} \\ O & I_2 \end{bmatrix} \\ &= \begin{bmatrix} I_1 & O \\ A_{21} A_{11}^{-1} & I_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ O & S \end{bmatrix}, \end{aligned} \tag{3.3}$$

where I_1 and I_2 are identity matrices of proper dimensions. The pivot block A_{11} is assumed to be nonsingular and

$$S = A_{22} - A_{21} A_{11}^{-1} A_{12}$$

is the exact Schur complement matrix. In our case (3.1), $A_{11} = A$, $A_{12} = B^T$, $A_{21} = B$ and $A_{22} = O$. So, $S = -BA^{-1}B^T$.

As preconditioners for such matrices of two-by-two block form, approximate block-factorization and block lower- or upper-triangular approximate factors are often used

$$\begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{S} \end{bmatrix}, \qquad \begin{bmatrix} I_1 & \widetilde{A}_{11}^{-1} A_{12} \\ O & I_2 \end{bmatrix}, \tag{3.4}$$

$$\begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{S} \end{bmatrix}, \qquad \begin{bmatrix} \widetilde{A}_{11} & A_{12} \\ O & \widetilde{S} \end{bmatrix}. \tag{3.5}$$

Here the matrix $\widetilde{A}_{11}$ denotes some approximation of A_{11}, given either in an explicit form or implicitly defined via an inner iterative solution method with a proper stopping tolerance. The matrix $\widetilde{S}$ is some approximation of the exact Schur complement S.

Comparing to the approximation of the pivot velocity block A_{11}, the most challenging task turns out to be the construction of approximations of the Schur complement S, which is in general dense and it is not practical to form it explicitly. For the two-by-two block system arising in the incompressible Navier-Stokes equations, several state-of-the-art approximations of the Schur complement are proposed and analysed, c.f., [6, 7, 12, 13, 15, 16, 18, 20, 23, 27, 29, 30, 35]. In this paper we choose the SIMPLER, augmented Lagrangian and 'grad-div' preconditioners for study and furthermore propose improved versions of those.

3.1. The SIMPLER preconditioner

SIMPLE (Semi-Implicit Pressure Linked Equation) is used by Patanker [28] as an iterative method to solve the Navier-Stokes problem. The scheme belongs to the class of basic iterative methods and exhibits slow convergence. Vuik et al [20, 35] use SIMPLE and its variant SIMPLER as a preconditioner in a Krylov subspace method to solve the incompressible Navier-Stokes equations, achieving in this way, a much faster convergence. SIMPLE and SIMPLER rely on an approximate block-factorization of the saddle point matrices and due to their simplicity, remain attractive preconditioning techniques. We briefly describe both formulations for the Oseen problem $\mathcal{A}$ in (3.1).

The SIMPLE preconditioner $\mathcal{P}_{SIMPLE}$ reads:

$$\mathcal{P}_{SIMPLE} = \begin{bmatrix} A & O \\ B & \widetilde{S} \end{bmatrix} \begin{bmatrix} I_1 & D^{-1}B^T \\ O & I_2 \end{bmatrix},$$

where D is the diagonal of the block A and $\widetilde{S} = -BD^{-1}B^T$. Solutions of systems with $\mathcal{P}_{SIMPLE}$ are straightforward, namely,

Algorithm SIMPLE

Given $\mathbf{y} = [\mathbf{y}_u; \mathbf{y}_p]$, $\mathbf{x} = \mathcal{P}_{SIMPLE}^{-1}\mathbf{y}$ is found within the following steps.

Step 1: Solve $A\mathbf{x}_u^\star = \mathbf{y}_u$

Step 2: Solve $\widetilde{S}\mathbf{x}_p = \mathbf{y}_p - B\mathbf{x}_u^\star$

Step 3: Compute $\mathbf{x}_u = \mathbf{x}_u^\star - D^{-1}B^T\mathbf{x}_p$

Step 4: Set $\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_p]$

The SIMPLER preconditioner $\mathcal{P}_{SIMPLER}$ differs slightly from the SIMPLE one. It includes a pressure prediction step and reads as

Algorithm SIMPLER

Given $\mathbf{y} = [\mathbf{y}_u; \mathbf{y}_p]$, $\mathbf{x} = \mathcal{P}_{SIMPLER}^{-1}\mathbf{y}$ is found within the following steps.

Step 0: Solve $\widetilde{S}\mathbf{x}_p^\star = \mathbf{y}_p - BD^{-1}\mathbf{y}_u$

Step 1: Solve $A\mathbf{x}_u^\star = \mathbf{y}_u - B^T\mathbf{x}_p^\star$

Step 2: Solve $\widetilde{S}\delta\mathbf{x}_p = \mathbf{y}_p - B\mathbf{x}_u^\star$

Step 3: Update $\mathbf{x}_p = \mathbf{x}_p^\star + \delta\mathbf{x}_p$ and $\mathbf{x}_u = \mathbf{x}_u^\star - D^{-1}B^T\delta\mathbf{x}_p$

Step 4: Set $\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_p]$

Some improvements related to the SIMPLE/SIMPLER preconditioners proposed in this paper are presented as follows.

1. Numerical experiments show that at high Reynolds numbers, D=diag(A) can become very small in certain elements leading to sudden divergence. Therefore, the velocity block A is approximated by a diagonal matrix $\widetilde{D} = \sum |A|$ in this paper, where $\sum |A|$ denotes the row sum of absolute values of A. In this paper $\mathcal{P}_{SIMPLE}$ and $\mathcal{P}_{SIMPLER}$ with $\widetilde{D} = \sum |A|$ are referred to as the ideal SIMPLE and SIMPLER preconditioners.

2. In the Newton linearization the velocity pivot block A is not of a block diagonal form. For example, $A := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ in the two dimensional case. We can approximate A by

$$\widetilde{A} = \begin{bmatrix} A_{11} & O \\ A_{21} & A_{22} \end{bmatrix},$$

and replace it with $\widetilde{A}$ in step 1 of the two algorithms above. The SIMPLE and SIMPLER preconditioners with $\widetilde{A}$ and $\widetilde{D} = \sum |\widetilde{A}|$ is referred to as the modified SIMPLE and SIMPLER preconditioner in this paper. Since A is already of block diagonal structure in the

Picard linearization, the strategy leading to the modified SIMPLE and SIMPLER preconditioners is not necessary. Numerical results in Section 4 illustrate that compared to the ideal SIMPLE/SIMPLER preconditioners, the average number of the linear iterations is less or nearly the same by using the modified ones, depending on testing problems. Thus, the computational time is reduced. To more clearly see this, we assume that the computational complexity of iteratively solving a system with $A \in \mathbb{R}^{n_u \times n_u}$ is $O(n_u^2)$. Then, the complexity of solving systems with A_{11} and A_{22} is $O(n_u^2/2)$. In Fig. 1 we plot the eigenvalues of the preconditioned matrix $\mathcal{P}_{SIMPLE}^{-1}\mathcal{A}$ by using the ideal and modified SIMPLE preconditioners. As seen, the modified SIMPLE preconditioner products a similar spectrum compared to the ideal one. This explains the numerical results that the average number of the linear iterations is not increased by using the modified SIMPLE/SIMPLER preconditioners. Theoretical analysis of the modified SIMPLE/SIMPLER preconditioners is considered to be one research direction in future.

3. In the ideal and modified SIMPLE/SIMPLER preconditioners, one needs to solve subsystems with A, A_{11} and A_{22}. In this paper we test an algebraic multigrid (AMG) solver. However, this AMG solver turns out to be less and less efficient when increasing the Reynolds number. The reason is that these sunblocks become far away from symmetry and diagonal dominance due to the strong effect of the convection term, especially for high Reynolds numbers. The AMG is designed to work efficiently for diagonally dominant matrices, ideally symmetric and positive definite matrices. In order to make the AMG solver work well for high Reynolds numbers, we change these blocks by adding $h^2 I$ to them, where h is the mesh size and I denotes the identity matrix with a proper size. This perturbation makes these blocks more diagonally dominant and the AMG solver is expected to work efficiently for diagonally dominant matrices. This perturbation is only used in preconditioners, not in the coefficient matrix.

3.2. The augmented Lagrangian method

In the so-called augmented Lagrangian approach, we first transform the linear system (3.1) into an equivalent one with the same solution, which is of the form

$$\begin{bmatrix} A+\gamma B^T W^{-1} B & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{f}} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{A}_{AL}\mathbf{x} = \widehat{\mathbf{b}}, \tag{3.6}$$

where $\widehat{\mathbf{f}} = \mathbf{f} + \gamma B^T W^{-1}\,\mathbf{g}$, and $\gamma > 0$ and W are suitable scalar and matrix parameters. Clearly, the transformed system (3.6) has the same solution as (3.1) for any value of γ and any nonsingular matrix W.

The equivalent system (3.6) is what we intend to solve and the AL-type preconditioner proposed for $\mathcal{A}_{AL}$ in (3.6) is of a block lower-triangular form

$$\mathcal{M}_{AL} = \begin{bmatrix} A+\gamma B^T W^{-1} B & O \\ B & -\frac{1}{\gamma} W \end{bmatrix}. \tag{3.7}$$

To distinguish from the modified AL preconditioner introduced later, the preconditioner $\mathcal{M}_{AL}$ is referred to as the ideal AL preconditioner [6]. It can be seen that the exact Schur complement

$$S_{\mathcal{A}_{AL}} = -B(A + \gamma B^T W^{-1} B)^{-1} B^T$$

of the transformed matrix $\mathcal{A}_{AL}$ is approximated by $-\frac{1}{\gamma}W$. As analysed in [6, 16], for any non-singular matrix W the eigenvalues of the preconditioned matrix $\mathcal{M}_{AL}^{-1}\mathcal{A}_{AL}$ will cluster to one with large values of γ. This means that large γ results in a few iterations for any W if subsystems with

$$A_{AL} := A + \gamma B^T W^{-1} B$$

are solved accurately enough. However, A_{AL} will be continually ill-conditioned by increasing γ, and finding solutions of systems with it turns out to be more difficult. Therefore, a nature choice of $\gamma = 1$ or $O(1)$ has been used in the numerical tests in many studies, for example [6, 16]. In this paper $\gamma = 1$ is chosen in the ideal AL preconditioner.

Efficient solution of the system with the modified pivot block A_{AL} is still an open question in the AL framework. Although the components A and B are sparse, the modified pivot block A_{AL} is in general much denser. Furthermore, A_{AL} contains discretizations of mixed derivatives, and A_{AL} is not block-diagonal in the Picard and Newton linearizations. Some approximation of the block A_{AL} leads to the modified AL preconditioner as follows. Here we take two dimensions as an example to illustrate the modified AL preconditioner, originally proposed in [7]. The original pivot matrix A is

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

(A_{12} and A_{21} are zero in the Picard linearization) and $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$.

Then the modified pivot block A_{AL} can be written as

$$\begin{aligned} A_{AL} &:= \begin{bmatrix} A_{AL,11} & A_{AL,12} \\ A_{AL,21} & A_{AL,22} \end{bmatrix} \\ &= \begin{bmatrix} A_{11} + \gamma B_1^T W^{-1} B_1 & A_{12} + \gamma B_1^T W^{-1} B_2 \\ A_{21} + \gamma B_2^T W^{-1} B_1 & A_{22} + \gamma B_2^T W^{-1} B_2 \end{bmatrix}, \end{aligned}$$

and its approximation can be obtained by neglecting one of the off-diagonal block

$$\widetilde{A}_{AL} = \begin{bmatrix} A_{AL,11} & O \\ A_{AL,21} & A_{AL,22} \end{bmatrix}.$$

Given the approximation $\widetilde{A}_{AL}$, the modified AL preconditioner is obtained as

$$\widetilde{\mathcal{M}}_{AL} = \begin{bmatrix} A_{AL,11} & O & O \\ A_{AL,21} & A_{AL,22} & O \\ B_1 & B_2 & -\frac{1}{\gamma}W \end{bmatrix}, \tag{3.8}$$

The modified AL preconditioner offers two main advantages compared to the ideal one. When solving systems with $\widetilde{A}_{AL}$ one needs to solve two sub-systems with $A_{AL,11}$ and $A_{AL,22}$. In this way, the size of the linear system to be solved and the computational time are reduced. Besides, there are approximations of mixed derivatives in A_{AL}, i.e., $A_{AL,21}$ and $A_{AL,12}$. This can be an obstacle when applying known solution techniques, such as AMG methods. On the other hand, there exists an optimal value of γ which minimizes the iterations of the Krylov subspace methods by using the modified AL preconditioner. Although there exists some theoretical prediction of the optimal value in [7], the optimal value is problem dependent and expensive to calculate. Therefore, the optimal γ are determined through numerical experiments in this paper.

An other parameter in the AL framework is the matrix parameter W. The transformation (3.6) holds true for any nonsingular matrix W. In practice W is often chosen to be the pressure mass matrix M, or its diagonal–diag(M) [6,7,16]. Theoretical analysis and numerical experiments therein show that the ideal and modified AL preconditioners with $W = M$ or diag(M) are independent of the mesh refinement. In this paper, $W =$ diag(M) is utilised in modified and ideal AL preconditioners.

Improvements related to the modified AL preconditioner are introduced here. In [7] systems with $A_{AL,11}$ and $A_{AL,22}$ involved in the modified AL preconditioner are solved by using the direct solution method, which is not feasible for large simulations in terms of the total solution time and memory storage. In this paper we test an AMG solver with a proper stopping tolerance. In this way we have a deeper insight of the performance of the modified AL preconditioner related to the inner inaccurate solutions of the systems with $A_{AL,11}$ and $A_{AL,22}$. Also, for a high Reynolds number the perturbations to $A_{AL,11}$ and $A_{AL,22}$ are done in a same way as in SIMPLE/SIMPLER preconditioners. The reason is still to make the AMG solver work efficiently when the Reynolds number is high.

More details on the comparison between the modified and ideal AL preconditioners are shown in Section 4.

3.3. The 'grad-div' preconditioner

The augmented Lagrangian framework is also known as 'First-Discretize-Then-Stabilize' technique. There are also other techniques, known as 'First-Stabilize-Then-Discretize', which are briefly described here. Adding a stabilization term $-\gamma\nabla(\nabla\cdot\mathbf{u})$ to the momentum equation in (2.1), one can obtain the so-called 'grad-div' stabilization formulation (cf. e.g., [9,17])

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \gamma\nabla(\nabla\cdot\mathbf{u}) + \nabla p &= \mathbf{f} \quad \text{on } \Omega, \\ \nabla\cdot\mathbf{u} &= 0 \quad \text{on } \Omega. \end{aligned} \tag{3.9}$$

The added term is zero since the velocity is divergence free. The set of the 'grad-div' stabilized Navier-Stokes equations is nonlinear and we still use the Picard and Newton methods to linearize it. After discretizing the linearized equations with stable FE pairs,

the system at each nonlinear step is still of two-by-two block form, namely,

$$\begin{bmatrix} F_{GD} & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_h \\ \delta \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{F}_{GD}\mathbf{x} = \mathbf{b}, \tag{3.10}$$

where the block F_{GD} is of the form $F_{GD} = A_\nu + N + \gamma G + \delta_1 \widehat{N}$. The matrices A_ν, N and $\widehat{N}$ are the same as given in (**??**). The term γG is the discrete operator of $\gamma(\nabla \cdot \delta\mathbf{u}, \nabla \cdot \mathbf{v})$ ($\mathbf{v}$ is the test function).

The preconditioner proposed for $\mathcal{F}_{GD}$ is of the same form as the ideal AL preconditioner

$$\mathcal{M}_{GD} = \begin{bmatrix} F_{GD} & O \\ B & -\frac{1}{\gamma}W \end{bmatrix}. \tag{3.11}$$

In this paper the preconditioner $\mathcal{M}_{GD}$ is referred to as the ideal 'grad-div' preconditioner. So far, the quite often used choices of W and γ has been the pressure mass matrix M (or diag(M)) and $\gamma = 1$ in [9,17]. In this paper $W = \text{diag}(M)$ and $\gamma = 1$ are chosen in the ideal 'grad-div' preconditioner.

The difference between the AL and 'grad-div' methods is in the velocity pivot block. It is clear that F_{GD} is sparser than A_{AL} since F_{GD} arises in the discretization of an operator (also noted in [9]). On the other hand, G is analogous to the matrix $B^T B$. Thus, the matrix F_{GD} is analogous to A_{AL} with W being the identity matrix. This indicates that the means to tune the 'grad-div' stabilization is only the constant γ, while in the AL framework we possess γ and W to play with.

In this paper we improve the ideal 'grad-div' preconditioner by using the same strategy which leads to the modified AL preconditioner. Here, we still take two dimensions as an example. The pivot block is defined as

$$F_{GD} := \begin{bmatrix} F_{GD,11} & F_{GD,12} \\ F_{GD,21} & F_{GD,22} \end{bmatrix},$$

which is not block diagonal in both the Picard and Newton linearizations due to the presence of the matrix G. We approximate it as

$$\widetilde{F}_{GD} = \begin{bmatrix} F_{GD,11} & O \\ F_{GD,21} & F_{GD,22} \end{bmatrix}$$

and the modified 'grad-div' preconditioner is obtained as

$$\widetilde{\mathcal{M}}_{GD} = \begin{bmatrix} F_{GD,11} & O & O \\ F_{GD,21} & F_{GD,22} & O \\ B_1 & B_2 & -\frac{1}{\gamma}W \end{bmatrix}, \tag{3.12}$$

We still choose $W = \text{diag}(M)$ in the modified 'grad-div' preconditioner. Numerical experiments show that the performance of the modified 'grad-div' preconditioner is dependent of the parameter γ. In this paper we use numerical experiments to determine

the optimal γ. Theoretical prediction on the effect of γ is an on-going research by the authors.

Inaccurate solutions of systems with $F_{GD,11}$ and $F_{GD,22}$ are obtained by using the AMG solver with a proper stopping tolerance. Perturbations to $F_{GD,11}$ and $F_{GD,22}$, as introduced before are necessary when the Reynolds number is high. We explore the performance of the ideal and modified 'grad-div' preconditioners and more details are presented in Section 4.

4. Numerical experiments

In this section we present numerical results by using these three preconditioners and their variants as introduced in Section 3. The academic test problems are the lid-driven cavity and the flow over a finite flat plate. When the Newton or Picard linearization method is used, the number of nonlinear iterations is refereed to as the Newton or Picard iterations. The stopping tolerance for the nonlinear solver is relative and chosen to be 10^{-10}. Such a small tolerance is used in this paper in order to exhibit the efficiency of the linear and nonlinear solvers. In practice a larger number of the tolerance can be sufficient, for example 10^{-6}. In order to achieve a fast convergence rate of the nonlinear solver, the *continuation* method is utilised. This method means that in the Newton and Picard linearization methods, the computed solution from a lower Reynolds number is given as a 'good' initial guess for the next simulation with a higher Reynolds number. To solve the linear system at each nonlinear iteration, the Generalized Conjugate Residual method (`GCR`) [11] is used since it allows variable preconditioners. The relative stopping tolerance for `GCR` is chosen to be 10^{-2} and the number of iterations is denoted as `GCR` iterations. The reported `GCR` iterations are averaged over the total number of nonlinear iterations.

Besides a direct solution method, in this paper we also test an aggregation based algebraic multigrid method– `agmg` (see [22,24,25]) to solve the subsystems in these three preconditioners. The implementation of `agmg` is in `Fortran` and a `Matlab` interface is provided. Its performance in terms of CPU time is comparable with that of 'backslash' sparse direct solver in `Matlab`. For nonsymmetric matrices `agmg` uses the `GCR` method accelerated by the multigrid preconditioner. For systems with symmetric and positive definite matrices the conjugate gradient (`CG`) method is chosen in `agmg`. When the `agmg` solver is used to solve the subsystems corresponding to the velocity and the pressure unknowns, the relative stopping tolerance is denoted as $\epsilon_{\texttt{agmg},u}$ and $\epsilon_{\texttt{agmg},p}$.

All experiments in this paper are carried out by using `Matlab` 7.13 (R2011b), and performed on a Linux-64 platform with 4 Intel(R) Core i5 CPUs, 660@3.33GHz. All reported execution times are in seconds. Whenever `agmg` is used, its setup time is included in the reported times.

The finite element pair used in this paper is Q2-Q1. The flow Reynolds number is defined by $Re = UL/\nu$, where U is the reference velocity, and L is the reference length of the computational domain. For each test problem, stretched grids are used. The stretching function is given in every test problem.

Table 1: LDC: Picard and average GCR iterations by using the ideal AL, 'grad-div' and SIMPLER preconditioners. Sub-systems are solved directly.

Re	100	400	1000	2500	5000
Grid	Picard / GCR iterations				
64^2					
AL:	14/2	29/2	32/2	57/2	stagnation
GD:	14/4	28/3	31/3	51/3	stagnation
SIMPLER:	15/23	28/32	32/30	58/29	stagnation
128^2					
AL:	13/2	28/2	31/2	48/2	308/2
GD:	14/4	27/3	30/3	50/3	282/3
SIMPLER:	17/32	32/41	32/41	48/42	284/42
256^2					
AL:	13/2	26/2	28/2	45/2	202/2
GD:	14/4	26/3	29/3	50/3	218/3
SIMPLER:	19/40	41/58	52/58	55/62	212/62

4.1. Lid Driven Cavity (LDC)

The first benchmark problem is the two-dimensional lid-driven cavity problem, equipped with the boundary conditions $u_1 = u_2 = 0$ for $x = 0, x = 1$ and $y = 0$; $u_1 = 1, u_2 = 0$ for $y = 1$. The reference velocity and the reference length are chosen as $U = 1$ and $L = 1$. Thus, the Reynolds number is $Re = \nu^{-1}$. The stretched grids are generated based on the uniform Cartesian grids with $n \times n$ cells. The stretching function is applied in both directions with parameters $a = 1/2$ and $b = 1.1$ [19]

$$x = \frac{(b+2a)c - b + 2a}{(2a+1)(1+c)}, \quad c = \left(\frac{b+1}{b-1}\right)^{\frac{\bar{x}-a}{1-a}}, \quad \bar{x} = 0, \frac{1}{n}, \frac{2}{n}, \cdots, 1.$$

The results in Tables 1-4 are obtained by solving subsystems in these three preconditioners through the direct solution method. The direct solution method is not practical in large scale simulations due to its limitations in solution time and memory requirements. The reason by using it is that we want to illustrate the 'best' performance of these preconditioners, without effected by inaccurate solutions of subsystems computed via an iterative solution method with a proper stopping tolerance.

The average number of GCR iterations by using the ideal AL and 'grad-div' preconditioners is fully independent of the mesh size and the Reynolds number in both the Newton and Picard linearization methods, see Tables 1-2. Results in Table 3 show that in the Picard linearization, the modified AL and 'grad-div' preconditioners are independent of the mesh size and the Reynolds number too. In the Newton linearization, the modified AL and 'grad-div' preconditioners are independent of the mesh size but dependent of the Reynolds number, see Table 4. We see from Table 4 that in the Newton linearization the number of GCR iterations by using the modified AL preconditioner is

Table 2: LDC: Newton and average GCR iterations by using the ideal AL, 'grad-div' and SIMPLER preconditioners. Sub-systems are solved directly.

Re	100	400	1000	2500	5000
Grid	Newton / GCR iterations				
64^2					
AL:	5/2	7/2	7/2	7/3	6/3
GD:	7/6	7/6	8/6	7/6	7/6
SIMPLER:	7/35	8/50	9/72	9/108	10/167
128^2					
AL:	5/2	7/2	6/2	6/3	6/3
GD:	7/6	8/6	7/6	8/6	7/6
SIMPLER:	7/56	8/66	9/90	9/140	10/232
256^2					
AL:	5/2	7/2	6/2	6/3	6/3
GD:	7/6	8/6	7/6	7/6	7/6
SIMPLER:	8/73	8/102	11/122	8/174	10/251

Table 3: LDC: Picard and average GCR iterations by using the modified AL and 'grad-div' preconditioners with optimal γ. Sub-systems are solved directly.

Re	100	400	1000	2500	5000
Grid	Picard / GCR iterations (optimal γ)				
64^2					
AL:	14/5(0.02)	29/9(0.01)	35/11(0.008)	55/9(0.006)	stagnation
GD:	14/7(0.06)	27/11(0.04)	32/15(0.02)	66/14(0.01)	stagnation
128^2					
AL:	14/5(0.02)	28/9(0.01)	33/11(0.008)	45/9(0.006)	306/14(0.004)
GD:	13/7(0.06)	27/11(0.04)	31/16(0.02)	55/15(0.01)	296/18(0.01)
256^2					
AL:	14/5(0.02)	28/9(0.01)	33/10(0.008)	45/10(0.006)	202/12(0.004)
GD:	13/7(0.06)	26/10(0.04)	30/15(0.02)	55/14(0.01)	218/17(0.01)

40, which is acceptable for the 'worst' situation with $Re = 5000$ and 256^2 grids. The optimal value of γ involved in the modified AL and 'grad-div' preconditioners is independent of the mesh refinement. Therefore, we can carry out numerical experiments on a coarse grid to determine the optimal γ, then use this choice also on a fine mesh. On a coarse mesh the user does not need to pay so many efforts to find the precisely optimal γ, since relatively small differences from the precisely optimal γ do not result in a significant change of the number of GCR iterations.

From Tables 3-4, we see that the optimal value of γ for the modified AL and 'grad-div' preconditioners changes with varying Reynolds numbers. We see that for the modified AL preconditioner the optimal value of γ lies in the interval $[0.004,\ 0.02]$ in the

Table 4: LDC: Newton and average GCR iterations by using the modified AL, 'grad-div' preconditioners with optimal γ and the modified SIMPLER preconditioner. Sub-systems are solved directly.

Re	100	400	1000	2500	5000
Grid	Newton / GCR iterations (optimal γ)				
64^2					
AL:	6/7(0.04)	7/14(0.04)	7/21(0.02)	9/34(0.01)	9/45(0.01)
GD:	6/10(0.06)	7/16(0.04)	7/27(0.04)	8/41(0.02)	8/55(0.02)
SIMPLER:	8/34	9/43	11/47	17/63	21/67
128^2					
AL:	6/7(0.04)	7/15(0.04)	7/22(0.02)	8/33(0.01)	9/43(0.01)
GD:	6/10(0.06)	7/17(0.04)	7/29(0.04)	8/45(0.02)	8/64(0.02)
SIMPLER:	9/52	9/71	13/67	18/64	26/71
256^2					
AL:	6/7(0.04)	6/15(0.04)	7/22(0.02)	8/34(0.01)	9/40(0.01)
GD:	6/10(0.06)	6/17(0.04)	7/29(0.04)	8/45(0.02)	8/65(0.02)
SIMPLER:	9/70	10/112	14/99	16/94	31/77

Picard linearization, and in $[0.01,\ 0.04]$ in the Newton linearization. For the modified 'grad-div' preconditioner, the optimal γ lies in the interval $[0.01,\ 0.06]$ in the Picard linearization, and in $[0.02,\ 0.06]$ in the Newton linearization. The meaning of finding out these intervals is that in practice they can help us to 'easily' choose a good γ, although these intervals containing the optimal values of γ are problem dependent.

The ideal SIMPLER preconditioner is independent of the Reynolds number in the Picard linearization, but depends on the Reynolds number in the Newton linearization, see Tables 1-2. Also, the Newton linearization results in more GCR iterations by using the ideal SIMPLER preconditioner, compared to the Picard linearization. The reason is that in the SIMPLER preconditioner, the approximation of the velocity sub-block is needed and it is taken as a diagonal matrix with the row sum of its absolute values in the corresponding diagonal positions. In the Newton linearization, the velocity sub-block is more complicated and this type of approximation does not work as well as in the Picard linearization. Independence of the mesh size by using the ideal SIMPLER preconditioner is not obtained in both the Picard and Newton linearizations, see Tables 1-2. Results in Table 4 show that in the Newton linearization, the number of GCR iterations by using the modified SIMPLER preconditioner is independent of the Reynolds number, and is also much less than that by using the ideal SIMPLER preconditioner. This result shows that the modified SIMPLER preconditioner is more efficient in terms of total solution time, compared to the ideal SIMPLER preconditioner in this test problem.

As mentioned already, the strategy leading to the modified SIMPLER preconditioner is only valid in the Newton linearization, not in the Picard linearization. This is the reason why we only present the modified AL and 'grad-div' preconditioners in Table 3.

For the Picard linearization it is reasonable to compare the modified AL and 'grad-div' preconditioners with the ideal SIMPLER preconditioner, as in Table 5.

Results in Tables 5-6 are obtained by using agmg for subsystems. As seen, the modified 'grad-div' preconditioner is most efficient in terms of total computational time, followed by the modified AL preconditioner. We see from Tables 5-6 that the number of GCR iterations by using the modified 'grad-div' preconditioner is more than that with the modified AL preconditioner. As mentioned already, the modified 'grad-div' preconditioner gains its superiority due to its relatively sparse structure. agmg is expected to work more efficiently for sparse matrices, and this pays off the more GCR iterations.

For a higher Reynolds number, such as $Re \geq 2500$, agmg fails for the velocity subsystems in these three preconditioners. It takes more than 1000 agmg iterations to converge to the desired accuracy. The reason is that for a large Reynolds number, the convection term is dominant and the diffusion term is weak. The velocity sub-block turns to be far away from diagonal dominance. agmg is designed to work well for diagonally dominant matrices, such as Laplacian matrix. In order to make agmg work, we perturb the velocity sub-blocks in the three preconditioners. This perturbation is done by adding $h^2 I$ to them, where h is the mesh size and I denotes the identity matrix with a proper size. By using this perturbation the sub-blocks become more diagonally dominant in those 'difficult' situations ($Re \geq 2500$) and agmg can work. This perturbation is only used in the preconditioners, not in the coefficient matrices. On the other hand, this perturbation changes the original preconditioner and its efficiency can be attenuated. When the perturbation is applied, more GCR iterations are needed. The results marked by $\star$ in Tables 5-6 are obtained by adding this perturbation. In Table 6 this perturbation starts from $Re = 400$ for the modified SIMPLER preconditioner.

Efficient solvers for subsystems are crucial. The performance of these three preconditioners can be quite different by using different inner solvers for subsystems involved therein. Improving some already known solution methods, such as agmg, for velocity subsystems with a high Reynolds number is an on-going research.

There are many references testing these three preconditioners and their variants on the LDC problem, c.f. [7, 9, 16, 17, 19]. It is not easy to compare the results given in this paper and other references, because different settings are used in different papers. For the ideal and modified AL preconditioners, the number of linear iterations given in this paper is close to that in [7, 16] when the same settings are used, such as the same stopping tolerance, the same mesh and the same inner solver for the subsystems. For the ideal and modified 'grad-div' preconditioner, we compare our results with [9, 17]. The same settings also result in a comparable result, in terms of the number of linear iterations. To the SIMPLER preconditioner, the reference [19] is chosen for a comparison. A Finite volume method is used in [19] while the number of linear iterations is still close to that shown in this paper. Again, the same settings are crucial to have a fair comparison.

Table 5: LDC: Picard, average GCR iterations and total solution time by using the modified AL, 'grad-div' preconditioners with optimal γ and the ideal SIMPLER preconditioner. Sub-systems are solved by agmg, $\epsilon_{\texttt{agmg},u} = 10^{-2}$, $\epsilon_{\texttt{agmg},p} = 10^{-4}$, grids: 128^2.

Re	100	400	1000	$2500^\star$	$5000^\star$
modified AL preconditioner					
Picard iterations:	14	27	33	66	286
GCR iterations:	5	9	11	17	19
total time:	22.7	65.1	119.6	457.7	2636.3
modified 'grad-div' preconditioner					
Picard iterations:	13	27	31	51	308
GCR iterations:	7	11	16	28	24
total time:	10.8	35.8	64.4	159.5	812.5
ideal SIMPLER preconditioner					
Picard iterations:	14	27	31	51	325
GCR iterations:	40	53	63	92	107
total time:	81.5	235.2	508.4	929.7	9548.7

Table 6: LDC: Newton, average GCR iterations and total solution time by using the modified AL, 'grad-div' preconditioners with optimal γ and the modified SIMPLER preconditioner. Sub-systems are solved by agmg, $\epsilon_{\texttt{agmg},u} = 10^{-2}$, $\epsilon_{\texttt{agmg},p} = 10^{-4}$, grids: 128^2.

Re	100	400	1000	$2500^\star$	$5000^\star$
modified AL preconditioner					
Newton iterations:	6	7	7	8	9
GCR iterations:	8	14	21	33	50
total time:	14.8	26.2	74.6	194.2	277.1
modified 'grad-div' preconditioner					
Newton iterations:	6	7	8	9	9
GCR iterations:	10	17	28	53	77
total time:	8.5	15.7	32.7	119.1	167.9
modified SIMPLER preconditioner					
Newton iterations:	10	$8^\star$	$8^\star$	11	15
GCR iterations:	43	82	84	80	90
total time:	68.3	102.9	232.8	203.2	561.6

4.2. Flow over a finite flat plate (FP) [14]

This example, known as Blasius flow, models a boundary layer flow over a flat plate. To model this flow, the Dirichlet boundary condition $u_x = 1, u_y = 0$ is imposed at the inflow boundary ($x = -1$; $-1 \leq y \leq 1$) and also on the top and bottom of the channel ($-1 \leq x \leq 5$; $y = \pm 1$), representing walls moving from left to right with speed unity. The plate is modelled by imposing a no-flow condition on the internal boundary

Table 7: FP: Picard and average GCR iterations by using the ideal AL, 'grad-div' and SIMPLER preconditioners. Sub-systems are solved directly.

Re	1000	5000	10000	50000	100000
Grid	Picard / GCR iterations				
96×32					
AL:	14/4	16/4	18/3	29/4	47/4
GD:	13/6	14/6	14/6	16/5	16/4
SIMPLER:	16/66	28/63	29/78	142/70	433/48
192×64					
AL:	13/4	14/4	15/3	29/2	33/3
GD:	13/6	14/5	14/5	16/4	16/4
SIMPLER:	16/95	21/123	34/106	38/153	74/132
384×128					
AL:	13/4	13/4	15/3	29/3	30/3
GD:	13/5	14/5	14/5	15/5	17/5
SIMPLER:	15/123	19/184	27/183	42/226	40/219

Table 8: FP: Newton and average GCR iterations by using the ideal AL, 'grad-div' and SIMPLER preconditioners. Sub-systems are solved directly.

Re	1000	5000	10000	50000	100000
Grid	Newton / GCR iterations				
96×32					
AL:	6/4	6/4	5/5	7/7	6/9
GD:	7/7	7/9	6/9	6/9	7/9
SIMPLER:	11/94	8/123	8/182	11/215	11/241
192×64					
AL:	6/4	5/4	5/4	6/5	6/6
GD:	7/7	6/8	6/8	6/7	6/8
SIMPLER:	10/139	8/216	8/232	12/327	14/362
384×128					
AL:	6/4	5/4	5/4	6/5	6/5
GD:	7/7	6/8	6/8	6/8	6/8
SIMPLER:	9/185	7/344	8/369	12/395	14/433

($0 \leq x \leq 5$; $y = 0$), and the Neumann condition is applied at the outflow boundary ($x = 5$; $-1 < y < 1$), i.e., $\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - \mathbf{n}p = \mathbf{0}$. The reference velocity and length are chosen as $U = 1$ and $L = 5$. Thus, the Reynolds number is $Re = 5/\nu$. The non-uniform grid is generated by applying the stretching function in the y-direction with $b = 1.01$ [c.f., [19]]:

$$y = \frac{(b+1) - (b-1)c}{(c+1)}, \quad c = \left(\frac{b+1}{b-1}\right)^{1-\bar{y}}, \quad \bar{y} = 0, \frac{1}{n}, \frac{2}{n}, \cdots, 1.$$

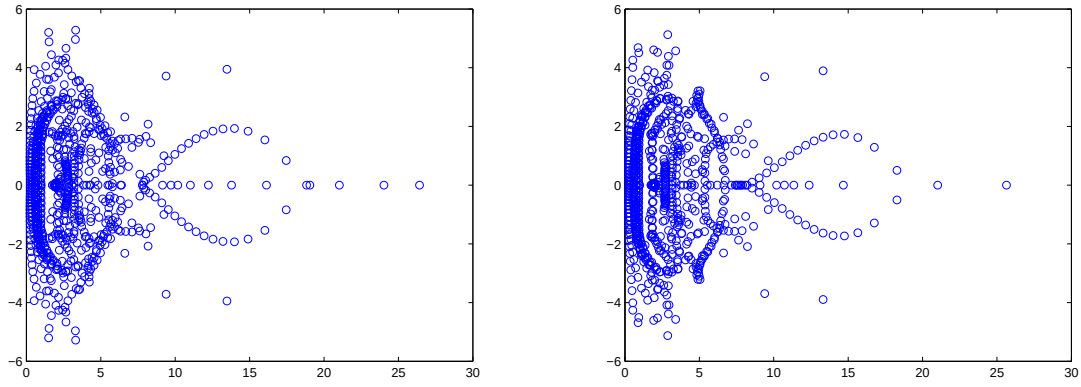
The results in Tables 7-8 illustrate that the ideal AL and 'grad-div' preconditioners are fully independent of the mesh size and the Reynolds number in both the Picard and Newton linearization methods.

The performance of the modified AL and 'grad-div' preconditioners, see Tables 9-10, is independent of the mesh size in both the Picard and Newton linearizations. Independence of the Reynolds number is clearly seen in the Picard linearization, and more clearly exhibited in the Newton linearization on the finest grid. Still we see that the optimal γ involved therein changes with varying Reynolds numbers, but does not depend on the mesh size. For the modified AL preconditioner, the interval containing the optimal values of γ is $[0.06,\ 0.3]$ in the Picard linearization, and is $[0.05,\ 0.1]$ in the Newton linearization. For the modified 'grad-div' preconditioner, the intervals are the same as the modified AL preconditioner in both the Picard and Newton linearizations. The intervals are different from those in LDC problem, and this observation also confirms that the optimal values of γ are problem dependent.

Results in Tables 7-8 show that the ideal SIMPLER preconditioner is nearly independent of the Reynolds number in the Picard linearization, but this is not valid in the Newton method. In both the Picard and Newton methods, the ideal SIMPLER preconditioner is not independent of the mesh size. Still, more `GCR` iterations are needed in the Newton linearization, the same reason as given in the LDC problem.

For the Newton linearization, the number of `GCR` iterations by using the modified SIMPLER preconditioner is nearly the same as that by using the ideal SIMPLER preconditioner, see Table 8 and Table 10. This illustrates that the modified SIMPLER preconditioner is also a successful improvement to the ideal one in this test case.

The results in Tables 11-12 are computed by using `agmg` for subsystems. The place where noted with $\star$ denotes that the perturbation technique is added there. The perturbation is the same and added due to the already mentioned reason as for the LDC



(a) ideal $\mathcal{P}_{SIMPLE}$ (b) modified $\mathcal{P}_{SIMPLE}$

Figure 1: FP: the spectrum of the preconditioned matrix $\mathcal{P}_{SIMPLE}^{-1}\mathcal{A}$ by using the ideal and modified SIMPLE preconditioners (at the last Newton iteration) on the grid 96×32.

Table 9: FP: Picard and average GCR iterations by using the modified AL and 'grad-div' preconditioners with optimal γ. Sub-systems are solved directly.

Re	1000	5000	10000	50000	100000
Grid	Picard / GCR iterations (optimal γ)				
96×32					
AL:	14/8(0.3)	16/12(0.1)	18/12(0.1)	29/12(0.08)	47/16(0.06)
GD:	14/16(0.3)	14/22(0.1)	14/22(0.1)	17/22(0.08)	17/24(0.06)
192×64					
AL:	13/10(0.3)	15/12(0.1)	16/12(0.1)	30/10(0.08)	35/14(0.06)
GD:	14/16(0.3)	14/22(0.1)	14/22(0.1)	17/22(0.08)	16/26(0.06)
384×128					
AL:	14/14(0.3)	14/13(0.1)	14/13(0.1)	22/14(0.08)	30/13(0.06)
GD:	14/18(0.3)	14/24(0.1)	14/24(0.1)	17/31(0.08)	15/32(0.06)

Table 10: FP: Newton and average GCR iterations by using the modified AL, 'grad-div' preconditioners with optimal γ and the modified SIMPLER preconditioner. Sub-systems are solved directly.

Re	1000	5000	10000	50000	100000
Grid	Newton / GCR iterations (optimal γ)				
96×32					
AL:	7/17(0.1)	6/20(0.08)	6/25(0.08)	7/50(0.05)	7/55(0.05)
GD:	7/22(0.1)	6/28(0.08)	6/34(0.06)	7/39(0.04)	7/40(0.04)
SIMPLER:	11/93	8/135	7/166	10/233	9/270
192×64					
AL:	7/14(0.1)	6/19(0.08)	6/21(0.08)	7/51(0.05)	7/55(0.05)
GD:	7/25(0.1)	6/32(0.08)	6/41(0.06)	7/48(0.04)	7/50(0.04)
SIMPLER:	10/140	7/208	7/230	11/314	11/354
384×128					
AL:	7/14(0.1)	6/21(0.08)	6/26(0.08)	7/32(0.05)	6/41(0.05)
GD:	7/26(0.1)	6/41(0.08)	6/47(0.06)	7/56(0.04)	6/55(0.04)
SIMPLER:	9/179	7/333	8/364	12/390	14/426

problem. Still, we see that the modified 'grad-div' preconditioner is most efficient in terms of the total solution time for this test problem. Again, this conclusion may change if other solution methods are used for subsystems involved in these preconditioners.

5. Conclusions

In this paper we consider to use the AL, 'grad-div' and SIMPLER preconditioners and new variants of them for the two-by-two block systems arising in the incompressible Navier-Stokes equations. The experiments in this paper confirm that the improvements for these three preconditioner are very successful. The modified 'grad-div' precondi-

Table 11: FP: Picard, average GCR iterations and total solution time by using the modified AL, 'grad-div' preconditioners with optimal γ and the ideal SIMPLER preconditioner. Sub-systems are solved by agmg, $\epsilon_{\texttt{agmg},u} = 10^{-2}$, $\epsilon_{\texttt{agmg},p} = 10^{-4}$, grids: 192×64.

Re	1000	5000	10000	50000	100000
modified AL preconditioner					
Picard iterations:	14	21	26	33	38
GCR iterations:	10	11	10	19	22
total time:	40.8	87.5	97.9	201.4	311.3
modified 'grad-div' preconditioner					
Picard iterations:	14	14	14	17	16
GCR iterations:	17	23	22	22	24
total time:	27.1	34.0	32.7	89.5	69.8
ideal SIMPLER preconditioner					
Picard iterations:	19	34	39	43	74
GCR iterations:	88	105	105	152	130
total time:	294.3	1154.8	1483.2	1375.3	2652.6

Table 12: FP: Newton, average GCR iterations and total solution time by using the modified AL, 'grad-div' preconditioners with optimal γ and the modified SIMPLER preconditioner. Sub-systems are solved by agmg, $\epsilon_{\texttt{agmg},u} = 10^{-2}$, $\epsilon_{\texttt{agmg},p} = 10^{-4}$, grids: 192×64.

Re	1000	5000	10000	50000	100000
modified AL preconditioner*					
Newton iterations:	8	7	7	8	8
GCR iterations:	19	23	27	59	89
total time:	69.3	51.9	65.3	115.1	214.5
modified 'grad-div' preconditioner					
Newton iterations:	8	7	7	9	8
GCR iterations:	25	32	39	44	44
total time:	27.5	22.9	82.7	102.8	97.4
modified SIMPLER preconditioner					
Newton iterations:	14	9	9	11	12
GCR iterations:	108	146	172	269	373
total time:	290.3	267.3	317.8	518.3	788.8

tioner turns out to be the most efficient in terms of total solution time and the memory storage. This conclusion may change if other solution methods are used for subsystems involved in the tested preconditioners. Improving some already known solvers, such as agmg, to make them more efficient is very crucial typically when the Reynolds number is high.

References

[1] O. Axelsson. Iterative Solution Methods. Cambridge University Press: Cambridge, 1994.

[2] O. Axelsson and R. Blaheta. Preconditioning of matrices partitioned in two-by-two block form: Eigenvalue estimates and Schwarz DD for mixed FEM. *Numer. Lin. Alg. Appl.,* 17:787–810, 2010.

[3] O. Axelsson and M. Neytcheva. A general approach to analyse preconditioners for two-by-two block matrices. *Numer. Lin. Alg. Appl.,* article first published online: 14 DEC 2011, dpi: 10.1002/nla.830.

[4] O. Axelsson and M. Neytcheva. Eigenvalue estimates for preconditioned saddle point matrices. *Numer. Lin. Alg. Appl.,* 13:339-360, 2006.

[5] M. Benzi. Preconditioning techniques for large linear systems: a survey. *J. Comput. Phys.,* 182:418-477, 2002.

[6] M. Benzi and M.A. Olshanskii. An augmented Lagrangian-based approach to the Oseen problem. *SIAM J. Sci. Comput.,* 28:2095-2113, 2006.

[7] M. Benzi, M.A. Olshanskii and Z. Wang. Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids,* 66:486-508, 2011.

[8] M. Benzi, G.H. Golub and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica,* 14:1-137, 2005.

[9] S. Börm and S. Le Borne. $\mathcal{H}$-LU factorization in preconditioners for augmented Lagrangian and grad-div stabilized saddle point systems. *Int. J. Numer. Meth. Fluids,* 68:83-98, 2012.

[10] A. de Niet and F.W. Wubs. Two preconditioners for saddle point problems in fluid flows. *Int. J. Numer. Meth. Fluids,* 54:355-377, 2007.

[11] S.C. Eisenstat, H.C. Elman and M.H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.,* 20:345-357, 1983.

[12] H.C. Elman, V.E. Howle, J. Shadid, R. Shuttleworth and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM J. Sci. Comput.,* 27:1651-1668, 2006.

[13] H.C. Elman and D.J. Silvester. Fast nonsymmetric iterations and preconditioning for the Navier-Stikes equations. *SIAM J. Sci. Comput.,* 17:33-46, 1996.

[14] H.C. Elman, D.J. Silvester and A.J. Wathen. Finite Element and Fast Iterative Solvers: with Application in Incompressible Fluid Dynamics. Oxford Series in Numerical Mathematics and Scientific Computation, Oxford University Press: Oxford, UK, 2005.

[15] H.C. Elman, D.J. Silvester and A.J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numer. Math.*, 90:665-688, 2002.

[16] X. He, M. Neytcheva and S. Serra Capizzano. On an Augmented Lagrangian-Based Preconditioning of Oseen Type Problems. *BIT Numerical Mathematics,* 51:865-888, 2011.

[17] T. Heister. *A Massively Parallel Finite Element Framework with Application to Incompressible Flows*. PhD Thesis, Göttingen University, 2011, available online.

[18] D. Kay, D. Loghin and A. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM J. Sci. Comput.,* 24:237-256, 2002.

[19] C.M. Klaij and C. Vuik. SIMPLE-type preconditioners for cell-centered, collocated finite volume discretization of incompressible Reynolds-averaged Navier-Stokes equations. *Int. J. Numer. Meth. Fluids,* 71:830-849, 2013.

[20] C. Li and C. Vuik. Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow. *Numer. Lin. Alg. Appl.,* 11:511-523, 2004.

[21] M.F. Murphy, G.H. Golub and A.J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.,* 21:1969-1972, 2000.
[22] A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM J. Sci. Comput.,* 34:A1079-A1109, 2012.
[23] M. Neytcheva, M. Do-Quang and X. He. Element-by-element Schur complement approximations for general nonsymmetric matrices of two-by-two block form. *Lecture Notes in Computer Science*, 5910:108-115, 2009.
[24] Y. Notay. An aggregation-based algebraic multigrid method. *Electron. T. Numer. Ana.,* 37:123-146, 2010.
[25] Y. Notay. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM J. Sci. Comput.,* 34:A2288-A2316, 2012.
[26] Y. Notay. A new analysis of block preconditioners for saddle point problems. *Report GANMN 13-01*, Universite Libre de Bruxelles, Brussels, Belgium, 2013.
[27] M.A. Olshanskii and Y.V. Vassilevski. Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM J. Sci. Comput.,* 29:2686-2704, 2007.
[28] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill, New York, 1980.
[29] M. ur Rehman, T. Geenen, C. Vuik, G. Segal and S.P. MacLachlan. On iterative methods for the incompressible Stokes problem. *Int. J. Numer. Meth. Fluids,* 65:1180-1200, 2011.
[30] T. Rusten and R. Winther. A preconditioned iterative method for saddle point problems. *SIAM J. Matrix Anal. Appl.,* 13:887-904, 1992.
[31] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM: Philadelphia, PA, 2003.
[32] Y. Saad, H.A. van der Vorst. Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.,* 123:1-33. 2000.
[33] M. Sahin and R.G. Owens. A novel fully implicit finite volume method applied to the lid-driven cavity problem-part I: high Reynolds number flow calculations. *Int. J. Numer. Meth. Fluids,* 42:57-77, 2003.
[34] A. Segal, M. ur Rehman and C. Vuik. Preconditioners for the incompressible Navier-Stokes equations. *Numer. Math. Theor. Meth. Appl.,* 3:245-275, 2010.
[35] C. Vuik, A. Saghir and G.P. Boerstoel. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *Int. J. Numer. Meth. Fluids,* 33:1027-1040, 2000.