Institute for Computational Mathematics
Hong Kong Baptist University

ICM Research Report
10-03

A Relaxed Dimensional Factorization Preconditioner for the Incompressible Navier-Stokes Equations

Michele Benzi[a], Michael Ng[b], Qiang Niu[c], Zhen Wang[a]

[a] *Department of Mathematics and Computer Science, Emory University, Atlanta, GA, 30322, USA.*
[b] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*
[c] *BNU-HKBU United International College, Zhuhai, 519085, P.R. China*

Abstract

In this paper we introduce a Relaxed Dimensional Factorization (RDF) preconditioner for saddle point problems. Properties of the preconditioned matrix are analyzed and compared with those of the closely related Dimensional Splitting (DS) preconditioner recently introduced by Benzi and Guo in [8]. Numerical results for a variety of finite element discretizations of both steady and unsteady incompressible flow problems indicate very good behavior of the RDF preconditioner with respect to both mesh size and viscosity.

Key words: saddle point problem, Navier–Stokes equations, Oseen problem, Krylov subspace method, dimensional splitting, dimensional factorization

1. Introduction

The Navier–Stokes equations are the fundamental model describing the flow of a viscous Newtonian fluid. In the incompressible case the Navier–Stokes equations in primitive variables are

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{on} \quad \Omega \times (0, T], \tag{1}$$

$$\text{div } \mathbf{u} = 0 \qquad \text{on} \quad \Omega \times [0, T], \tag{2}$$

$$\mathbf{u} = \mathbf{g} \qquad \text{on} \quad \partial\Omega \times [0, T], \tag{3}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{on} \quad \Omega, \tag{4}$$

where $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ is an open bounded domain with sufficiently smooth boundary $\partial\Omega$, $[0, T]$ is the time interval of interest, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ and $p = p(\mathbf{x}, t)$ are the unknown velocity and pressure fields, ν is the kinematic viscosity, Δ is the vector Laplacian, ∇ is the gradient, div the divergence, and $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{u}_0$ are given functions. The Stokes problem is obtained by dropping the nonlinearity $(\mathbf{u} \cdot \nabla)\mathbf{u}$ from the momentum equation; in the steady case, we can assume that the viscosity $\nu = 1$. We refer to [22] for an introduction to the numerical solution of the Stokes and Navier–Stokes equations.

Implicit time discretization together with spatial discretization of the Navier–Stokes system by, e.g., LBB-stable finite element methods (FEMs) leads to a sequence of large nonlinear systems of equations. Using Picard linearization, the solution of these nonlinear systems is reduced to a sequence of large sparse linear systems of equations with the following saddle point matrix structure:

$$\mathbf{H}\mathbf{x} = \mathbf{b}, \tag{5}$$

with

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ -\mathbf{B} & 0 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} \mathbf{f} \\ -g \end{pmatrix}.$$

Here $\mathbf{u}$ and p denote the discrete velocity and pressure, respectively. In two dimensions $(d = 2)$, $\mathbf{A} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$ denotes the discretization of the reaction, diffusion, and convection terms; $\mathbf{B}^T$ is the discrete gradient, $\mathbf{B}$ the (negative) discrete divergence, $\mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$ and g contain the forcing and boundary terms. Note that writing the constraint equation in the form $-\mathbf{B}\mathbf{u} = -g$ instead of the more frequently used form $\mathbf{B}\mathbf{u} = g$ leads to a coefficient matrix with eigenvalue spectrum entirely contained in the right half-plane. In contrast, the more symmetric-looking system with the positive sign in front of the constraint is highly indefinite, in the sense that its eigenvalues surround the origin in the complex plane. See [7, 12] for a discussion of these issues.

In recent years, considerable effort has been spent in developing efficient solvers for systems of form (5). Recent works on sparse direct methods for symmetric saddle point problems include [15, 18]. While highly reliable, direct methods usually require extensive resources in terms of computing time and memory, especially for three-dimensional (3D) problems [13]. Most of the recent work on saddle point problems has focused on the development of preconditioners for Krylov subspace methods, especially block precondition-

ers and multilevel schemes. We refer the readers to [7] for a comprehensive survey of existing approaches for solving saddle point problems. The ultimate goal of this research activity is the development of efficient and robust preconditioners, i.e., preconditioners that result in (fast) convergence rates that are as much as possible independent of problem parameters such as discretization parameters and viscosity.

An important class of preconditioners is based on the block LU factorization of the coefficient matrix $\mathbf{H}$ [3, 4, 22, 23, 25, 29, 33, 34, 35]. This class includes a variety of block diagonal and block triangular preconditioners. A major issue with such preconditioners is how to find a good approximation of the Schur complement $\mathbf{S} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$. While this class of preconditioners has proven to be effective in many cases, they are not yet completely robust for small values of the viscosity, especially for steady problems on stretched grids. Somewhat related to this class of methods are preconditioners based on the augmented Lagrangian (AL) reformulation of the saddle point problem; see [10, 11, 27] for recent work in this direction. AL-type preconditioners appear to be remarkably robust for a broad range of problem parameters, and they are currently the focus of intense development.

Constraint preconditioning, see [16, 19, 26, 31, 32], is another type of preconditioning techniques which has proven particularly useful in the solution of saddle point problems from large-scale optimization, including PDE-constrained optimization. We note that in these problems, the submatrix $\mathbf{A}$ is symmetric and positive (semi-)definite. Constraint preconditioning is seldom used in solving the Navier–Stokes equations, where $\mathbf{A}$ is nonsymmetric. Other types of preconditioners for (possibly nonsymmetric) saddle point problems include those based on the Hermitian and skew-Hermitian Splitting (HSS) and on the Dimensional Splitting (DS) of the coefficient matrix $\mathbf{H}$; see, respectively, [2, 6, 9, 17, 27] and [8]. These preconditioners have been shown to be effective on a wide range of problems. However, HSS is difficult to implement efficiently for the Oseen problem (except for the rotation form of the Navier–Stokes equations, cf. [9]), and DS preconditioning has difficulties dealing with low-viscosity problems on stretched grids.

In this paper we build on the DS preconditioner introduced in [8] and develop a technique which will be referred to as the *Relaxed Dimensional Factorization* preconditioner. The idea for this method comes from explicitly forming the (parameter-dependent) DS preconditioner, originally given in factorized form, and comparing the DS preconditioner with the original coefficient matrix $\mathbf{H}$. This reveals that certain diagonal terms of the DS preconditioner can be neglected without adversely affecting the quality of the approximation; indeed, dropping some of these terms actually leads to

a *better* approximation of **H**, suggesting the possibility of improvements in the performance of the preconditioner. This intuition is indeed confirmed in many cases both by numerical experiments and by comparing the clustering effect of RDF preconditioning with that of DS on the spectrum of the preconditioned matrices.

The convergence of Krylov subspace iterations is influenced by the spectrum distribution of the preconditioned matrix. In general, favorable convergence rates are often associated with a clustering of most of the eigenvalues around $\lambda = 1$ and away from zero. Here we derive some simple results for the spectrum of RDF-preconditioned matrices. Furthermore, we apply a (simplified) Fourier analysis to guide in the choice of the RDF parameter. We present the result of extensive numerical experiments, including comparisons with other preconditioners, using test problems generated from discretizations of the two-dimensional (2D) Stokes and Oseen equations by Q2-Q1 and Q2-P1 finite elements. We also present a few results for 3D problems.

The remainder of the paper is organized as follows. In section 2, after a brief introduction of the DS preconditioner, we present the RDF preconditioner and derive some spectral properties of the preconditioned matrix. In the same section we also show how Fourier analysis can be used to select the parameter in RDF. In section 3 we present the results of a series of numerical experiments demonstrating the performance of the RDF preconditioner, including comparisons with other preconditioners. Finally, some concluding remarks are drawn in section 4.

2. A Relaxed Dimensional Factorization Preconditioner

We begin this section with a brief description of DS preconditioning; for further details, see [8].

2.1. Dimensional Splitting Preconditioner

For simplicity, we limit ourselves to the 2D case. In this case **H** has the block structure

$$\mathbf{H} = \begin{pmatrix} A_1 & 0 & B_1^T \\ 0 & A_2 & B_2^T \\ -B_1 & -B_2 & 0 \end{pmatrix}, \tag{6}$$

i.e., $\mathbf{A} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} B_1 & B_2 \end{pmatrix}$ in (5). Each diagonal block matrix $A_i \in \mathcal{R}^{n_i \times n_i}$ is a discrete scalar convection-diffusion-reaction opera-

tor

$$A_i = \sigma M + \nu L + N_i, \tag{7}$$

with M being the velocity mass matrix, L the discrete (negative) Laplacian, and N_i the convective terms; the parameter $\sigma \geq 0$ is typically proportional to the reciprocal of the time step, and is zero in the steady case. Moreover, $B_1^T \in \mathcal{R}^{n_1 \times m}$, $B_2^T \in \mathcal{R}^{n_2 \times m}$ are discretizations of the partial derivatives $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, respectively. Therefore, the saddle point matrix $\mathbf{H}$ is $N \times N$ with $N = n_1 + n_2 + m$.

The *Dimensional Splitting* (DS) preconditioner proposed in [8] is of the form

$$\mathbf{P} = \frac{1}{\alpha} \begin{pmatrix} A_1 + \alpha I & 0 & B_1^T \\ 0 & \alpha I & 0 \\ -B_1 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & 0 & 0 \\ 0 & A_2 + \alpha I & B_2^T \\ 0 & -B_2 & \alpha I \end{pmatrix}, \tag{8}$$

and is suggested by splitting $\mathbf{H}$ as follows:

$$\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2 = \begin{pmatrix} A_1 & 0 & B_1^T \\ 0 & 0 & 0 \\ -B_1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & 0 \end{pmatrix}. \tag{9}$$

The alternating (ADI-like) stationary iteration corresponding to the DS splitting is

$$\begin{cases} (\alpha \mathbf{I} + \mathbf{H}_1)\mathbf{x}^{(k+\frac{1}{2})} = (\alpha \mathbf{I} - \mathbf{H}_2)\mathbf{x}^{(k)} + \mathbf{b}, \\ (\alpha \mathbf{I} + \mathbf{H}_2)\mathbf{x}^{(k+1)} = (\alpha \mathbf{I} - \mathbf{H}_1)\mathbf{x}^{(k+\frac{1}{2})} + \mathbf{b}, \end{cases} \tag{10}$$

(with $k = 0, 1, \ldots$, and $\mathbf{x}^{(0)}$ arbitrary). It is obtained alternating between the following two splittings of $\mathbf{H}$:

$$\mathbf{H} = (\alpha \mathbf{I} + \mathbf{H}_1) - (\alpha \mathbf{I} - \mathbf{H}_2)$$

and

$$\mathbf{H} = (\alpha \mathbf{I} + \mathbf{H}_2) - (\alpha \mathbf{I} - \mathbf{H}_1).$$

In [8], it is shown that the iteration (10) is convergent for all $\alpha > 0$ to the unique solution of $\mathbf{Hx} = \mathbf{b}$, provided that $\mathbf{A} + \mathbf{A}^T$ is positive definite and $\mathbf{B}$ has full rank. We should point out that the factor $\frac{1}{\alpha}$ used in (8) originally appears as $\frac{1}{2\alpha}$ in [8]. Since this factor has no effect on the preconditioned system, we use $\frac{1}{\alpha}$ in this paper just for analysis purpose.

By performing the matrix multiplication on the right-hand side of (8), it follows that $\mathbf{P}$ has the block structure

$$\mathbf{P} = \begin{pmatrix} \alpha I + A_1 & -\frac{1}{\alpha} B_1^T B_2 & B_1^T \\ 0 & \alpha I + A_2 & B_2^T \\ -B_1 & -B_2 & \alpha I \end{pmatrix}. \tag{11}$$

From (6) and (11), we can see that the difference between the preconditioner $\mathbf{P}$ and the coefficient matrix $\mathbf{H}$ is given by

$$\mathbf{P} - \mathbf{H} = \begin{pmatrix} \alpha I & -\frac{1}{\alpha} B_1^T B_2 & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{pmatrix}. \tag{12}$$

Equation (12) shows that as α tends to zero, the weight of the three diagonal blocks in the difference matrix decreases, whereas the weight of an off-diagonal block becomes unbounded. Hence, the choice of α requires a balancing act. The size of α actually depends on the scaling of the equations in the linear system. For a properly scaled linear system the optimal value of α is not too small.

2.2. A Relaxed Dimensional Factorization Preconditioner

Based on the previous observations, we propose an improved variant of the DS preconditioner. The new preconditioner is defined as follows:

$$\mathbf{M} = \begin{pmatrix} A_1 & -\frac{1}{\alpha} B_1^T B_2 & B_1^T \\ 0 & A_2 & B_2^T \\ -B_1 & -B_2 & \alpha I \end{pmatrix}. \tag{13}$$

By comparing the preconditioner $\mathbf{M}$ defined in (13) with the DS preconditioner $\mathbf{P}$ defined in (11), we can see that the new preconditioner no longer contains the shift terms αI appearing in the $(1,1)$ and $(2,2)$ blocks of $\mathbf{P}$. It is important to note that the preconditioner $\mathbf{M}$ can be written in factorized form as

$$\mathbf{M} = \frac{1}{\alpha} \begin{pmatrix} A_1 & 0 & B_1^T \\ 0 & \alpha I & 0 \\ -B_1 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & 0 & 0 \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & \alpha I \end{pmatrix}. \tag{14}$$

Note that both factors on the right are invertible provided that A_1, A_2 have positive definite symmetric part, hence in this case the preconditioner itself is nonsingular. This (sufficient) condition is satisfied for both Stokes and

Oseen problems. In the particular case $\alpha = 1$, the RDF preconditioner $\mathbf{M}$ reduces to

$$\tilde{\mathbf{M}} = \begin{pmatrix} A_1 & 0 & B_1^T \\ 0 & I & 0 \\ -B_1 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & I \end{pmatrix}. \tag{15}$$

By analogy with the concept of *dimensional splitting* for (9), it follows that the preconditioner $\tilde{\mathbf{M}}$ given by (15) can be regarded as a *dimensional factorization* preconditioner, and hence the preconditioner $\mathbf{M}$ given by (14) is referred to as the *Relaxed Dimensional Factorization* (RDF) preconditioner in this paper.

By comparing (13) with (6), we can see that the difference between $\mathbf{M}$ and $\mathbf{H}$ is given by

$$\mathbf{R} = \mathbf{M} - \mathbf{H} = \begin{pmatrix} 0 & -\frac{1}{\alpha} B_1^T B_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \alpha I \end{pmatrix}. \tag{16}$$

Hence, compared to DS preconditioning, now only one of the three diagonal blocks (the smallest one in size) is nonzero, while the nonzero off-diagonal block is the same for both RDF and DS preconditioning. This observation suggests that $\mathbf{M}$ may be a better preconditioner than $\mathbf{P}$, since it gives a better approximation of the system matrix $\mathbf{H}$ for the same value of α. Furthermore, the structure of (16) somewhat facilitates the analysis of the eigenvalue distribution of the preconditioned matrix. We should remark that the RDF preconditioner $\mathbf{M}$ no longer relates to an alternating direction iteration like (10). Clearly, this fact is of no consequence when $\mathbf{M}$ is used as a preconditioner for a Krylov subspace method like GMRES [36].

We now provide two lemmas which are needed in analyzing the spectral properties of the preconditioned matrix $\mathbf{T} = \mathbf{M}^{-1}\mathbf{H}$.

Lemma 1. *Let*

$$\mathbf{M}_1 = \begin{pmatrix} A_1 & 0 & B_1^T \\ 0 & \alpha I & 0 \\ -B_1 & 0 & \alpha I \end{pmatrix},$$

then we have

$$\mathbf{M}_1^{-1} = \begin{pmatrix} \hat{A}_1^{-1} & 0 & -\frac{1}{\alpha}\hat{A}_1^{-1} B_1^T \\ 0 & \frac{1}{\alpha} I & 0 \\ \frac{1}{\alpha} B_1 \hat{A}_1^{-1} & 0 & \frac{1}{\alpha} I - \frac{1}{\alpha^2} S_1 \end{pmatrix},$$

with $\hat{A}_1 = A_1 + \frac{1}{\alpha} B_1^T B_1$ *and* $S_1 = B_1 \hat{A}_1^{-1} B_1^T$.

Proof. Let

$$\mathbf{U}_1 = \begin{pmatrix} I & 0 & -\frac{1}{\alpha}B_1^T \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}, \ \mathbf{L}_1 = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \frac{1}{\alpha}B_1 & 0 & I \end{pmatrix} \text{ and } \mathbf{D}_1 = \begin{pmatrix} \hat{A}_1 & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{pmatrix},$$

then we have

$$\mathbf{U}_1\mathbf{M}_1\mathbf{L}_1 = \mathbf{D}_1.$$

Thus,

$$\begin{aligned} \mathbf{M}_1^{-1} &= \mathbf{L}_1\mathbf{D}_1^{-1}\mathbf{U}_1 \\ &= \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \frac{1}{\alpha}B_1 & 0 & I \end{pmatrix} \begin{pmatrix} \hat{A}_1^{-1} & 0 & 0 \\ 0 & \frac{1}{\alpha}I & 0 \\ 0 & 0 & \frac{1}{\alpha}I \end{pmatrix} \begin{pmatrix} I & 0 & -\frac{1}{\alpha}B_1^T \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \\ &= \begin{pmatrix} \hat{A}_1^{-1} & 0 & -\frac{1}{\alpha}\hat{A}_1^{-1}B_1^T \\ 0 & \frac{1}{\alpha}I & 0 \\ \frac{1}{\alpha}B_1\hat{A}_1^{-1} & 0 & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_1 \end{pmatrix}. \end{aligned}$$

□

Lemma 2. *Let*

$$\mathbf{M}_2 = \begin{pmatrix} \alpha I & 0 & 0 \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & \alpha I \end{pmatrix},$$

then we have

$$\mathbf{M}_2^{-1} = \begin{pmatrix} \frac{1}{\alpha}I & 0 & 0 \\ 0 & \hat{A}_2^{-1} & -\frac{1}{\alpha}\hat{A}_2^{-1}B_2^T \\ 0 & \frac{1}{\alpha}B_2\hat{A}_2^{-1} & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_2 \end{pmatrix},$$

with $\hat{A}_2 = A_2 + \frac{1}{\alpha}B_2^TB_2$ *and* $S_2 = B_2\hat{A}_2^{-1}B_2^T$.

Proof. Let

$$\mathbf{U}_2 = \begin{pmatrix} I & 0 & 0 \\ 0 & I & -\frac{1}{\alpha}B_2^T \\ 0 & 0 & I \end{pmatrix}, \ \mathbf{L}_2 = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\alpha}B_2 & I \end{pmatrix} \text{ and } \mathbf{D}_2 = \begin{pmatrix} \alpha I & 0 & 0 \\ 0 & \hat{A}_2 & 0 \\ 0 & 0 & \alpha I \end{pmatrix},$$

then we have

$$\mathbf{U}_2\mathbf{M}_2\mathbf{L}_2 = \mathbf{D}_2.$$

Thus,

$$\begin{aligned}
\mathbf{M}_2^{-1} &= \mathbf{L}_2\mathbf{D}_2^{-1}\mathbf{U}_2 \\
&= \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\alpha}B_2 & I \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha}I & 0 & 0 \\ 0 & \hat{A}_2^{-1} & 0 \\ 0 & 0 & \frac{1}{\alpha}I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & -\frac{1}{\alpha}B_2^T \\ 0 & 0 & I \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{\alpha} & 0 & 0 \\ 0 & \hat{A}_2^{-1} & -\frac{1}{\alpha}\hat{A}_2^{-1}B_2^T \\ 0 & \frac{1}{\alpha}B_2\hat{A}_2^{-1} & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_2 \end{pmatrix}. \qquad \square
\end{aligned}$$

Lemma 3. *Let*

$$T_{22} = \begin{pmatrix} \frac{1}{\alpha^2}\hat{A}_2^{-1}B_2^TS_1B_2 & -\hat{A}_2^{-1}B_2^T + \frac{1}{\alpha}\hat{A}_2^{-1}B_2^TS_1 \\ -\frac{1}{\alpha^2}S_1B_2 + \frac{1}{\alpha^3}S_2S_1B_2 & \frac{1}{\alpha^2}(\alpha I - S_2)(\alpha I - S_1) \end{pmatrix}.$$

Then the eigenvalues of T_{22} are given by 0 with multiplicity at least n_2, and the remaining eigenvalues are $1 - \mu_i$, where μ_i are the eigenvalues of the $p \times p$ matrix $Z_\alpha := \alpha^{-1}(S_1 + S_2) - 2\alpha^{-2}S_1S_2$.

Proof. Firstly, we observe that

$$\begin{aligned}
T_{22} &= \alpha^{-2} \begin{pmatrix} -\hat{A}_2^{-1}B_2^T \\ \alpha^{-1}(\alpha I - S_2) \end{pmatrix} \begin{pmatrix} -S_1B_2 & \alpha(\alpha I - S_1) \end{pmatrix} \\
&= \alpha^{-2}XY^T, \qquad (17)
\end{aligned}$$

where

$$X = \begin{pmatrix} -\hat{A}_2^{-1}B_2^T \\ \alpha^{-1}(\alpha I - S_2) \end{pmatrix} \in \mathcal{R}^{(n_2+p)\times p}$$

and

$$Y^T = \begin{pmatrix} -S_1B_2 & \alpha(\alpha I - S_1) \end{pmatrix} \in \mathcal{R}^{p\times(n_2+p)}.$$

Hence, from (17) we can see that T_{22} is a matrix with rank at most p. Therefore, T_{22} has an eigenvalue 0 of multiplicity at least n_2, and the remaining eigenvalues are the eigenvalues of the matrix

$$\begin{aligned}
\alpha^{-2}Y^TX &= \alpha^{-2}(S_1B_2\hat{A}_2^{-1}B_2^T + (\alpha I - S_1)(\alpha I - S_2)) \\
&= I + 2\alpha^{-2}S_1S_2 - \alpha^{-1}(S_1 + S_2) = I - Z_\alpha. \qquad \square
\end{aligned}$$

Based on Lemmas 1-3, we have the following result.

Theorem 1. *The preconditioned matrix $\mathbf{T}$ has an eigenvalue at 1 with multiplicity at least $n_1 + n_2$. The remaining eigenvalues are the eigenvalues μ_i of the matrix $Z_\alpha = \alpha^{-1}(S_1 + S_2) - 2\alpha^{-2}S_1S_2$.*

Proof. Indeed, we have

$$\begin{aligned}
\mathbf{T} &= \mathbf{I} - \mathbf{M}^{-1}\mathbf{R} \\
&= \mathbf{I} - \alpha\mathbf{M_2}^{-1}\mathbf{M_1}^{-1}\mathbf{R} \\
&= \mathbf{I} - \begin{pmatrix} \frac{1}{\alpha} & 0 & 0 \\ 0 & \hat{A}_2^{-1} & -\frac{1}{\alpha}\hat{A}_2^{-1}B_2^T \\ 0 & \frac{1}{\alpha}B_2\hat{A}_2^{-1} & \frac{1}{\alpha}I - \frac{1}{\alpha^2}S_2 \end{pmatrix} \begin{pmatrix} \alpha\hat{A}_1^{-1} & 0 & -\hat{A}_1^{-1}B_1^T \\ 0 & I & 0 \\ B_1\hat{A}_1^{-1} & 0 & I - \frac{1}{\alpha}S_1 \end{pmatrix} \begin{pmatrix} 0 & -\frac{1}{\alpha}B_1^TB_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \alpha I \end{pmatrix} \\
&= \mathbf{I} - \begin{pmatrix} 0 & -\alpha^{-1}\hat{A}_1^{-1}B_1^TB_2 & -\hat{A}_1^{-1}B_1^T \\ 0 & \alpha^{-2}\hat{A}_2^{-1}B_2^TS_1B_2 & -\hat{A}_2^{-1}B_2^T + \alpha^{-1}\hat{A}_2^{-1}B_2^TS_1 \\ 0 & -\alpha^{-2}S_1B_2 + \alpha^{-3}S_2S_1B_2 & \alpha^{-2}(\alpha I - S_2)(\alpha I - S_1) \end{pmatrix} \\
&= \mathbf{I} - \begin{pmatrix} 0 & T_{12} \\ 0 & T_{22} \end{pmatrix},
\end{aligned} \tag{18}$$

where $T_{12} = \begin{pmatrix} -\alpha^{-1}\hat{A}_1^{-1}B_1^TB_2 & -\hat{A}_1^{-1}B_1^T \end{pmatrix}$ and

$$T_{22} = \begin{pmatrix} \alpha^{-2}\hat{A}_2^{-1}B_2^TS_1B_2 & -\hat{A}_2^{-1}B_2^T + \alpha^{-1}\hat{A}_2^{-1}B_2^TS_1 \\ -\alpha^{-2}S_1B_2 + \alpha^{-3}S_2S_1B_2 & \alpha^{-2}(\alpha I - S_2)(\alpha I - S_1) \end{pmatrix}.$$

According to Lemma 3, the eigenvalues of T_{22} are given by 0 and $1 - \mu_i$. Therefore, from (18) we can see that the eigenvalues of $\mathbf{T}$ are given by 1 (with multiplicity at least n_1+n_2) and by the μ_i's. □

Remarks 2.1.
(1) The above results hold for left preconditioning. However, identical conclusions hold for right preconditioning ($\mathbf{AM}^{-1}$), and more generally for split preconditioning ($\mathbf{M}_1^{-1}\mathbf{AM}_2^{-1}$), since all these matrices are similar.
(2) It is possible to show that the eigenvalues μ_i are of the form

$$\mu_i = \frac{\alpha\lambda_i}{1 + \alpha\lambda_i},$$

where the λ_i's satisfy the generalized eigenvalue problem

$$\mathbf{BA}^{-1}\mathbf{B}^T\phi_i = \lambda_i(\alpha^2 I + C_1C_2)\phi_i, \quad \text{where} \quad C_k = B_kA_k^{-1}B_k^T \quad (k = 1, 2).$$

This fact can be used to derive estimates on the magnitude of the non-unit eigenvalues of the preconditioned matrix; for example, it can be used to show that they go to zero like $O(\alpha^{-1})$ for $\alpha \to \infty$. In the interest of brevity, we leave these issues for future work.
(3) Eigenvalue plots of the preconditioned matrices obtained with DS and RDF preconditioners (with optimal values of α) are displayed in Figs. 1–2. These two plots confirm that for both DS and RDF preconditioning,

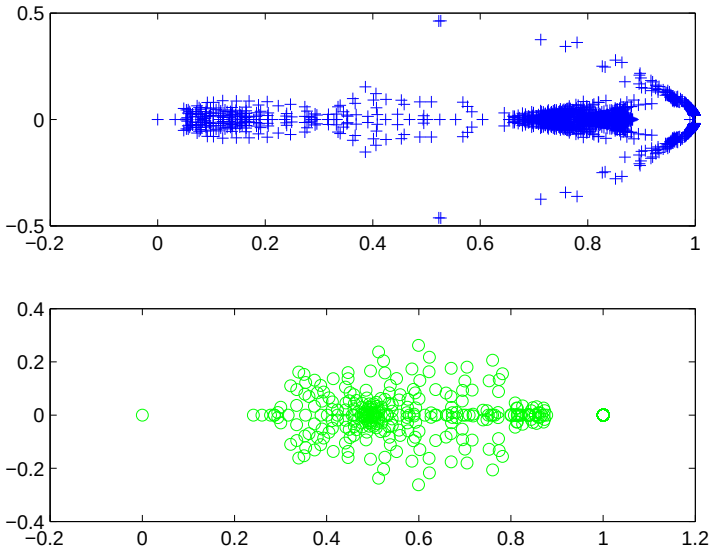

Figure 1: Spectrum of the preconditioned Oseen matrix, 32×32 grid (Q2-Q1 FEM), $\nu = 0.01$ and experimentally optimal α. Top: $\mathbf{P}^{-1}\mathbf{H}$. Bottom: $\mathbf{M}^{-1}\mathbf{H}$.

the eigenvalues of the preconditioned matrices are confined to a rectangular region in the half-plane $\mathrm{Re}(z) \geq 0$; note that the appearance of a zero eigenvalue is due to the singularity of the saddle point system (5), which is caused by the hydrostatic pressure mode [22]. This zero eigenvalue is harmless in practice and can be ignored [22, section 2.3]. In these two examples, corresponding to the viscosities $\nu = 0.01$ and $\nu = 0.001$, it is clear that RDF produces a much more favorable eigenvalue distribution than DS. Indeed, in these examples the DS preconditioner fails to force many of the eigenvalues away zero (especially in the case of $\nu = 0.001$), which indicates that the GMRES method preconditioned by DS preconditioner may converge slowly. In contrast, the RDF preconditioner is able to cluster most of the eigenvalues at 1. Indeed, according to Theorem 1, there are at least 2178 eigenvalues equal to 1 in this case, and the plots show that the remaining nonzero eigenvalues are well separated from the origin. The clustering of the spectrum obtained with DS preconditioning can be greatly improved by diagonally scaling $\mathbf{H}$ prior to applying the DS preconditioner (see [8]), but it is interesting to see that for these examples RDF achieves excellent clustering without the need for scaling.

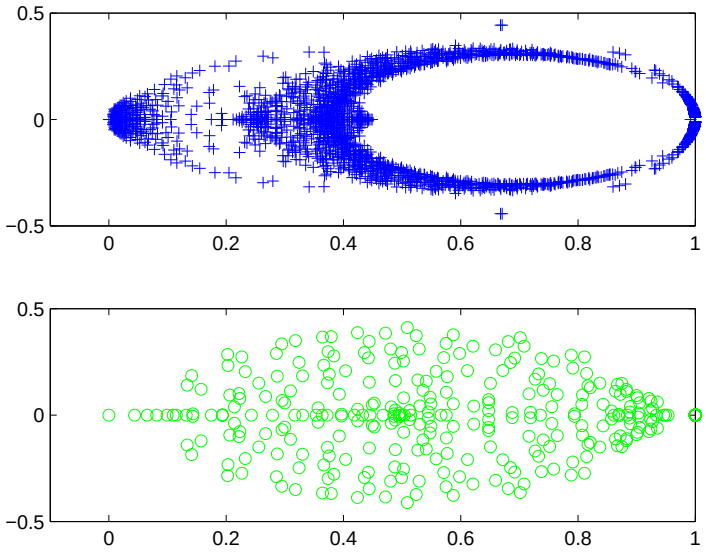

Figure 2: Spectrum of the preconditioned Oseen matrix, 32×32 grid (Q2-Q1 FEM), $\nu = 0.001$ and experimentally optimal α. Top: $\mathbf{P}^{-1}\mathbf{H}$. Bottom: $\mathbf{M}^{-1}\mathbf{H}$.

2.3. Practical implementation of the RDF Preconditioner

In this section we outline the practical implementation of the RDF preconditioner in a Krylov subspace iterative method, such as GMRES. The main step is applying the preconditioner, i.e., solving linear systems with coefficient matrix $\mathbf{M}$. The RDF preconditioner can be factorized as follows:

$$\begin{aligned}
\mathbf{M} &= \begin{pmatrix} A_1 & 0 & \frac{B_1^T}{\alpha} \\ 0 & I & 0 \\ -B_1 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & A_2 & B_2^T \\ 0 & -B_2 & \alpha I \end{pmatrix} \\
&= \begin{pmatrix} I & 0 & \frac{B_1^T}{\alpha} \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} \hat{A}_1 & 0 & 0 \\ 0 & I & 0 \\ -B_1 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \hat{A}_2 & B_2^T \\ 0 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{-B_2}{\alpha} & I \end{pmatrix},
\end{aligned}$$

showing that the preconditioner requires solving two linear systems at each step, with coefficient matrices $\hat{A}_1 = A_1 + \frac{1}{\alpha} B_1^T B_1$ and $\hat{A}_2 = A_2 + \frac{1}{\alpha} B_2^T B_2$. Note that these systems can be interpreted as discretizations of anisotropic scalar elliptic boundary value problems of convection-diffusion type (for the unsteady case, the equations are of reaction-convection-diffusion type).

These linear systems can be solved using a state-of-the-art sparse direct solver. In Matlab, this corresponds to computing the Cholesky (for Stokes) or LU (for Oseen) factorization of $\hat{A}_i$ ($i = 1 : d$) in combination with AMD or column AMD reordering [1]. For 3D problems, however, the cost of factoring the relevant matrices quickly becomes prohibitive in terms of computational cost and storage as the mesh is refined, and iterative methods are required. Fortunately, in most cases the effectiveness of the preconditioner is not compromised if the linear systems associated with $\hat{A}_i$ are solved inexactly, and in practice a rather low accuracy is often sufficient. In subsection 3.4 we investigate the effect of inexact solves performed with an inner GMRES iteration preconditioned with an incomplete LU (ILU) factorization [5, 36] and with an Algebraic Multigrid (AMG) method [14, 37]. For sufficiently large problems, this turns out to be a competitive alternative to using direct solvers, even in 2D.

2.4. The effect of scaling

In [8] it was pointed out that the performance of DS preconditioning is strongly affected by diagonal scaling. Without scaling, DS preconditioning performs poorly for small values of the viscosity ν on both uniform and stretched grids. For stretched grids the performance turns out to be poor (in particular, h-dependent) even for the Stokes problem and for the Oseen problem with relatively large values of ν. However, it was shown in [8] that with the appropriate diagonal scaling the DS preconditioner performs very well on uniform grids, where the rate of convergence is h-independent and only moderatly affected by the viscosity value; on stretched grids the combination of scaling and DS preconditioning results in h-independent convergence rates, although the performance tends to deteriorate for very small ν. The scaling matrix that was found to give the best results in [8] was

$$\mathbf{D} = \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{pmatrix},$$

where $\mathrm{diag}\,(D_1, D_2)$ is the main diagonal of the velocity mass matrix $\mathbf{M}_v$ (of size $(n_1 + n_2) \times (n_1 + n_2)$) and D_3 is the main diagonal of the pressure mass matrix M_p (of size $m \times m$). The scaled system is then $\widehat{\mathbf{H}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, where $\widehat{\mathbf{H}} = \mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$, $\hat{\mathbf{x}} = \mathbf{D}^{1/2}\mathbf{x}$, and $\hat{\mathbf{b}} = \mathbf{D}^{-1/2}\mathbf{b}$. The DS preconditioner is then applied to this scaled system.

It is natural to apply a similar scaling strategy in conjunction with the RDF preconditioner. It turns out that on both uniform and stretched grids,

the scaling has little or no effect on RDF preconditioning. For this reason we do not use any scaling in the numerical tests with RDF.

2.5. Estimation of the optimal α using Fourier analysis

In this section we describe an inexpensive technique for approximating the optimal value of the parameter α using a simplified form of Fourier analysis. As usual, the use of Fourier analysis requires several simplifications and assumptions on the problem being solved. First, the Oseen problem is assumed to have constant coefficients, to be defined on the unit square with periodic boundary conditions, and to be discretized on a uniform grid, with grid size $h = 1/l$. Moreover, we assume that the matrices A_1, A_2, B_1, B_2 are all square (of the same order) and commute. Though some of the assumptions are generally not consistent with real problems, we emphasize that these assumptions are made to guide in the choice of α, and that it is often the case that parameters determined by Fourier analysis give good results for more general problems that are defined on more general domains or do not have periodic boundary conditions (see [20] for an example in a similar context).

To further simplify the description of the Fourier-based approach, we replace the unit square (or cube) with the unit interval $[0, 1]$ and the 2D (or 3D) differential operators with 1D surrogates. Thus, the vector convection-diffusion operator $-\nu\Delta + \mathbf{v} \cdot \nabla$ is replaced by a direct sum of d $(= 2, 3)$ 1D convection-diffusion operators of the form $-\nu\frac{d^2}{dx^2} + \frac{d}{dx}$, and the divergence operator $\text{div} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$ becomes $\left(\frac{d}{dx}, \frac{d}{dy}\right)$.

The assumptions above imply that the $l \times l$ matrices A_1, A_2, B_1 and B_2 are all diagonalizable by the discrete Fourier transform, i.e., $XA_1X^H = D_1 = \text{diag}(a_1)$, $XA_2X^H = D_2 = \text{diag}(a_2)$, $XB_1X^H = D_3 = \text{diag}(b_1)$, $XB_2X^H = D_4 = \text{diag}(b_2)$. Here a_1, a_2, b_1, b_2 are vectors of length l containing the eigenvalues of the corresponding matrices, and the unitary matrix X is composed of Fourier components

$$\frac{\mathrm{e}^{\mathrm{i}2\pi h\theta k}}{\sqrt{l}}, \quad \text{with} \quad \mathrm{i} = \sqrt{-1}, \quad k = 1, 2, \ldots, l \quad \text{and} \quad \theta = 1, 2, \ldots, l.$$

In the following, to simplify the notation, we use a_1 to denote the generic diagonal entry of D_1, although in reality there are of course l such entries, and likewise for a_2, b_1, and b_2. In other words, we use a_1, a_2, b_1, b_2 to denote the generic eigenvalues of A_1, A_2, B_1, B_2. Furthermore, from $S_1 = B_1(A_1 + \frac{1}{\alpha}B_1^TB_1)^{-1}B_1^T$, $S_2 = B_2(A_2 + \frac{1}{\alpha}B_2^TB_2)^{-1}B_2^T$ and $Z_\alpha = \alpha^{-1}(S_1 + S_2) - 2\alpha^{-2}S_1S_2$, it follows that S_1, S_2 and Z_α can also be diagonalized by

X, and their eigenvalues can be represented in terms of a_1, a_2, b_1 and b_2. To be more specific, the eigenvalues of S_1, S_2 and Z_α can be expressed as

$$s_1 = \frac{b_1^2}{a_1 + \frac{1}{\alpha} b_1^2}, \quad s_2 = \frac{b_2^2}{a_2 + \frac{1}{\alpha} b_2^2}, \quad z = \frac{1}{\alpha}(s_1 + s_2) - \frac{2}{\alpha^2} s_1 s_2 \,.$$

From Theorem 1 we expect that clustering the eigenvalues of Z_α around 1 could lead to fast convergence of the RDF-preconditioned iteration. This can be achieved by choosing α so as to cluster the values of z around 1.

Assuming that both the diffusion and convection terms are discretized by centered finite differences, we find for the symbols of the discrete convection-diffusion operators the expression

$$a_1 = a_2 = \nu \left(2 - \mathrm{e}^{\mathrm{i}2\pi h\theta} - \mathrm{e}^{-\mathrm{i}2\pi h\theta}\right) + h \left(\mathrm{e}^{\mathrm{i}2\pi h\theta} - \mathrm{e}^{-\mathrm{i}2\pi h\theta}\right), \quad \theta = 1, 2, \ldots, l.$$

Similarly, using one-sided finite differences for the first order derivatives in the conservation equation leads to the symbols

$$b_1 = b_2 = h \left(1 - \mathrm{e}^{-\mathrm{i}2\pi h\theta}\right), \quad \theta = 1, 2, \ldots, l.$$

Notice that in terms of their dependence on h, the scaling of a_1, a_2, b_1 and b_2 respects the scaling of the matrices obtained from finite elements discretization. Then $z = z(\alpha)$ can be written in terms of $\theta = 1, 2, \ldots, l$, and we need to solve the following optimization problem:

$$\min_{\alpha > 0} \quad |z - 1| \tag{19}$$

$$\text{s. t.} \quad \theta = 1, 2, \ldots, l \tag{20}$$

to determine the value of α. In practice, we solve the optimization problem by brute force. That is, we test α over the range of $[0.0001, 1]$ (with step 0.001) and over the range $[1, 30]$ (with step 0.1) and we pick the value of α that minimizes the average of $|z-1|$. (Numerical experiments show that the optimal α lies in the interval [0.0001, 30] for most problem parameters and discretizations of interest.) This computation is inexpensive and imposes almost no overhead compared with the cost of solving the Stokes (or Oseen) problem.

Finally, we note that when dealing with unsteady problems using finite elements we also have to take into account the mass matrices. For the purpose of the Fourier analysis, we use h^2 as the corresponding symbol in 2D, and h^3 in 3D.

3. Numerical examples

In this section we present the results of numerical experiments on saddle point systems arising from linearization and discretization of incompressible flow problems. We consider both the Stokes problem and the Oseen problem. Both steady and unsteady flows are considered; for unsteady cases, we let $\sigma = h^{-1}$ in (7). We focus mainly on the 2D lid-driven cavity problem discretized by Q2-Q1 and Q2-P1 finite elements on uniform grids [22]. In the case of the Oseen problem on uniform grids, for each viscosity value $\nu = 0.1$, $\nu = 0.01$ and $\nu = 0.001$, we generate linear systems corresponding to 16×16, 32×32, 64×64 and 128×128 meshes. A few additional tests are performed using stretched grids. We also consider the backward facing step problem discretized by Q2-Q1 finite elements using uniform grids. For the step problem, the number of cells in the two directions x and y is unequal. All these test problems are generated by using the IFISS software package [21]. Calculations are performed in Matlab on an AMD Athlon 64x2 dual core processor 4800+ (2.51GHz, 1.37Gb of RAM). We also include a few experiments using a 3D Marker-and-Cell (MAC, see [28]) discretization of the Oseen problem. These experiments were performed in Matlab on a Sun Microsystems SunFire V40z with 4 Dual Core AMD Opteron processors and 32 GB of memory.

Unless otherwise specified, we use right preconditioning with restarted GMRES as the Krylov subspace method, with the maximum subspace dimension set to 20. In the tests, we always use a uniformly distributed random vector as the initial approximate solution. The iteration stops when

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq 10^{-6},$$

where r_k is the residual vector at the kth iteration. In the following tables, we use its to denote the number of iterations required to converge, and α_{opt} to denote the experimentally optimal parameter.

3.1. The leaky lid driven cavity problem discretized by Q2-Q1 finite elements

Experimental results for the steady case are displayed in Tables 1–5; the latter table is for stretched grids (with the default parameter setting in IFISS), the remaining ones for uniform grids. Results for the unsteady problems are reported in Tables 6–9. For DS, diagonal scaling is used, as described in subsection 2.4. The dependence of the RDF preconditioned GMRES on the parameter α is illustrated in Figs. 3–4; these results are

Table 1: Preconditioned GMRES(20) on steady Stokes problems with different grid size (Q2-Q1 FEM, uniform grids).

	scaled DS optimal		RDF optimal	
Grid	α_{opt}	its	α_{opt}	its
16×16	0.006	11	0.006	12
32×32	0.001	12	0.002	13
64×64	0.0006	12	0.002	11
128×128	0.0002	10	0.0005	11

Table 2: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.1$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.03	14	0.05	11	0.07	11
32×32	0.01	14	0.01	11	0.025	11
64×64	0.002	15	0.005	10	0.0008	11
128×128	0.0008	14	0.002	10	0.002	10

Table 3: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.01$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.2	26	0.2	14	0.12	16
32×32	0.05	27	0.07	13	0.057	14
64×64	0.02	25	0.025	11	0.024	11
128×128	0.005	23	0.007	10	0.001	10

Table 4: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.001$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.8	45	0.55	27	0.13	63
32×32	0.2	54	0.15	30	0.063	53
64×64	0.07	52	0.05	30	0.032	36
128×128	0.025	42	0.02	30	0.016	30

Table 5: Iteration counts for RDF-preconditioned GMRES(20) on steady Oseen problems (Q2-Q1 FEM, stretched grids), different viscosities, optimal α.

Grid	$\nu = 0.1$	$\nu = 0.01$	$\nu = 0.001$
16×16	15	19	28
32×32	24	30	37
64×64	34	43	41
128×128	31	42	54

Table 6: Preconditioned GMRES(20) on generalized Stokes problems with different grid size (Q2-Q1 FEM, uniform grids).

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	11	0.006	10	0.006
32×32	12	0.001	11	0.002
64×64	11	0.0005	11	0.0005
128×128	12	0.0001	11	0.0001

Table 7: Preconditioned GMRES(20) on generalized Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.1$.

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	11	0.01	10	0.01
32×32	11	0.004	11	0.005
64×64	11	0.001	11	0.001
128×128	10	0.001	10	0.001

Table 8: Preconditioned GMRES(20) on generalized Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.01$.

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	11	0.02	10	0.02
32×32	12	0.015	11	0.01
64×64	11	0.003	10	0.03
128×128	10	0.005	8	0.002

Table 9: Preconditioned GMRES(20) on generalized Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.001$.

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	11	0.015	10	0.025
32×32	12	0.001	10	0.01
64×64	11	0.003	10	0.003
128×128	10	0.005	8	0.002

Table 10: Iteration counts for GMRES(20) with RDF preconditioner on Oseen problems (Q2-Q1 FEM) generated in each Picard iteration, 128×128 grid, different viscosities.

	$\nu = 0.1$		$\nu = 0.01$		$\nu = 0.001$	
Picard	its	α_{opt}	its	α_{opt}	its	α_{opt}
1	10	0.001	10	0.01	30	0.02
2	10	0.001	10	0.01	29	0.02
3	10	0.001	10	0.01	28	0.02
4	10	0.001	10	0.01	28	0.02
5	10	0.001	10	0.01	28	0.02
6	10	0.001	10	0.01	28	0.02
7	10	0.001	10	0.01	28	0.02

from tests on two representative steady Oseen problems with $\nu = 0.01$ and $\nu = 0.001$, using 16×16 and 32×32 grids.

Remarks 3.1.
(1) From these tables, we can observe that the RDF preconditioner leads to similar or slightly better numerical results than the DS preconditioner on all the Stokes problems, generalized Oseen problems, and steady Oseen problems with $\nu = 0.1$. Note that the results for the value of α obtained by Fourier analysis ('FA estimate' in Tables 2-4) are essentially optimal.
(2) For the cases $\nu = 0.01$ and $\nu = 0.001$, the RDF preconditioner clearly outperforms the scaled DS preconditioner. Note that for $\nu = 0.001$, Fourier analysis accurately estimates the optimal α only if the mesh is fine enough to yield physical solutions. For all cases with uniform grids, we can see that both DS and RDF preconditioned GMRES are essentially independent of h. Indeed, for $\nu = 0.001$ the rate of convergence tends to improve as the mesh is refined.
(3) For the steady Oseen problem on stretched grids (Table 5), we observe a slight deterioration in the performance of RDF preconditioning, especially for the case of small viscosity. For $\nu = 0.1$ and $\nu = 0.01$ faster convergence is attained with DS preconditioning, provided diagonal scaling is used; see Table 5 in [8]. However, RDF is vastly superior to DS for the more difficult case $\nu = 0.001$.
(4) An important property of the RDF preconditioner is that both the optimal α and the performance of the preconditioner remain virtually unchanged throughout the solution of the Navier–Stokes equation by Picard iteration.

Table 11: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-P1 FEM, uniform grids), viscosity $\nu = 0.1$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.06	16	0.12	9	0.07	10
32×32	0.02	17	0.03	8	0.025	9
64×64	0.005	16	0.01	8	0.008	8
128×128	0.001	17	0.003	8	0.002	8

Table 12: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-P1 FEM, uniform grids), viscosity $\nu = 0.01$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.22	26	0.5	14	0.12	24
32×32	0.08	27	0.17	13	0.057	19
64×64	0.021	25	0.04	11	0.024	14
128×128	0.008	23	0.015	10	0.01	10

For the Q2-Q1 discretization of the lid driven cavity problem on a 128×128 grid, this phenomenon is illustrated in Table 10, which displays the optimal value of α and the number of linear iterations required at each of the seven Picard steps needed to solve the Navier–Stokes equations with $\nu = 0.1, 0.01$ and $\nu = 0.001$.

(5) From Figs. 3–4 we can see that the RDF preconditioner is not overly sensitive to the value of the parameter α, in the sense that the iteration number does not change dramatically near the experimental optimal α. We observe that there is a fairly wide range of values of the parameter α that produce similar convergence results. In the case $\nu = 0.001$, we see a slightly more pronounced sensitivity to α.

