INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **54**:355–377
Published online 19 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1401

Two preconditioners for saddle point problems in fluid flows

A. C. de Niet and F. W. Wubs*, †

Institute of Mathematics and Computing Science, University of Groningen, Blauwborgje 3, Groningen 9747 AC, The Netherlands

SUMMARY

In this paper two preconditioners for the saddle point problem are analysed: one based on the augmented Lagrangian approach and another involving artificial compressibility. Eigenvalue analysis shows that with these preconditioners small condition numbers can be achieved for the preconditioned saddle point matrix. The preconditioners are compared with commonly used preconditioners from literature for the Stokes and Oseen equation and an ocean flow problem. The numerical results confirm the analysis: the preconditioners are a good alternative to existing ones in fluid flow problems. Copyright © 2006 John Wiley & Sons, Ltd.

Received 15 February 2006; Revised 27 September 2006; Accepted 30 September 2006

KEY WORDS: saddle point problem; preconditioning; iterative methods; augmented Lagrangian; grad-div stabilization; artificial compressibility

1. INTRODUCTION

Commonly used finite-element and finite-difference discretizations of the equations describing an incompressible flow, like for example the Navier–Stokes equations, lead to an equation $Kx = b$, where K is of saddle point type:

$$K = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix} \tag{1}$$

*Correspondence to: F. W. Wubs, Institute of Mathematics and Computing Science, University of Groningen, PO Box 800, Groningen 9700 AV, The Netherlands.
†E-mail: f.w.wubs@math.rug.nl

Contract/grant sponsor: The Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs

In this matrix we have $A \in \mathbb{R}^{n\times n}$, a positive definite matrix, not necessarily symmetric, and $B \in \mathbb{R}^{n\times m}$, with $m \leqslant n$. Because of the zero diagonal block, K is indefinite, that is there are eigenvalues with positive real part as well as eigenvalues with negative real part.

In general, there are two ways to solve large linear systems like the equation $Kx = b$: *via* a direct or an iterative method. If the problems become large direct methods require too much memory, so one is forced to use iterative methods. For an overview of commonly used Krylov subspace methods see [1, Chapters 6 and 7]. To speed up the convergence it is profitable to use a preconditioner. In this paper we will focus on that for the case of saddle point matrices.

From literature a number of preconditioners for saddle point matrices is available. For a broad overview of the numerical solution of saddle point problems in all kind of applications see [2]. We will concentrate on incompressible flow problems. In Section 2, we recall the most important preconditioners for those kind of problems. In Section 3, we analyse two alternative preconditioners. In both sections we pay attention to the eigenvalues of the preconditioned saddle point matrix in order to get insight into the quality of the preconditioners and the convergence behaviour in a Krylov subspace method. The results of the eigenvalue analysis are supported by numerical experiments presented in Section 4. We compare the preconditioners on three saddle point problems: the Stokes equation, the Oseen equation and an equation from ocean circulation.

2. PRECONDITIONERS FROM LITERATURE

The general idea of constructing a preconditioner for the saddle point matrix is to exploit the structure of the matrix in such a way that (reduced) systems result to which existing methods can be applied.

Given a preconditioner $\hat{K}$, the size and distribution of the eigenvalues of the generalized eigenvalue problem $Kx = \lambda\hat{K}x$ are important for the convergence of the Krylov subspace method. For many of the methods presented in this paper the eigenvalues λ are related to the eigenvalues of the matrix $C = B^{\mathrm{T}}A^{-1}B$, which is minus the Schur-complement of A in the saddle point matrix K. In general the matrix C will be dense, so one would never want to construct this matrix in practice. However in several cases, for example the Stokes equation, good estimates for the eigenvalues of C are available. In the remainder of the paper, we will assume that C is diagonizable and that (μ_i, q_i) is an eigenpair of C, so $Cq_i = \mu_i q_i$; there are m of these pairs.

2.1. SIMPLE and SIMPLER

The first two preconditioners we want to mention are SIMPLE and SIMPLER. The original methods have been proposed by Patankar [3] and are stand-alone iterative methods, based on a special splitting. We follow the formulation as a preconditioner of Vuik and Saghir [4] and use it in a Krylov subspace method. Both methods involve the matrix D, that is the diagonal of A, and an approximation to the Schur-complement $\hat{C}_{\mathrm{SI}} = B^{\mathrm{T}}D^{-1}B$. Because D is diagonal, $\hat{C}_{\mathrm{SI}}$ will be sparse in contrast to C.

The SIMPLE preconditioner has the following form:

$$\hat{K}_{\mathrm{SI}} = \begin{pmatrix} A & AD^{-1}B \\ B^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} A & 0 \\ B^{\mathrm{T}} & I \end{pmatrix} \begin{pmatrix} I & D^{-1}B \\ 0 & -\hat{C}_{\mathrm{SI}} \end{pmatrix}$$

DOI: 10.1002/fld

The factorization shows that solving an equation with $\hat{K}_{\mathrm{SI}}$ can be reduced to solving equations with A and $\hat{C}_{\mathrm{SI}}$.

Theorem 1
The matrix $\hat{K}_{\mathrm{SI}}^{-1}K$ has an eigenvalue 1 with multiplicity n. The remaining eigenvalues are equal to the eigenvalues of the matrix $\hat{C}_{\mathrm{SI}}^{-1}C$.

Proof
See [5]. □

Note that, even in the case of a symmetric K, the SIMPLE preconditioner is not symmetric, therefore its transpose $\hat{K}_{\mathrm{SI}}^{\mathrm{T}}$ could serve as a preconditioner as well. This idea is used in the SIMPLER preconditioner, that is formed by a combination of $\hat{K}_{\mathrm{SI}}$ and its transpose. The resulting preconditioner is symmetric. We will denote the SIMPLER preconditioner by $\hat{K}_{\mathrm{SR}}$. The determination of the eigenvalues of $\hat{K}_{\mathrm{SR}}^{-1}K$ is complicated. We refer to [6] for details. The most important result of the analysis is that the eigenvalue 1 has multiplicity $2m$. The remaining $n-m$ eigenvalues are the eigenvalues of a matrix that is a function of A, D and B. In general, the SIMPLER preconditioner is more effective than SIMPLE and it is used quite often in industrial codes.

2.2. *The preconditioner of Wathen and Silvester*

Silvester and Wathen [7] have proposed the following block diagonal preconditioner:

$$\hat{K}_{\mathrm{WS}}(\omega)=\begin{pmatrix} A & 0 \\ 0 & I/\omega \end{pmatrix}$$

Note that a symmetric A gives a symmetric preconditioner. In the rest of the paper we will call it the WS-preconditioner.

For the eigenvalues of the WS-preconditioned matrix we have the following theorem.

Theorem 2
The matrix $\hat{K}_{\mathrm{WS}}^{-1}(\omega)K$ has an eigenvalue 1 with multiplicity $(n-m)$. The remaining $2m$ eigenvalues are equal to $(1\pm\sqrt{1+4\omega\mu_i})/2$.

Proof
The proof can be found in [7]. □

Usually $\omega=1$ in case of the Stokes equation and $\omega=\nu$ (the viscosity) in case of the Oseen equation. However, the method is not very sensitive to the precise value. If A is symmetric positive definite, C is also symmetric and positive definite, so the eigenvalues μ_i are real and we can assume that there is a largest and smallest eigenvalue: $0\leqslant\mu_{\min}\leqslant\mu_i\leqslant\mu_{\max}$. If $\omega<0$ all eigenvalues of the preconditioned matrix have positive real part and it becomes positive definite. For values $\omega<-1/(4\mu_{\max})$, the eigenvalues of the matrix $\hat{K}_{\mathrm{WS}}^{-1}(\omega)K$ become complex. For large negative values of ω, the imaginary component becomes large as well and this will slow down the convergence.

The theorem shows that for A symmetric positive definite and for positive values of ω the eigenvalues are real and clustered in three regions. The first cluster coincides with the point

DOI: 10.1002/fld

1, the second cluster contains the negative eigenvalues $(1-\sqrt{1+\omega\mu_i})/2$ and the third cluster contains the positive eigenvalues $(1+\sqrt{1+\omega\mu_i})/2$. The clusters are clearly separated, therefore, the convergence in a Krylov subspace method will be dominated by the two clusters away from 1.

2.3. The preconditioner of Elman and Silvester

Elman and Silvester [8] studied the following non-symmetric preconditioner (from now on the ES-preconditioner):

$$\hat{K}_{\mathrm{ES}}(\omega)=\begin{pmatrix} A & B \\ 0 & I/\omega \end{pmatrix}$$

Normally the preconditioner is used only if A is not symmetric. Nevertheless, it can be used as a preconditioner for symmetric K as well and it even appears to be better than the WS-preconditioner. The advantage of the ES-preconditioner is that the preconditioned matrix has more eigenvalues equal to 1 and the distribution of the other eigenvalues is better as well. This is shown in the following theorem.

Theorem 3
The matrix $\hat{K}_{\mathrm{ES}}^{-1}(\omega)K$ has an eigenvalue 1 with multiplicity n. The remaining eigenvalues are equal to $\omega\mu_i$.

Proof
The proof can be found in [8]. □

The theorem shows that at most two clusters can occur. One cluster is around the eigenvalues $\omega\mu_i$ and if 1 is not in this cluster, there is another cluster namely the point 1 itself. The 'cluster' of all eigenvalues 1 can be approximated very fast by a Krylov subspace method, therefore, the speed of convergence will be dominated by the scattering of the eigenvalues $\omega\mu_i$. In case of real symmetric A the eigenvalues are real and lie in the interval $[\mu_{\max}, \mu_{\min}]$. Convergence then is determined by the condition number $\mu_{\max}/\mu_{\min}$, which is equal to $\kappa(C)$. The parameter ω does not occur in the condition number, so its value appears to be arbitrary. We observed this ω-independency in numerical experiments that are not included in this paper.

2.4. The preconditioner of Kay, Loghin and Wathen

The fourth preconditioner we describe here is different from the previous ones because it has been developed by Kay, Loghin and Wathen for a special class of saddle point problems, namely the Navier–Stokes equations. In the rest of the paper we will refer to it as the KLW-preconditioner. We treat the preconditioner here because it is quite popular and as far as we know the best block-triangular preconditioner based on an approximation of the pressure Schur complement.

The incompressible Navier–Stokes equations on an open bounded domain are

$$\begin{aligned} -\nu\Delta u + (u\cdot\nabla)u + \nabla p &= 0 \\ \nabla\cdot u &= 0 \end{aligned} \tag{2}$$

DOI: 10.1002/fld

The linearized version of this equation is

$$\begin{aligned} -\nu\Delta u + (w\cdot\nabla)u + \nabla p &= 0 \\ \nabla\cdot u &= 0 \end{aligned} \tag{3}$$

where the 'wind' w is such that $\nabla\cdot w=0$. These *Oseen equations* occur if we solve the Navier–Stokes equations *via* a Picard iteration, where we take $u=u^{(m)}$ and $w=u^{(m-1)}$, the previous guess for the solution.

After discretization of (3) we get a saddle point problem of the form (1), where A is of the form $\nu H+N$. Here H is the discrete Laplacian (Δ), a symmetric and positive definite matrix. The matrix N contains the convective part $(w\cdot\nabla)$ of the equations and is non-symmetric.

In [9] the following preconditioner is proposed:

$$\hat{K}_{\mathrm{KLW}}=\begin{pmatrix} A & B \\ 0 & \hat{C}_{\mathrm{KLW}} \end{pmatrix} \tag{4}$$

with special choice for $\hat{C}_{\mathrm{KLW}}$

$$\hat{C}_{\mathrm{KLW}}=B^{\mathrm{T}}BA_p^{-1}M_p \tag{5}$$

where $A_p=\nu H_p+N_p$. Here the matrices H_p and N_p represent again the diffusive and convective part, respectively, however, they are now defined on the pressure space. The matrix M_p is the pressure–mass matrix. In our applications we use finite-difference or finite-volume methods and we scale the equations *a priori*, therefore, we usually have $M_p=I$.

Note that $\hat{C}_{\mathrm{KLW}}=C$ would imply that all eigenvalues of the preconditioned matrix $\hat{K}_{\mathrm{KLW}}^{-1}K$ are 1. In that case m of these eigenvalues are defective. Defective eigenvalues with large geometric multiplicity can slow down the convergence in the Krylov subspace method. Fortunately, one can easily show that the geometric multiplicity of all eigenvalues is at most 2.

In general the KLW-preconditioner will perform well if $\hat{C}_{\mathrm{KLW}}$ as defined in Equation (5) is a good approximation to the true Schur complement. There are several ways to derive the given expression. In [9] it is done *via* Green's functions. We will use an argument that is given in [10].

The first step is the search for a matrix A_p such that

$$AB\approx BA_p \tag{6}$$

We want this approximation to be as good as possible. This appears to be the case if the problem giving A_p is similar to that giving A, but then defined on the pressure spaces. Left multiplication with $B^{\mathrm{T}}A^{-1}$ gives $B^{\mathrm{T}}B\approx CA_p$. A rearrangement of the equation results in the following approximation to the Schur complement:

$$B^{\mathrm{T}}BA_p^{-1}\approx C \tag{7}$$

Note that $B^{\mathrm{T}}B$ is the discrete form of $\nabla\cdot\nabla=\Delta$ on the pressure space, therefore $B^{\mathrm{T}}B=H_p$. In case of the Stokes equation (where the wind field $w=0$, and so $N_p=0$) the preconditioner should coincide with the ES-preconditioner (as described in Section 2.3). This is ensured by an additional scaling of the approximation with the pressure–mass matrix, which gives us (5).

DOI: 10.1002/fld

Application of the KLW-preconditioner requires the solution of an equation with $\hat{C}_{\mathrm{KLW}}$. Solving that equation requires the application of A_p to a vector and the solution of an equation with $B^{\mathrm{T}}B$, which is quite easy because it represents a discretization of a Laplacian.

The preconditioner introduces the matrix A_p, which is related to an artificial convection–diffusion equation in the pressure space. We have to determine appropriate boundary conditions for this problem and its matrix. The derivation and motivation of the preconditioner do not give a clear indication what conditions we should impose. Kay *et al.* [9] and Elman *et al.* [10] advised to choose standard Neumann boundary conditions in case of velocity boundary conditions. The main argument is that the pressure field is determined up to a constant and that Neumann boundary conditions keep this property for A_p.

For the eigenvalues of the KLW-preconditioned matrix we have the following result.

Theorem 4
The matrix $\hat{K}_{\mathrm{KLW}}^{-1}K$ has an eigenvalue 1 with multiplicity n. The remaining eigenvalues are equal to the eigenvalues of the matrix $\hat{C}_{\mathrm{KLW}}^{-1}C$.

Proof
The eigenvalues of $\hat{K}_{\mathrm{KLW}}^{-1}K$ are equal to the ones of $K\hat{K}_{\mathrm{KLW}}^{-1}$. The last matrix is equal to

$$\begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix}\begin{pmatrix} A^{-1} & -A^{-1}B\hat{C}_{\mathrm{KLW}}^{-1} \\ 0 & \hat{C}^{-1} \end{pmatrix}=\begin{pmatrix} I & 0 \\ B^{\mathrm{T}}A^{-1} & C\hat{C}_{\mathrm{KLW}}^{-1} \end{pmatrix}$$

From this matrix we get the eigenvalues 1 (n times) and the eigenvalues of $C\hat{C}^{-1}$ which are the same as the ones of $\hat{C}_{\mathrm{KLW}}^{-1}C$. □

Now we have the same problem as with the eigenvalues of the SIMPLE preconditioned matrix (note the similarity between this theorem and Theorem 1). The eigenvalues of $\hat{C}_{\mathrm{KLW}}^{-1}C$ are hard to determine. There are merely numerical results that show that most eigenvalues in case of the Oseen equations are clustered around 1. Only a few eigenvalues can be found outside a small circle around 1. Theoretical results in the form of bounds on the eigenvalues are only obtained for the more 'symmetric' approximation to the Schur complement $M_p^{1/2}(B^{\mathrm{T}}B)^{1/2}A_p^{-1}(B^{\mathrm{T}}B)^{1/2}M_p^{1/2}$, that is never used in practice. See [10] for these results.

3. TWO ALTERNATIVE PRECONDITIONERS

We want to discuss two alternative preconditioners for saddle point equations. Both are symmetric if K is symmetric.

3.1. A preconditioner based on the augmented Lagrangian

The first preconditioner can be viewed as a variant of the WS-preconditioner that we described in Section 2.2. The difference: we add a term ωBB^{T} to the matrix A. This results in what we will call the grad-div preconditioner

$$\hat{K}_{\mathrm{GD}}(\omega)=\begin{pmatrix} A+\omega BB^{\mathrm{T}} & 0 \\ 0 & I/\omega \end{pmatrix}$$

DOI: 10.1002/fld

If A is symmetric positive definite and $\omega>0$, we have a symmetric and positive definite preconditioner $\hat{K}_{\mathrm{GD}}$ for the symmetric and indefinite matrix K.

Let us define the following transformation matrix:

$$T(\omega)=\begin{pmatrix} I & \omega B \\ 0 & I \end{pmatrix} \tag{8}$$

Application of the transformation with $\omega>0$ to a saddle point equation $Kx=b$, where K has the structure (1), gives

$$T(\omega)Kx=T(\omega)b \tag{9}$$

where

$$T(\omega)K=\begin{pmatrix} A+\omega BB^{\mathrm{T}} & B \\ B^{\mathrm{T}} & 0 \end{pmatrix} \tag{10}$$

Note that $T(\omega)K$ inherits symmetry from K.

Equation (9) is well known as the augmented Lagrangian formulation of the saddle point problem [11, 12]. In case of incompressible flow problems it is also called grad-div stabilization, because B and B^{T} are the discrete versions of the gradient and the divergence operator, respectively, and the addition of the term ωBB^{T} allows a more accurate solution of the velocity field for small viscosity [13, 14].

The most important difference between the GD-preconditioner and the augmented Lagrangian approach is that we use the augmented preconditioner for the non-augmented matrix (1). The preconditioner is in fact the WS-preconditioner for $T(\omega)K$ used as preconditioner for K itself. This approach seems a little strange but the eigenvalue analysis will show the advantages.

We have the following theorem about the eigenvalues of the grad-div preconditioned matrix.

Theorem 5

The matrix $\hat{K}_{\mathrm{GD}}^{-1}(\omega)K$ has an eigenvalue 1 with multiplicity n. The remaining eigenvalues are equal to

$$-\frac{\omega\mu_i}{1+\omega\mu_i}$$

Proof

We solve the generalized eigenvalue problem $Kx=\lambda\hat{K}_{\mathrm{GD}}(\omega)x$. First, we split x in a natural way in two variables: $x^{\mathrm{T}}=(u^{\mathrm{T}}, p^{\mathrm{T}})$. Expanding the eigenvalue problem gives

$$(1-\lambda)Au-\lambda\omega BB^{\mathrm{T}}u+Bp=0 \tag{11}$$

$$B^{\mathrm{T}}u-\lambda p/\omega=0 \tag{12}$$

Suppose $\lambda=1$, then the term $(1-\lambda)Au$ drops out of the first equation. The remainder can be written as $-\omega B(B^{\mathrm{T}}u-p/\omega)=0$. Between the brackets we recognize the second equation, so if that equation is satisfied, the first becomes trivial. Because the first equation reduces to a trivial one, we can choose u totally free. For any u we have an eigenvector $(u^{\mathrm{T}}, \omega u^{\mathrm{T}}B)^{\mathrm{T}}$ with eigenvalue 1.

DOI: 10.1002/fld

Since $u \in \mathbb{R}^n$, there are at most n independent vectors u. Therefore, the multiplicity of the eigenvalue 1 is n.

Now suppose that $\lambda \neq 1$. We rewrite Equation (12) with $B^{\mathrm{T}}u = \lambda p/\omega$ and substitute $B^{\mathrm{T}}u$ in Equation (11). We end up with

$$(1-\lambda)Au + (1-\lambda^2)Bp = 0$$

We have chosen $\lambda \neq 1$ and A is non-singular, so we can invert $(1-\lambda)A$ and transform this equation into $u = -(1+\lambda)A^{-1}Bp$. This u can be substituted in Equation (12), which gives

$$-[(1+\lambda)C + \lambda/\omega I]p = 0$$

Now we replace p by q_i, one of the eigenvectors of C. Because $Cq_i = \mu_i q_i$, the equation will be satisfied if $[(1+\lambda)\mu_i + \lambda/\omega] = 0$. Solving λ from this expression gives

$$\lambda = -\frac{\omega\mu_i}{1+\omega\mu_i}$$

which proves the theorem. □

For positive ω and μ_i real, the eigenvalues are contained in $(-1, 0) \cup \{1\}$. Note that we cannot choose ω equal or close to $-1/\mu_i$, because then $1 + \omega\mu_i$ will become very small and we get a very big eigenvalue. We can choose a negative value $\omega < -1/\mu_{\min}$. However, we need an estimate of the value of $\mu_{\min}$ to choose an appropriate ω. In general we do not have that, therefore, we choose positive ω.

It is remarkable that for $\omega \to \pm\infty$ we have only two eigenvalues namely ± 1. Therefore, for large values of $|\omega|$ the Krylov subspace method should converge in few iterations. However, in practice it is not possible to use large ω, because then the grad-div added system will become almost singular.

We add one more remark on the eigenvalues. If we compare the grad-div preconditioner with the preconditioner of Wathen and Silvester we see that by simply adding ωBB^{T} to A we half the number of eigenvalues that is not equal to one and the remaining eigenvalues are scaled and shifted such that they are much closer to -1.

3.2. A preconditioner based on artificial compressibility

The second preconditioner that we analyse in this section is constructed by adding artificial compressibility to the matrix K. This means that the zero diagonal block is replaced by the identity matrix multiplied with the parameter $-1/\omega$

$$\hat{K}_{\mathrm{AC}}(\omega) = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & -I/\omega \end{pmatrix} \tag{13}$$

The preconditioner can be viewed as a variant of the ES-preconditioner (see Section 2.3). We obtain $\hat{K}_{\mathrm{AC}}(\omega)$ by replacing the zero block in $\hat{K}_{\mathrm{ES}}$ by B^{T}.

This approach is not completely new. Almost three decades ago, Axelsson [15] proposed it as preconditioning by regularization. Maybe due to the lack of support by numerical results in that paper it has not been used much. In [16] we studied the spectral properties of the preconditioner.

DOI: 10.1002/fld

Recently, a similar approach was used in [17, 18] where it is called a primal-based penalty preconditioner. In the last paper, a preconditioner is proposed for more general saddle point problems where the lower (2, 2) block in (1) can be any negative semidefinite matrix. The AC-preconditioner we describe here can be seen as a special case of the primal-based penalty preconditioner. The preconditioner in [17] is applied to incompressible elasticity problems. We will apply our preconditioner to incompressible flow equations as is done in [18].

In analogy to the case of the grad-div preconditioner, the eigenvalues of the matrix K, preconditioned with $\hat{K}_{\mathrm{AC}}(\omega)$, are related to the eigenvalues of the Schur complement. This is stated in the following theorem.

Theorem 6
The matrix $\hat{K}_{\mathrm{AC}}^{-1}(\omega)K$ has an eigenvalue 1 with multiplicity n. The remaining eigenvalues are equal to

$$\frac{\omega\mu_i}{1+\omega\mu_i}$$

Proof
First, we expand the generalized eigenvalue problem $Kx = \lambda\hat{K}_{\mathrm{AC}}(\omega)x$ with $x^{\mathrm{T}} = (u^{\mathrm{T}}, p^{\mathrm{T}})$.

$$(1-\lambda)Au + (1-\lambda)Bp = 0$$

$$(1-\lambda)B^{\mathrm{T}}u + \frac{\lambda}{\omega}p = 0$$

If $\lambda = 1$, these equations reduce to $p/\omega = 0$. So p has to be zero and u is totally free. Any vector $(u^{\mathrm{T}}, 0)^{\mathrm{T}}$ is an eigenvector with eigenvalue 1. Since $u \in \mathbb{R}^n$, the multiplicity of eigenvalue 1 is n.

Now consider the case where $\lambda \neq 1$. We premultiply the first equation with $B^{\mathrm{T}}A^{-1}$ in order to get $(1-\lambda)B^{\mathrm{T}}u + (1-\lambda)Cp = 0$. If we subtract this equation from the second one we get

$$\frac{\lambda}{\omega}p - (1-\lambda)Cp = 0$$

In this equation we replace p by q_i, one of the eigenvectors of C. Now we use $Cq_i = \mu_i q_i$ and we have $(\lambda/\omega - \mu_i + \lambda\mu_i)q_i = 0$. The expression between brackets becomes zero if

$$\lambda = \frac{\omega\mu_i}{1+\omega\mu_i}$$

which proves the theorem. □

For μ_i real and positive values of ω, the eigenvalues are contained in the semi-open interval (0, 1]. For negative values of ω we have the same conditions as for $\hat{K}_{\mathrm{WS}}$: ω should not be equal to $-1/\mu_i$ and if $\omega < -1/\mu_{\min}$ the eigenvalues are in $[1, \infty)$.

If we send ω to $\pm\infty$ all eigenvalues of the preconditioned matrix converge to 1. Here, this is less surprising than in case of the grad-div preconditioner, because the matrix $\hat{K}_{\mathrm{AC}}(\omega)$ converges to K for large ω.

The eigenvalues of the matrices $\hat{K}_{\mathrm{GD}}^{-1}(\omega)K$ and $\hat{K}_{\mathrm{AC}}^{-1}(\omega)K$ are the same, except for a minus sign in front of the eigenvalues that are not one. This suggests some relation between the two

DOI: 10.1002/fld

preconditioners. Indeed there is one and it can be easily understood from the following factorization:

$$\begin{pmatrix} A & B \\ B^{\mathrm{T}} & -I/\omega \end{pmatrix} = \begin{pmatrix} I & -\omega B \\ 0 & I \end{pmatrix} \begin{pmatrix} A+\omega BB^{\mathrm{T}} & 0 \\ 0 & -I/\omega \end{pmatrix} \begin{pmatrix} I & 0 \\ -\omega B^{\mathrm{T}} & I \end{pmatrix} \tag{14}$$

This shows that $\hat{K}_{\mathrm{AC}}(\omega)$ is a composition of transformations $T(-\omega)$, $T(-\omega)^{\mathrm{T}}$, defined in Equation (8), and a block diagonal matrix that is almost equal to $\hat{K}_{\mathrm{GD}}$ except for a minus sign in the second diagonal block.

The factorization is useful not only for theoretical purposes. It provides a way to compute the action of $\hat{K}_{\mathrm{AC}}^{-1}(\omega)$. The inverses of the transformation matrices are easy ($T(-\omega)^{-1} = T(\omega)$), so the remaining problem is to solve an equation with $A + \omega BB^{\mathrm{T}}$ in the block diagonal matrix.

We add another remark on the eigenvalues of the preconditioned matrix. Benzi and Olshanskii [12] used an ES-preconditioner (with parameter $-(\omega+\nu)$) for the augmented system $T(\omega)K$. The generalized eigenvalues then become $(\omega+\nu)\mu_i/(1+\omega\mu_i)$. Note that these are almost the same as the ones we obtained in Theorem 6 for the AC-preconditioner. We can explain the similarity by transformations with $T(\omega)$. If we ignore the parameter ν (which is very small compared with ω in most cases), the ES-preconditioner in [12] is

$$\begin{pmatrix} A+\omega BB^{\mathrm{T}} & B \\ 0 & -I/\omega \end{pmatrix} = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & -I/\omega \end{pmatrix} \begin{pmatrix} I & 0 \\ \omega B^{\mathrm{T}} & I \end{pmatrix} = \hat{K}_{\mathrm{AC}}(\omega) T(\omega)^{\mathrm{T}} \tag{15}$$

This preconditioner is used for the matrix $T(\omega)K$ which is equal to $KT(\omega)^{\mathrm{T}}$. