DELFT UNIVERSITY OF TECHNOLOGY

REPORT 13-10

Comparison of some preconditioners for the incompressible Navier-Stokes equations

X. He and C. Vuik

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2013

Abstract

In this report we explore the performance of the SIMPLER , augmented Lagrangian, 'grad-div' preconditioners and their 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 Reynolds number varies from a low to the limiting number for a laminar flow.

Mathematics subject classification: 65F10, 65F08.

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

1 Introduction

A mathematical model for incompressible flows reads as follows:

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{on } \Omega, \\ \nabla\cdot\mathbf{u} &= 0 \quad \text{on } \Omega, \\ \mathbf{u} &= \mathbf{g} \quad \text{on } \partial\Omega_D, \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p &= 0 \quad \text{on } \partial\Omega_N. \end{aligned} \tag{1}$$

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 $\mathbf{n}$ denotes the outward-pointing unit normal to the boundary.

Due to the presence of the term $(\mathbf{u}\cdot\nabla)\mathbf{u}$, the system (1) is nonlinear. Thus, a linearization technique is needed. Quite often, the Picard and Newton methods are utilised. In both approaches the finite element (FE) discretization of the linearized problems results in systems of two-by-two block form. In the past decades many state-of-art preconditioners are proposed for the two-by-two block systems arising in the incompressible Navier-Stokes equations. In this report we choose the SIMPLER, augmented Lagrangian and 'grad-div' preconditioners to explore their efficiency on some academic benchmark problems. Besides, we propose improvements for these three preconditioners. The organization of the report 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

For the weak formulation of the stationary Navier-Stokes equations (1), we define the approximate solution and test spaces for the velocity as

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

and for the pressure as

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

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

$$\begin{aligned}\nu \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} d\Omega + \int_\Omega (\mathbf{u} \cdot \nabla\mathbf{u})\mathbf{v} d\Omega - \int_\Omega p\nabla \cdot \mathbf{v} d\Omega &= \int_\Omega \mathbf{f}\mathbf{v} d\Omega,\\ \int_\Omega q\nabla \cdot \mathbf{u} d\Omega &= 0,\end{aligned} \tag{2}$$

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 [12].

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 [12], 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 (2), 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

$$\begin{aligned}\nu \int_\Omega \nabla\delta\mathbf{u}_k : \nabla\mathbf{v} d\Omega + \int_\Omega (\mathbf{u}_k \cdot \nabla\delta\mathbf{u}_k) \cdot \mathbf{v} d\Omega + \int_\Omega (\delta\mathbf{u}_k \cdot \nabla\mathbf{u}_k) \cdot \mathbf{v} d\Omega - \int_\Omega \delta p_k \, (\nabla \cdot \mathbf{v}) d\Omega &= R_k\\ \int_\Omega q \, (\nabla \cdot \delta\mathbf{u}_k) d\Omega &= P_k,\end{aligned} \tag{3}$$

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

$$\begin{aligned}R_k &= \int_\Omega \mathbf{f} \cdot \mathbf{v} d\Omega - \nu \int_\Omega \nabla\mathbf{u}_k : \mathbf{D}\mathbf{v} d\Omega - \int_\Omega (\mathbf{u}_k \cdot \nabla\mathbf{u}_k) \cdot \mathbf{v} d\Omega + \int_\Omega p_k(\nabla \cdot \mathbf{v}) d\Omega\\ P_k &= -\int_\Omega q \, (\nabla \cdot \mathbf{u}_k) d\Omega.\end{aligned} \tag{4}$$

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

Picard linearization is obtained in a similar way as 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 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

$$\begin{aligned}\nu\int_\Omega\nabla\delta\mathbf{u}_k:\nabla\mathbf{v}d\Omega+\int_\Omega(\mathbf{u}_k\cdot\nabla\delta\mathbf{u}_k)\cdot\mathbf{v}d\Omega-\int_\Omega\delta p_k\ (\nabla\cdot\mathbf{v})d\Omega&=R_k\\ \int_\Omega q\ (\nabla\cdot\delta\mathbf{u}_k)d\Omega&=P_k,\end{aligned}\tag{5}$$

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.

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 Newton and Picard linearizations are of the form

$$\begin{bmatrix}A & B^T\\ B & O\end{bmatrix}\begin{bmatrix}\delta\mathbf{u}_h\\ \delta\mathbf{p}_h\end{bmatrix}=\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\quad\text{or}\quad\mathcal{A}\mathbf{x}=\mathbf{b},\tag{6}$$

where the system matrix $\mathcal{A}=\begin{bmatrix}A & B^T\\ B & O\end{bmatrix}$ is nonsymmetric 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., [12]). Here we assume that the LBB condition is satisfied, otherwise, the $(2,2)$ block becomes nonzero because some stabilization is required. When comparing Newton and Picard linearization methods, the difference appears in the pivot block $F\in\mathbb{R}^{n_u\times n_u}$, 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

$$\begin{aligned}A_\nu\in\mathbb{R}^{n_u\times n_u},&\quad [A_\nu]_{i,j}=\nu\int_\Omega\nabla\vec{\varphi}_i:\nabla\vec{\varphi}_j,\\ N\in\mathbb{R}^{n_u\times n_u},&\quad [N]_{i,j}=\int_\Omega(\mathbf{u}_h\cdot\nabla\vec{\varphi}_j)\vec{\varphi}_i,\\ \widehat{N}\in\mathbb{R}^{n_u\times n_u},&\quad [\widehat{N}]_{i,j}=\int_\Omega(\vec{\varphi}_j\cdot\nabla\mathbf{u}_h)\vec{\varphi}_i.\end{aligned}\tag{7}$$

Newton method corresponds to $\delta_1 = 1$, while Picard method corresponds to $\delta_1 = 0$. Also, in Picard linearization, the velocity pivot block A is of a block diagonal form. However, this structure does not hold in Newton linearization due to the presence of $\widehat{N}$. The linear system (6) arising in Newton (3) or Picard method (5) is often referred to as the Oseen problem.

Finding the solutions of the linear systems in (6) is the kernel and most time-consuming part in the numerical simulations. Therefore, fast and reliable solution techniques are critical. In this report Krylov subspace iterative solution methods [12, 28] accelerated by numerically and computationally efficient preconditioners are utilised.

Linear systems of the form (6) 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, 3, 4, 18, 23], the surveys [5, 8, 10, 30] and the books [12, 28], with numerous references therein. In general, the exact factorization of a two-by-two block matrix is

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & O \\ A_{21} & S \end{bmatrix} \begin{bmatrix} I_1 & A_{11}^{-1}A_{12} \\ O & I_2 \end{bmatrix} = \begin{bmatrix} I_1 & O \\ A_{21}A_{11}^{-1} & I_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ O & S \end{bmatrix}, \tag{8}$$

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, $A_{11} = A$, $A_{12} = B^T$, $A_{21} = B$ and $A_{22} = O$. So, $S = -BF^{-1}B^T$.

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

$$\begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{S} \end{bmatrix} \begin{bmatrix} I_1 & \widetilde{A}_{11}^{-1}A_{12} \\ O & I_2 \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{S} \end{bmatrix}, \qquad \begin{bmatrix} \widetilde{A}_{11} & A_{12} \\ O & \widetilde{S} \end{bmatrix}. \tag{10}$$

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-art approximations of the Schur complement are proposed and analysed, c.f., [6, 7, 11, 13, 14, 17, 20, 24, 26, 27, 31]. In this report we choose the SIMPLER, augmented Lagrangian and 'grad-div' preconditioners for study and furthermore propose some improvements.

3.1 The SIMPLER preconditioner

SIMPLE (semi-implicit pressure linked equation) is used by Patanker [25] 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 [17, 31] 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 saddle point matrices and due to their simplicity, remain attractive preconditioning techniques. We briefly describe both formulations for the Oseen problem $\mathcal{A}$ in (6).

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 $\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_p]$

SIMPLER differs slightly from SIMPLE. 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 $\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_p]$

It is pointed out in [16] that at high Reynolds number $D = \mathrm{diag}(A)$ can become very small in certain cells leading to sudden divergence. Therefore, the velocity block A is approximated by a diagonal matrix $\widetilde{D} = \sum |A|$ in [16], where $\sum |A|$ denotes the row sum of absolute values of A. In this report $\mathcal{P}_{SIMPLER}$ with $\widetilde{D} = \sum |A|$ is referred to as the ideal SIMPLER preconditioner.

In 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 $\widetilde{A} = \begin{bmatrix} \widetilde{A}_{11} & O \\ A_{21} & \widetilde{A}_{22} \end{bmatrix}$, and replace it with $\widetilde{A}$ in step 1 of the two algorithms above. Here the terms of $\widetilde{A}_{11}$ and $\widetilde{A}_{22}$ indicate that we use an inner iterative solver with a proper stopping tolerance for systems with the blocks A_{11} and A_{22}. The SIMPLER preconditioner with $\widetilde{A}$ and $\widetilde{D} = \sum |\widetilde{A}|$ is referred to as the modified SIMPLER preconditioner in this report. Since A is already of block diagonal structure in Picard linearization, the strategy leading to the modified SIMPLER preconditioner is not necessary. Numerical results in Section 4 illustrate that compared to the ideal SIMPLER preconditioner, the average number of the linear iterations does not increase by using the modified one. Thus, the computational time can be 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 $O(n_u^2/2)$. Theoretical analysis of the modified SIMPLER preconditioner is considered to be one research direction in future.

3.2 The augmented Lagrangian method

In the so-called augmented Lagrangian approach, we first transform the linear system (6) into an equivalent one with the same solution, which is of the form

$$\begin{bmatrix} A+\gamma B^T W^{-1} B & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{f}} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{A}_{AL}\mathbf{x} = \widehat{\mathbf{b}}, \tag{11}$$

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 (11) has the same solution as (6) for any value of γ and any nonsingular matrix W.

The equivalent system (11) is what we intend to solve and the AL-type preconditioner proposed for $\mathcal{A}_{AL}$ in (11) is of a block lower-triangular form

$$\mathcal{M}_{AL} = \begin{bmatrix} A+\gamma B^T W^{-1} B & O \\ B & -\frac{1}{\gamma} W \end{bmatrix}. \tag{12}$$

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, 14], for any non-singular matrix W the eigenvalues of the preconditioned matrix $\mathcal{M}_{AL}^{-1}\mathcal{A}_{AL}$ will cluster to the unit 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, 14]. In this report $\gamma = 1$ is chosen in the ideal AL preconditioner.

How to efficiently solve 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 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 Picard linearization) and $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$. Then the modified pivot block A_{AL} can be written as

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

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

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

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

$$\widetilde{\mathcal{M}}_{AL} = \begin{bmatrix} \widetilde{A}_{AL,11} & O & O \\ A_{AL,21} & \widetilde{A}_{AL,22} & O \\ B_1 & B_2 & -\frac{1}{\gamma}W \end{bmatrix}, \tag{13}$$

where the terms $\widetilde{A}_{AL,11}$ and $\widetilde{A}_{AL,22}$ denote approximations of $A_{AL,11}$ and $A_{AL,22}$, for instance, obtained via an inner iterative solution method with a proper stopping tolerance. In [7] systems with $A_{AL,11}$ and $A_{AL,22}$ are solved by using the direct solution method, which is not feasible for large simulations in terms of the total solution time and storage. In this report we test an algebraic multigrid (AMG) solver.

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 is 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 report.

An other parameter in the AL framework is the matrix parameter W. The transformation (11) holds true for any nonsingular matrix W. In practice W is often chosen to be the pressure mass matrix M, or its diagonal $\mathrm{diag}(M)$ [6, 7, 14]. Theoretical analysis and numerical experiments therein show that the ideal and modified AL preconditioners with $W = M$ or $\mathrm{diag}(M)$ are independent of the mesh refinement. In this report, $W = \mathrm{diag}(M)$ is utilised in modified and ideal AL preconditioners.

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 (1), one can obtain the so-called 'grad-div' stabilization formulation (cf. e.g., [9, 15])

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \gamma\nabla(\nabla\cdot\mathbf{u}) + \nabla p = \mathbf{f} \quad \text{on } \Omega, \\ \nabla\cdot\mathbf{u} = 0 \quad \text{on } \Omega. \end{aligned} \tag{14}$$

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,

$$\begin{bmatrix} F_{GD} & B^T \\ B & O \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_h \\ \delta \mathbf{p}_h \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{or} \quad \mathcal{F}_{GD}\mathbf{x} = \mathbf{b}, \tag{15}$$

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 (7). 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

$$\mathcal{M}_{GD} = \begin{bmatrix} F_{GD} & O \\ B & -\frac{1}{\gamma}W \end{bmatrix}. \tag{16}$$

In this report 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 $\mathrm{diag}(M)$) and $\gamma = 1$ in [9, 15]. In this report $W = \mathrm{diag}(M)$ and $\gamma = 1$ are chosen.

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.

Using the same strategy which leads to the modified AL preconditioner, we can modify the 'grad-div' preconditioner as follows. 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 Picard and Newton linearizations due to the presence of the matrix G. We approximate it as $\widetilde{F}_{GD} = \begin{bmatrix} \widetilde{F}_{GD,11} & O \\ F_{GD,21} & \widetilde{F}_{GD,22} \end{bmatrix}$ and the modified 'grad-div' preconditioner is obtained as

$$\widetilde{\mathcal{M}}_{GD} = \begin{bmatrix} \widetilde{F}_{GD,11} & O & O \\ F_{GD,21} & \widetilde{F}_{GD,22} & O \\ B_1 & B_2 & -\frac{1}{\gamma}W \end{bmatrix}, \tag{17}$$

where $\widetilde{F}_{GD,11}$ and $\widetilde{F}_{GD,22}$ denote that we use an inner iterative solution method with a proper stopping tolerance for solving systems with them. In the modified 'grad-div' preconditioner, we still choose $W = \mathrm{diag}(M)$ and its efficiency is dependent of the parameter γ. In this report we use numerical experiments to determine the optimal γ.

We thoroughly 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 the three preconditioners introduced in Section 3. The test academic problems are the lid-driven cavity and the flow over a finite flat plate. When Newton or Picard linearization method is used, the number of nonlinear iterations is refereed to as Newton or Picard iterations. The stopping tolerance for the nonlinear iterations is relative and chosen to be 10^{-10}. In order to achieve a fast convergence rate of the nonlinear solver, the *continuation* method is utilised. This method means that in Newton or Picard linearization method, 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`) [1] 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 report we also use an aggregation based algebraic multigrid method– `agmg` (see [21, 19, 22]) to solve the subsystems in the 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 pressure unknowns, the relative stopping tolerance is denoted as $\epsilon_{\texttt{agmg},u}$ and $\epsilon_{\texttt{agmg},p}$.

All experiments in this report 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 report 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 introduced in every test problem.

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 length are chosen as $U = 1$ and $L = 1$. Thus, 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 $b = 1.1$ [c.f., [16]]

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

The results in Table 1-4 are obtained by solving subsystems in these three preconditioners through a direct solution method. A 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 Reynolds number in both Newton and Picard linearization methods, see Table 1-2. Results in Table 3 show that in Picard linearization, the modified AL and 'grad-div' preconditioners are independent of the mesh size and Reynolds number too. In Newton linearization, the modified AL and 'grad-div' preconditioners are independent of the mesh size but dependent of Reynolds number, see Table 4. We see from Table 4 that in Newton linearization the number of `GCR` iterations by using the modified AL preconditioner is 44, which is acceptable for the 'worst' situation with $Re = 10000$ 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 big change of the number of `GCR` iterations.

From Table 3-4, we see that the optimal value of γ for the modified AL and 'grad-div' preconditioners changes with varying Reynolds number. We see that for the modified AL preconditioner the optimal value of γ lies in the interval $[0.004,\ 0.02]$ in Picard linearization, and in $[0.008,\ 0.04]$ in Newton linearization. For the modified 'grad-div' preconditioner, the optimal γ lies in the interval $[0.01,\ 0.06]$ in Picard linearization, and in $[0.02,\ 0.06]$ in Newton linearization. The meaning of finding out these intervals is that in practice they can help users 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 Reynolds number in Picard linearization, but this independence is not remained in Newton linearization, see Table 1-2. Also, Newton linearization results in more `GCR` iterations by using the ideal SIMPLER preconditioner, compared to 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 Newton linearization, the velocity sub-block is more complicated and this type of approximation of it does not work as well as in Picard linearization. Independence of the mesh size by using the ideal SIMPLER preconditioner is not obtained in both Picard and Newton linearizations, see Table 1-2. Results in Table 4 show that in Newton linearization, the number of `GCR` iterations by using the modified SIMPLER preconditioner is independent of 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, in Picard linearization it is not necessary to use the strategy leading to the modified SIMPLER preconditioner. In Newton linearization, the velocity block in the ideal SIMPLER preconditioner is of a block two-by-two form in 2D. In order to reduce computational complexity, we convert it to a block lower-triangular form as used in the modified SIMPLER preconditioner. In Picard linearization, the velocity block is already of a block diagonal structure. This is the reason why we only present the modified AL and 'grad-div' preconditioners in Table 3. For 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 Table 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 Table 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 from diagonally dominant. `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. 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 Table 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 on-going research.

Picard iterations may stagnate if Reynolds number is large, such as $Re = 10000$. To avoid the possibly slow convergence rate of Picard linearization and possibly narrow convergence region of Newton method, a combination of these two methods can be utilised. One can carry out some Picard iterations to obtain a reasonably 'good' solution, and then use this solution as an initial guess for Newton iterations. In this way, a fast convergence rate of the nonlinear iterations can be achieved. More numerical experiments on this type of nonlinear solution method are carrying out now and a comparison between it and the *continuation* method will be reported.

Figure 1-2 plot the profiles of two components of the computed velocity, and these plots

are close to the results given in [29].

There are many references testing the three preconditioners and their variants on the LDC problem, c.f. [7, 14, 9, 15, 16]. It is not easy to compare the results given in this report and other references, since different settings are used in different papers. For the ideal and modified AL preconditioners, the number of linear iterations are close to each other in this report and [7, 14] 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, 15]. The same settings also result in a comparable result, in terms of the number of linear iterations. To the SIMPLER preconditioner, the reference [16] is chosen for comparison. Finite volume method is used in [16] while the number of linear iterations is still close to the results shown in this report. Sure, the same settings are crucial to have a fair comparison.

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	10000
Grid	Picard / `GCR` iterations					
64^2						
AL:	14/2	29/2	32/2	57/2	stagnation	stagnation
GD:	14/4	28/3	31/3	51/3	stagnation	stagnation
SIMPLER:	15/23	28/32	32/30	58/29	stagnation	stagnation
128^2						
AL:	13/2	28/2	31/2	48/2	308/2	stagnation
GD:	14/4	27/3	30/3	50/3	282/3	stagnation
SIMPLER:	17/32	32/41	32/41	48/42	284/42	stagnation
256^2						
AL:	13/2	26/2	28/2	45/2	202/2	stagnation
GD:	14/4	26/3	29/3	50/3	218/3	stagnation
SIMPLER:	19/40	41/58	52/58	55/62	212/62	stagnation

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	10000
Grid	Newton / GCR iterations					
64^2						
AL:	5/2	7/2	7/2	7/3	6/3	7/3
GD:	7/6	7/6	8/6	7/6	7/6	7/7
SIMPLER:	7/35	8/50	9/72	9/108	10/167	18/210
128^2						
AL:	5/2	7/2	6/2	6/3	6/3	6/4
GD:	7/6	8/6	7/6	8/6	7/6	8/7
SIMPLER:	7/56	8/66	9/90	9/140	10/232	26/223
256^2						
AL:	5/2	7/2	6/2	6/3	6/3	6/4
GD:	7/6	8/6	7/6	7/6	7/6	7/6
SIMPLER:	8/73	8/102	11/122	8/174	10/251	34/237

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	10000
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	stagnation
GD:	14/7(0.06)	27/11(0.04)	32/15(0.02)	66/14(0.01)	stagnation	stagnation
128^2						
AL:	14/5(0.02)	28/9(0.01)	33/11(0.008)	45/9(0.006)	306/14(0.004)	stagnation
GD:	13/7(0.06)	27/11(0.04)	31/16(0.02)	55/15(0.01)	296/18(0.01)	stagnation
256^2						
AL:	14/5(0.02)	28/9(0.01)	33/10(0.008)	45/10(0.006)	202/12(0.004)	stagnation
GD:	13/7(0.06)	26/10(0.04)	30/15(0.02)	55/14(0.01)	218/17(0.01)	stagnation

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.

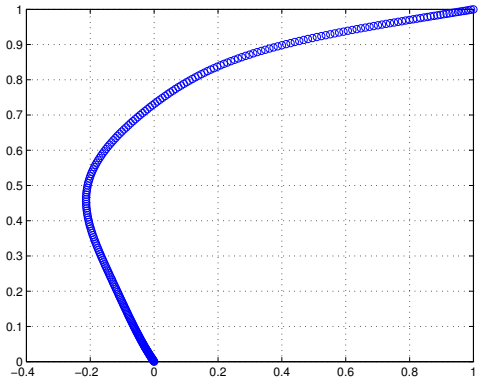
Re	100	400	1000	2500	5000	10000
Grid	Newton / GCR iterations (optimal γ)					
64^2						
AL:	6/7(0.04)	7/14(0.04)	7/21(0.02)	9/34(0.01)	9/45(0.01)	11/49(0.008)
GD:	6/10(0.06)	7/16(0.04)	7/27(0.04)	8/41(0.02)	8/55(0.02)	10/84/(0.02)
SIMPLER:	8/34	9/43	11/47	17/63	21/67	stagnation
128^2						
AL:	6/7(0.04)	7/15(0.04)	7/22(0.02)	8/33(0.01)	9/43(0.01)	10/46(0.008)
GD:	6/10(0.06)	7/17(0.04)	7/29(0.04)	8/45(0.02)	8/64(0.02)	10/90(0.02)
SIMPLER:	9/52	9/71	13/67	18/64	26/71	stagnation
256^2						
AL:	6/7(0.04)	6/15(0.04)	7/22(0.02)	8/34(0.01)	9/40(0.01)	10/44(0.008)
GD:	6/10(0.06)	6/17(0.04)	7/29(0.04)	8/45(0.02)	8/65(0.02)	10/95(0.02)
SIMPLER:	9/70	10/112	14/99	16/94	31/77	stagnation

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.

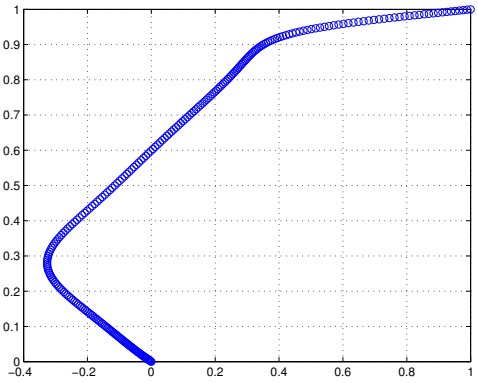
Re	100	400	1000	2500*	5000*	10000
modified AL preconditioner						
Picard iterations:	14	27	33	66	286	stagnation
GCR iterations:	5	9	11	17	19	
total time:	22.7	65.1	119.6	457.7	2636.3	
modified 'grad-div' preconditioner						
Picard iterations:	13	27	31	51	308	stagnation
GCR iterations:	7	11	16	28	24	
total time:	10.8	35.8	64.4	159.5	812.5	
ideal SIMPLER preconditioner						
Picard iterations:	14	27	31	51	325	stagnation
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.

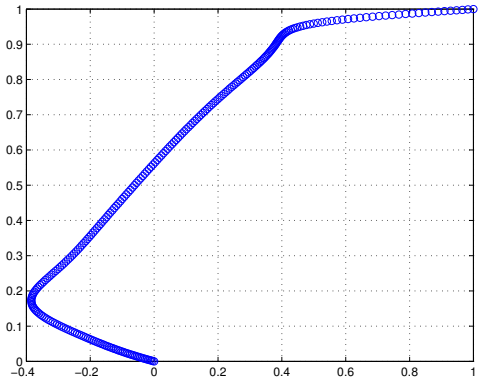
Re	100	400	1000	2500*	5000*	10000*
modified AL preconditioner						
Newton iterations:	6	7	7	8	9	13
GCR iterations:	8	14	21	33	50	95
total time:	14.8	26.2	74.6	194.2	277.1	606.7
modified 'grad-div' preconditioner						
Newton iterations:	6	7	8	9	9	13
GCR iterations:	10	17	28	53	77	126
total time:	8.5	15.7	32.7	119.1	167.9	355.6
modified SIMPLER preconditioner						
Newton iterations:	10	8*	8*	11	15	stagnation
GCR iterations:	43	82	84	80	90	
total time:	68.3	102.9	232.8	203.2	561.6	



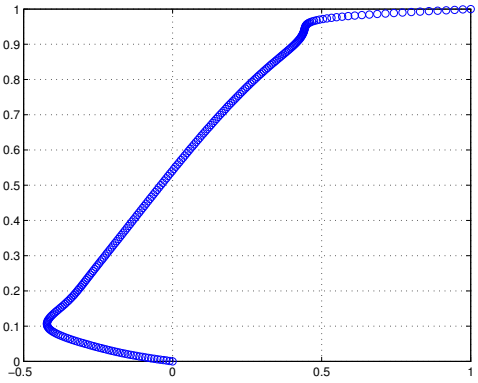
(a) $Re = 100$



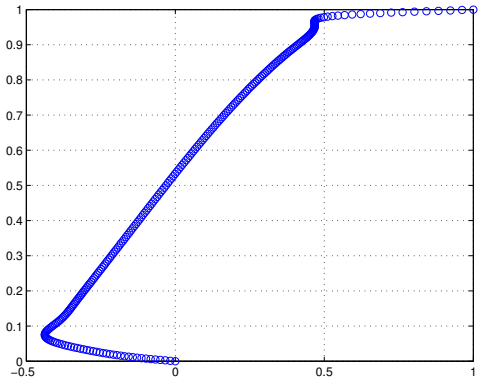
(b) $Re = 400$



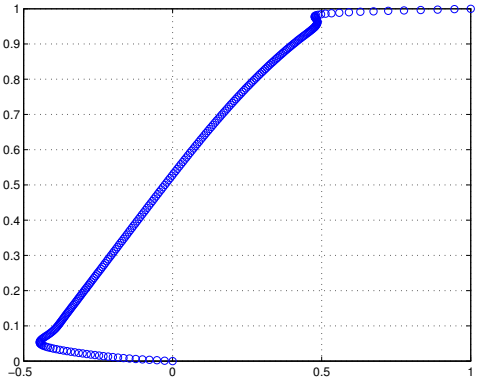
(c) $Re = 1000$



(d) $Re = 2500$

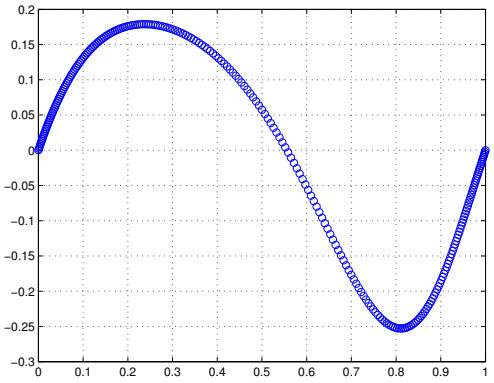


(e) $Re = 5000$

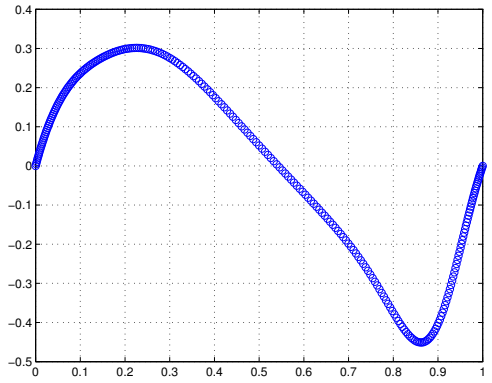


(f) $Re = 10000$

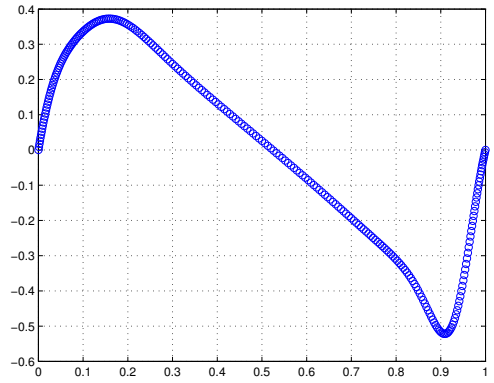
Figure 1: LDC: profiles of u along the line $x = 0.5$ computed on the stretched grid 256^2.



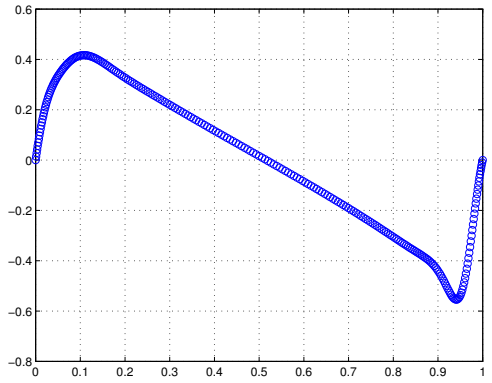
(a) $Re = 100$



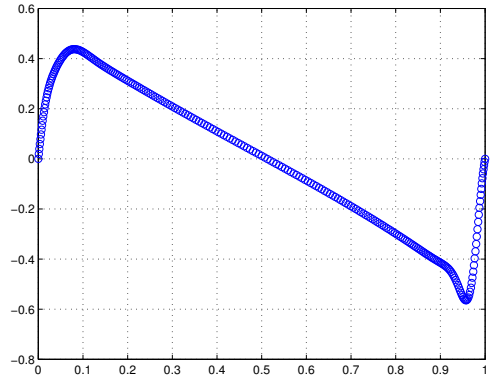
(b) $Re = 400$



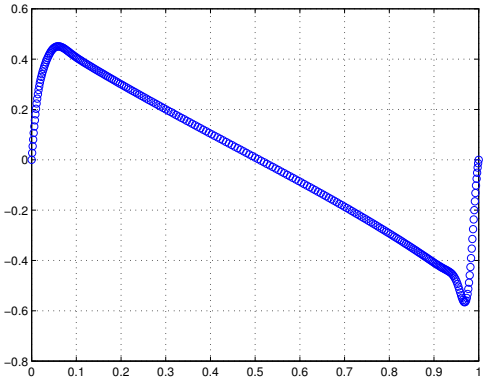
(c) $Re = 1000$



(d) $Re = 2500$



(e) $Re = 5000$



(f) $Re = 10000$

Figure 2: LDC: profiles of v along the line $y = 0.5$ computed on the stretched grid 256^2.

4.2 Flow over a finite flat plate (FP)

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 ($0 \leq x \leq 5$; $y = 0$), 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, 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., [16]]:

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

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

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

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

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

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

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

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

5 Conclusions

In this report we consider to use the AL, 'grad-div' and SIMPLER preconditioners and their variants for the two-by-two block systems arising in the incompressible Navier-Stokes equations. The main aim of this report is to explore their performance, based on thorough experiments on some benchmark problems. The modified 'grad-div' preconditioner turns out to be the most efficient in terms of total solution time. But this conclusion may change if other efficient solution methods are used for subsystems involved in the preconditioners. Improving some already known solvers, such as `agmg`, to make them more efficient is very crucial typically when Reynolds number is large.

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** (2010), 787–810.

[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. DOI: 10.1002/nla.830.

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

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

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

[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** (2011), 486-508.

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

[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** (2012), 83-98.

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

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

[12] 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.

[13] 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** (2002), 665-688.

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

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

[16] 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** (2013), 830-849.

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

[18] M.F. Murphy, G.H. Golub and A.J. Wathen, A note on preconditioning for indefinite linear systems, *SIAM J. Sci. Comput.,* **21** (2000), 1969-1972.

[19] A. Napov and Y. Notay, An algebraic multigrid method with guaranteed convergence rate, *SIAM J. Sci. Comput.,* **34** (2012), A1079-A1109.

[20] 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** (2009), 108-115.

[21] Y. Notay, An aggregation-based algebraic multigrid method, *Electron. T. Numer. Ana.,* **37** (2010), 123-146.

[22] Y. Notay, Aggregation-based algebraic multigrid for convection-diffusion equations, *SIAM J. Sci. Comput.,* **34** (2012), A2288-A2316.

[23] Y. Notay, A new analysis of block preconditioners for saddle point problems. *Report GANMN 13-01*, Universite Libre de Bruxelles, Brussels, Belgium, 2013.

[24] M.A. Olshanskii and Y.V. Vassilevski, Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM J. Sci. Comput.,* **29** (2007), 2686-2704.

[25] S.V. Patankar, Numerical heat transfer and fluid flow, McGraw-Hill, New York, 1980.

[26] 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** (2011), 1180-1200.

[27] T. Rusten and R. Winther, A preconditioned iterative method for saddle point problems, *SIAM J. Matrix Anal. Appl.,* **13** (1992), 887-904.

[28] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM: Philadelphia, PA, 2003.

[29] 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** (2003), 57-77.

[30] A. Segal, M. ur Rehman, C. Vuik, Preconditioners for the incompressible Navier-Stokes equations, *Numer. Math. Theor. Meth. Appl.,* **3** (2010), 245-275.

[31] C. Vuik, A. Saghir, G.P. Boerstoel, The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces, *Int. J. Numer. Meth. Fluids,* **33** (2000), 1027-1040.