3.2. The leaky lid driven cavity problem discretized by Q2-P1 finite elements

Here we present results of some tests on problems generated from the discretization of steady Oseen problems using Q2-P1 elements. The numerical results are summarized in Tables 11–15. The first three of these tables are for steady problems, the last two for unsteady problems. One can

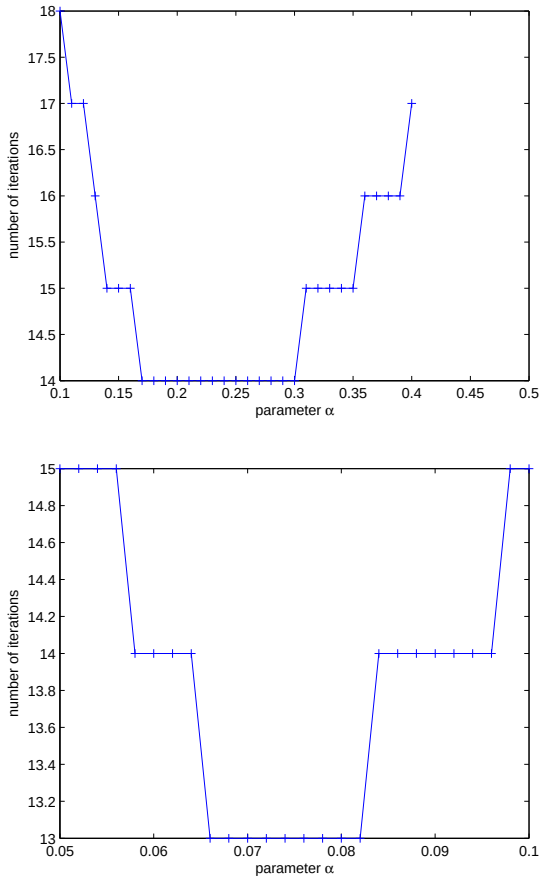

Figure 3: Number of iterations vs. parameter α, RDF preconditioned GMRES(20), steady Oseen problem, with $\nu = 0.01$. Top: 16×16 grid. Bottom: 32×32 grid (Q2-Q1 FEM, uniform grids).

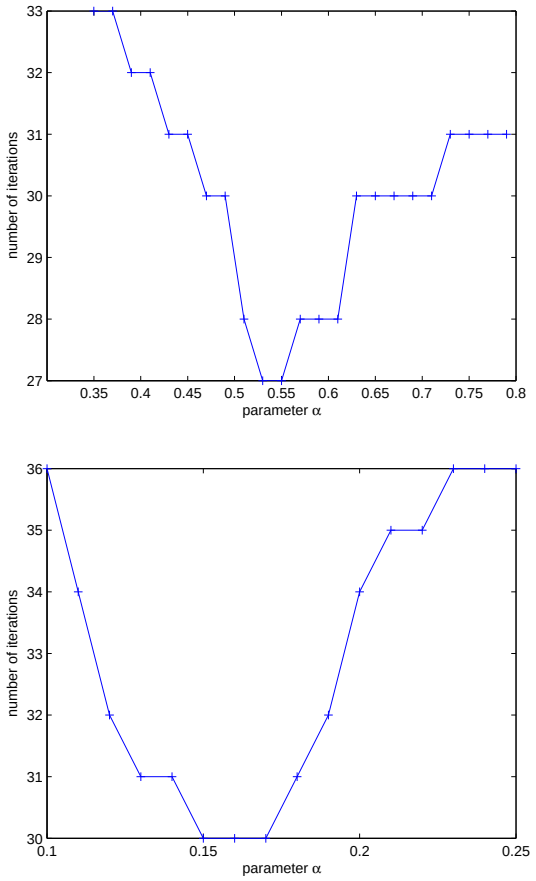

Figure 4: Number of iteration vs. parameter α, RDF preconditioned GMRES(20), steady Oseen problem, $\nu = 0.001$. Top: 16×16 grid. Bottom: 32×32 grid (Q2-Q1 FEM, uniform grids).

Table 13: Preconditioned GMRES(20) on steady Oseen problems with different grid size (Q2-P1 FEM, uniform grids), viscosity $\nu = 0.001$.

	scaled DS optimal		RDF optimal		RDF FA estimate	
Grid	α_{opt}	its	α_{opt}	its	α_F	its
16×16	0.90	71	1.3	28	0.13	99
32×32	0.05	59	0.38	30	0.063	86
64×64	0.07	61	0.1	32	0.032	57
128×128	0.03	55	0.03	30	0.016	34

Table 14: Preconditioned GMRES(20) on generalized Oseen problems with different grid size (Q2-P1 FEM, uniform grids), viscosity $\nu = 0.01$.

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	11	0.01	10	0.01
32×32	11	0.005	11	0.005
64×64	12	0.001	11	0.001
128×128	10	0.0005	9	0.0005

see from these tables that both DS and RDF preconditioning deliver rates of convergence that are essentially h-independent. Moreover, we see again that RDF outperforms DS, both with the optimal α and with the α value provided by Fourier analysis. Once again, the Fourier estimate is almost optimal, provided the mesh is sufficiently fine (if ν is small).

For RDF, the dependence of iteration numbers on the parameter α is displayed in Figs. 5–6. Similar to the dependence illustrated in Figs. 3–4, we again observe that the RDF preconditioner is robust with respect to the choice of parameter. Indeed, there is a fairly wide range of parameters that lead to nearly optimal iteration counts.

3.3. Results for the backward facing step problem

Here we present some results for a backward facing step problem (steady Oseen equations) with viscosity $\nu = 0.005$ (the flow is unsteady for $\nu \approx 0.001$). From Table 16 we can see that both the DS and the RDF preconditioner are almost independent of grid size; once again, the convergence rate of the preconditioned GMRES method using RDF is significantly better

Table 15: Preconditioned GMRES(20) on generalized Oseen problems with different grid size (Q2-P1 FEM, uniform grids), viscosity $\nu = 0.001$.

	scaled DS optimal		RDF optimal	
Grid	its	α_{opt}	its	α_{opt}
16×16	10	0.012	10	0.01
32×32	11	0.005	11	0.005
64×64	12	0.001	11	0.001
128×128	11	0.0004	11	0.0005

Table 16: Preconditioned GMRES(20) on backward facing step Oseen problem with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.005$.

	scaled DS optimal		RDF optimal	
Grid	α_{opt}	its	α_{opt}	its
16×48	0.17	34	0.15	24
32×96	0.06	37	0.07	23
64×192	0.018	40	0.02	26

than that obtained with DS.

3.4. The use of inexact solvers

In all the numerical tests described so far, the linear systems arising at each application of the preconditioner were solved 'exactly' by means of sparse direct methods. The matrices $\hat{A}_1$ and $\hat{A}_2$ were reordered and factored once and for all at the outset, and triangular solves with the resulting sparse factors were performed at each step. Because of the rapidly increasing cost of the factorizations, this approach does not scale well as the problem size increases. For 3D problems the scaling is even worse, and we are unable to solve such problems even for moderately fine grids due to storage limitations. Therefore, the use of inexact (iterative) solvers needs to be investigated. In this subsection, we examine the influence of an inexact inner solver on the convergence of the outer iteration.

First, we consider steady Oseen equations (driven cavity problem) with viscosity $\nu = 0.01$ and $\nu = 0.001$ discretized by Q2-Q1 FEM on uniform grids. Only the two larger problems (64×64 and 128×128 grids) are considered. For the inner iterations we use GMRES preconditioned by ILU(0) to

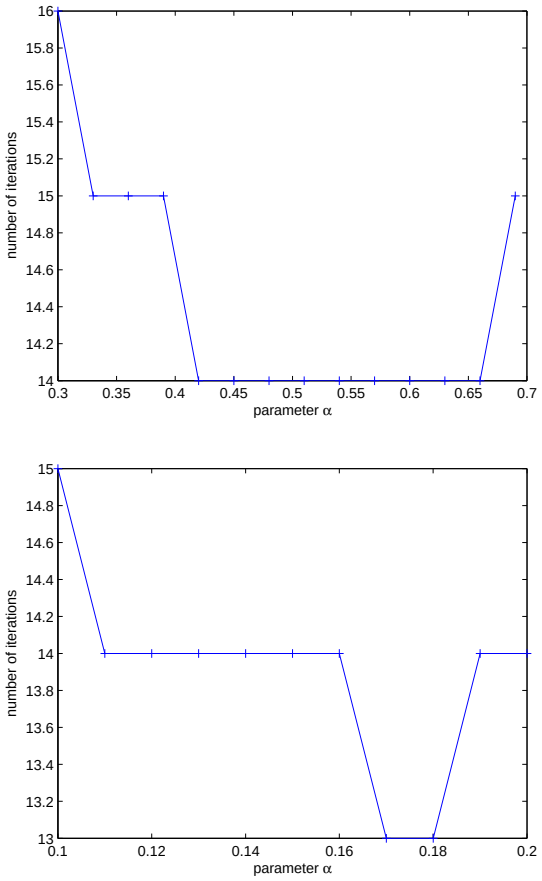

Figure 5: Iteration numbers vs. parameter α, RDF preconditioned GMRES(20) on steady Oseen problem, $\nu = 0.01$. Top: 16×16 grid. Bottom: 32×32 grid (Q2-P1 FEM, uniform grids).

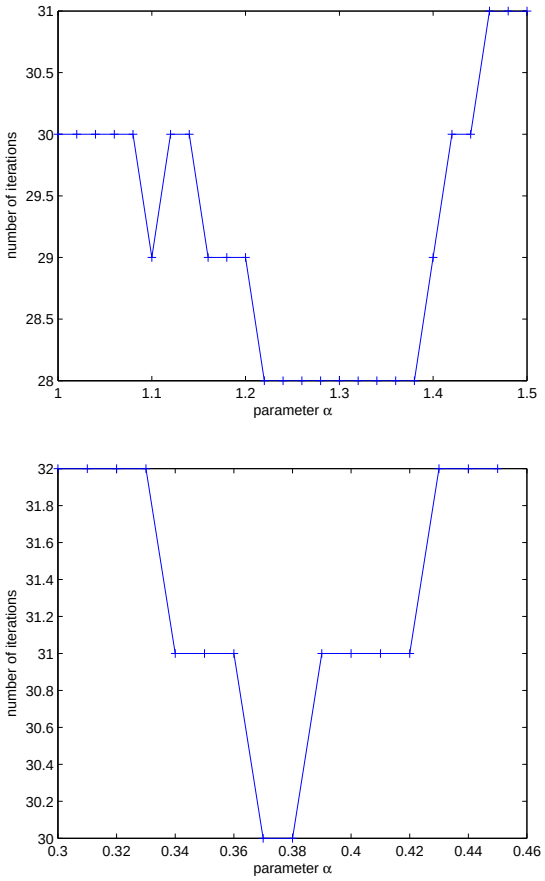

Figure 6: Iteration numbers vs. parameter α, RDF preconditioned GMRES(20) on steady Oseen problem, $\nu = 0.001$. Top: 16×16 grid. Bottom: 32×32 grid (Q2-P1 FEM, uniform grids).

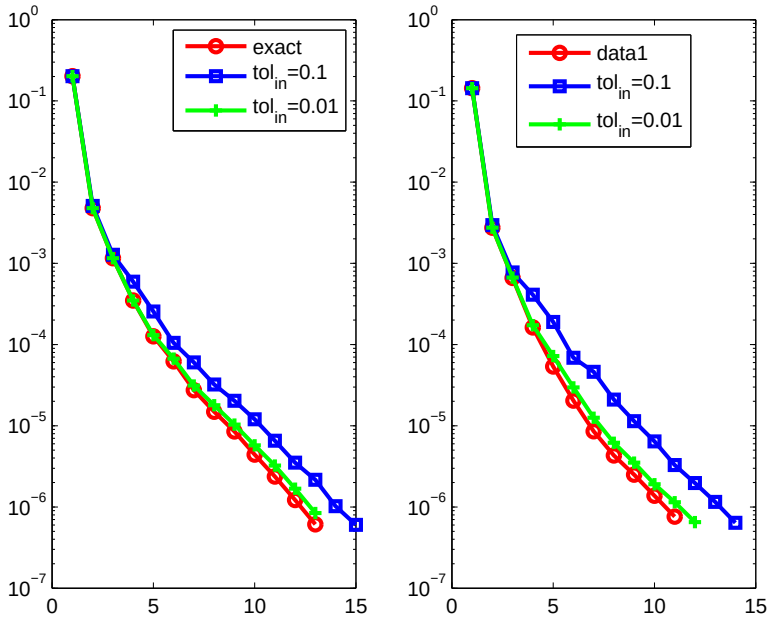

Figure 7: The convergence curves by using the exact solver and inexact solver tested on steady Oseen problem with $\nu = 0.01$. Left: 64×64 grid. Right: 128×128 grid.

solve the linear systems associated with $\hat{A}_1$ and $\hat{A}_2$. Of course, this strategy means that a flexible GMRES (FGMRES) algorithm with right preconditioning is used instead of GMRES as the outer solver [36]. The tolerance of the inner iterations is denoted by tol_{in}. Convergence curves for the cases $\nu = 0.01$ and $\nu = 0.001$ are depicted in Figs. 7-8, respectively. We can observe from these figures that setting the inner tolerance to 0.1 is enough to obtain comparable results to those obtained with the exact solver. Convergence results including timing comparisons are summarized (for $\nu = 0.001$) in Table 17. From this table we can see that for small problems, using exact solves is faster than using inexact ones. However, as the problem dimension increases, the inexact solver becomes more advantageous, and for an inner tolerance 0.1 it becomes faster already for the 64×64 grid. Although still not quite scalable, we see that the inexact method scales better than the exact one and leads to considerable savings in execution times for the largest problem. It should be kept in mind that in Matlab, the built-in functions implementing iterative methods like GMRES, unlike the sparse direct solvers (backslash), are not highly optimized.

Next, we solve 3D steady and unsteady Oseen problems discretized by Marker-and-Cell [28] using FGMRES(50) with the RDF preconditioner. The

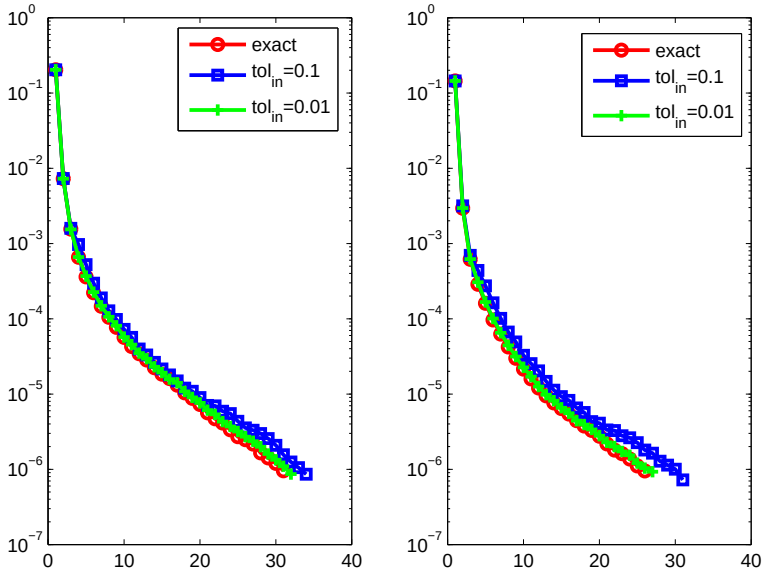

Figure 8: The convergence curves by using the exact solver and inexact solver tested on steady Oseen problem with $\nu = 0.001$. Left: 64×64 grid. Right: 128×128 grid.

Table 17: Comparison of exact vs. inexact inner solvers, RDF preconditioned GMRES(20) or FGMRES, Oseen problem (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.001$.

	$tol_{in} = 0.1$		$tol_{in} = 0.01$		Exact	
Grid	*its*	*cpu*	*its*	*cpu*	*its*	*cpu*
16×16	31	0.35	28	0.35	27	0.21
32×32	32	0.86	30	0.99	30	0.48
64×64	31	2.72	31	4.22	30	3.61
128×128	30	18.68	26	30.00	25	33.76

subproblems with $A_i + \frac{1}{\alpha} B_i^T B_i$ $(i = 1, 2, 3)$ are solved by GMRES with algebraic multigrid (AMG) preconditioning with a relative tolerance 10^{-2} and a maximum number of iterations equal to 10. The flexible GMRES implementation used is the one described in [30] based on the simple GMRES algorithm in [38]. The AMG implementation used is the one included in IFISS 3.0 and described in [14]. We choose point damped Jacobi (PDJ) as the smoother in AMG, with damping parameter 0.5. The interpolation is Stüben's direct interpolation method. In Tables 18-19 we report results obtained for the the unsteady case using the optimal α, respectively. Note

Table 18: FGMRES iteration counts and timings for 3D steady Oseen problem.

Grid	$\nu = 0.1$	$\nu = 0.01$	$\nu = 0.005$
	its (α_{opt})	its (α_{opt})	its (α_{opt})
$16 \times 16 \times 16$	11 (1.9-2.3)	17 (15-19)	25 (22)
Iter time	2.86	2.28	4.65
$32 \times 32 \times 32$	12 (3.1-5.8)	18 (13-19)	24 (23 - 26)
Iter time	12.16	26.46	44.02
$48 \times 48 \times 48$	12 (1.9-2.4)	18 (14-18)	25 (21 - 29)
Iter time	54.89	100.16	161.02
$64 \times 64 \times 64$	13 (1.4-3.1)	18 (14-18)	25 (25 - 27)
Iter time	155.69	239.34	414.70

Table 19: FGMRES iteration counts and timings for 3D unsteady Oseen problem.

Grid	$\nu = 0.1$	$\nu = 0.01$	$\nu = 0.005$
	its (α_{opt})	its (α_{opt})	its (α_{opt})
$16 \times 16 \times 16$	12 (1.1 -1.5)	18(3.1 - 7.3)	20 (5 - 8)
Iter time	1.05	1.69	1.69
$32 \times 32 \times 32$	13 (0.9)	22 (2.1 - 2.5)	24 (3.5)
Iter time	13.70	17.67	18.23
$48 \times 48 \times 48$	14 (0.7)	23 (1.7 - 1.9)	26 (3)
Iter time	42.43	63.85	82.53
$64 \times 64 \times 64$	15 (0.7)	22 (1.7)	27 (2.5 - 3)
Iter time	125.18	165.96	252.93

that the optimal α is now much bigger than in the case of finite elements, due to the different scaling of the matrix entries. As in the 2D case, we set $\sigma = h^{-1}$ for the reciprocal of the time step in (7). Once again, our results show that RDF is able to achieve convergence rates that are essentially h-independent and only mildly dependent on ν. Because of the size of the matrices, the finest grid we are able to generate is $64 \times 64 \times 64$, for a total of $N = 1,048,576$ degrees of freedom. The smallest value of the viscosity we consider is $\nu = 0.005$, since smaller values would require finer grids to give physically meaningful solutions.

The reported timings are for the iterative solution phase only, showing fairly good scalability. Note that timings are generally smaller for the unsteady case, which is what one would expect since the subsystems to be solved at each step tend to be more diagonally dominant than in the steady

case. We observed that in all cases, no more than 2-3 AMG-preconditioned GMRES iterations are needed to satisfy the (loose) convergence criterion for the inner iterations. The AMG set-up times, on the other hand, appear not to scale for these problems, and we do not include them in the tables.

3.5. Comparison with other preconditioners

In this subsection we briefly compare RDF preconditioning with some state-of-the-art preconditioners available in the literature. In particular, we report on numerical experiments with the block triangular preconditioners discussed in [22, 24], i.e., the pressure convection diffusion preconditioner PCD, the modified version mPCD described in [24], and the least squares commutator preconditioner LSC. We use IFISS 3.0 to carry out the experiments with these preconditioners. It should be kept in mind that in IFISS, no restarting is used with GMRES, therefore the results presented here are for full GMRES. Moreover, exact solves are done by backslash, which is rather inefficient since the corresponding matrices are factored anew at each solve, rather than reusing the triangular factors computed after the first application of the preconditioners. For these reasons, we do not include timings for these preconditioners. However, the actual cost of a single iteration of preconditioned GMRES is of the same order for all methods considered here when the preconditioners are applied 'exactly'. The *inexact* variants of the PCD, mPCD and LSC preconditioners (that is, the variants of these preconditioners that use inexact solves) are based on the AMG code already referred to; however, only a single AMG iteration is applied to approximately invert A and the approximate Schur complement. Hence, there is no need for a flexible outer iteration. Unfortunately, we found that for $\nu = 0.001$ this strategy is 'too inexact' and full GMRES does not converge within 300 iterations with any of the preconditioners PCD, mPCD, or LSC. For this reason, we do not present results for the inexact variants of these preconditioners. Generally speaking, even for larger values of ν the use of inexact solves seems to impact these preconditioners more than RDF.

We use the PCD, LSC and mPCD preconditioners to solve the Oseen equations from the third Picard iteration for the lid driven cavity problem. (The initial iteration corresponds to the Stokes problem, so this is the second Picard iteration following that.) The discretization is obtained with Q2-Q1 elements. For $\nu = 0.1$ and $\nu = 0.01$, the performance of the two preconditioners PCD and mPCD is generally quite good. These solvers result in h-independent convergence with iteration counts only slightly higher than those obtained with RDF. The LSC preconditioner is somewhat less efficient, displaying a mild dependence on h. (In the interest of brevity, we

Table 20: Iteration counts for preconditioned full GMRES on steady Oseen problems with different grid size (Q2-Q1 FEM, uniform grids), viscosity $\nu = 0.001$. Exact solves.

Grid	PCD	LSC	mPCD
16×16	81	56	80
32×32	104	79	105
64×64	119	90	110
128×128	104	86	99

Table 21: Iteration counts for preconditioned full GMRES on steady Oseen problems with different grid size (Q2-Q1 FEM, stretched grids), viscosity $\nu = 0.001$. Exact solves.

Grid	PCD	LSC	mPCD
16×16	79	50	81
32×32	105	78	201
64×64	117	117	135
128×128	117	174	144

do not report these results here.) For $\nu = 0.001$, on the other hand, all these three preconditioners require a significantly higher number of iterations than RDF; see Table 20, which should be compared with Table 4.

In Table 21 we show iteration counts for Q2-Q1 discretizations of the lid driven cavity problem on stretched grids, again for $\nu = 0.001$. Comparing these iteration counts with those for RDF given in Table 5, we see again that all three preconditioners require a significantly higher number of iterations than RDF.

In summary, it appears that RDF preconditioning compares favorably, in terms of both robustness and effectiveness, with some of the best existing preconditioners. The new preconditioner appears to be especially advantageous for small values of the viscosity, both for uniform and stretched grids.

4. Conclusions and future work

We have introduced a novel relaxed dimensional factorization preconditioner for solving saddle point systems. Although the preconditioner can be applied to rather general linear systems in saddle point form, in this paper we have focused on discretizations of systems of PDEs arising in incompressible fluid flow simulations. Some results on the eigenvalues of the preconditioned matrices have been obtained, and an inexpensive technique

for estimating the relaxation parameter has been described based on Fourier analysis. Unlike several other available techniques for incompressible flow problems, the RDF preconditioner does not require any approximation to the (pressure) Schur complement.

Numerical experiments on a variety of test cases indicate very fast convergence of RDF-preconditioned GMRES independent of mesh size in the case of uniform grids. The convergence rate is only mildy dependent on the viscosity ν, and appears to be much less sensitive to it than other current approaches (including the closely related DS preconditioner). The convergence behavior is also quite good for problems posed on stretched grids. In spite of some deterioration of the preconditioner quality in the steady case for very low viscosity values, RDF appears to be very competitive when compared to some of the best existing methods.

Efficient implementation of the RDF preconditioner in 3D requires the use of inexact (inner) iterative solves. Our experiments indicate that the excellent convergence properties of the 'exact' RDF preconditioner are retained even when the inner solves are performed with low accuracy. In our 3D tests we have used a few (2-3) iterations of GMRES with an AMG preconditioner for the inner solves. It appears that while the iterative phase scales reasonably well, the set-up times for AMG are very high and do not lead to a scalable algorithm. This, of course, may be due to the particular AMG implementation used for the numerical tests. Future work should focus on finding more effective ways to efficiently apply the RDF preconditioner. A more detailed study of the spectral properties of RDF preconditioning is also warranted. In particular, the dependence of the eigenvalues of the preconditioned matrices on problem parameters such as mesh size, time step (for unsteady problems) and viscosity should be elucidated. Finally, RDF preconditioning should be compared with the augmented Lagrangian-based methods proposed in [10, 11].

Acknowledgements. Michele Benzi's work was supported in part by a grant from the University Research Council of Emory University. Michael Ng's work was supported in part by HKRGC grants and HKBU FRGs.

References

[1] P. R. Amestoy, T. A. Davis and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] Z.-Z. Bai, G. H. Golub and M. K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.

[3] Z.-Z. Bai and M. K. Ng, *On inexact preconditioners for nonsymmetric matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1710–1724.

[4] Z.-Z. Bai, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comp., 75 (2006), pp. 791–815.

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

[6] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20-41.

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

[8] M. Benzi and X.-P. Guo, *A dimensional split preconditioner for Stokes and linearized Navier–Stokes equations*, Math/CS Technical Report TR-2009-021, Emory University, Atlanta, GA. Revised, April 2010.

[9] M. Benzi and J. Liu, *An efficient solver for the Navier–Stokes equations in rotation form*, SIAM J. Sci. Comput., 9 (2007), pp. 1959–1981.

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

[11] M. Benzi, M. A. Olshanskii and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, Int. J. Numer. Meth. Fluids, (2010), DOI: 10.1002/fld.2267.

[12] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., 103 (2006), pp. 173–196.

[13] M. Benzi and A. J. Wathen, *Some preconditioning techniques for saddle point problems*, in W. Schilders, H. A. Van der Vorst and J. Rommes, eds., *Model Order Reduction: Theory, Research Aspects and Applications*, Springer-Verlag (Series: Mathematics in Industry), 2008, pp. 195–211.

[14] J. Boyle, M. D. Mihajlovic, J. A. Scott, *HSL MI20: An Efficient AMG preconditioner for finite element problems in 3D*, Int. J. Numer. Methods Engrg., 82 (2010), pp. 64–98.

[15] R. Bridson, *KKTDirect: A Direct Solver Package for Saddle-Point (KKT) Matrices*, Preprint, Department of Computer Science, University of British Columbia, 2009.

[16] Z.-H. Cao, *A note on constraint preconditioning for nonsymmetric indefinite matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 121–125.

[17] L. Chan, M. K. Ng and N. Tsing, *Spectral analysis of the HSS preconditioners*, Numer. Math. Theor. Meth. Appl., 1 (2008), pp. 113–137.

[18] A. C. de Niet and F. W. Wubs, *Numerically stable LDL^T-factorization of $\mathcal{F}$-type saddle point matrices*, IMA J. Numer. Anal., 29 (2009), pp. 208–234.

[19] H. S. Dollar and A. J. Wathen, *Approximate factorization constraint preconditioners for saddle-point matrics*, SIAM J. Sci. Comput., 27 (2006), pp. 1555–1572.

[20] H. C. Elman, V. E. Howle, J. Shadid, D. J. Silvester and R. Tuminaro, *Least squares preconditioners for stabilized discretizations of the Navier–Stokes equations*, SIAM J. Sci. Comput., 30 (2007), pp. 290–311.

[21] H. C. Elman, A. Ramage and D. J. Silvester, *IFISS: A Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Soft., 33 (2007), Article 14.

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

[23] 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), pp. 665–688.

[24] H. C. Elman and R. S. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electr. Trans. Numer. Anal., 35 (2009), pp. 257–280.

[25] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Application to the Solution of Boundary Value Problems*, Stud. Math. Appl., Vol 15, North-Holland, Amsterdam, 1983.

[26] N. I. M. Gould, M. E. Hribar and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1375–1394.

[27] S. P. Hamilton, M. Benzi and E. Haber, *New multigrid smoothers for the Oseen problem*, Numer. Linear Algebra Appl., 17 (2010), pp. 557–576.

[28] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Phys. Fluids, 8 (1965), pp. 2182–2189.

[29] I. C. F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[30] P. Jiranek and M. Rozloznik, *Adaptive version of Simpler GMRES*, Numer. Algorithms, 53 (2010), pp. 93–112.

[31] C. Keller, N. I. M. Gould and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[32] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.

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

[34] I. Perugia and V. Simoncini, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.

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

[36] Y. Saad, *Iterative Methods for Sparse Linear Systems. Second Edition*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[37] U. Trottenberg, C. Oosterlee and A. Schuller, *Multigrid*, Academic Press, San Diego, 2001.

[38] H. F. Walker, L. Zhou, *A simpler GMRES*, Numer. Linear Algebra Appl., 1 (1994), pp. 571–581.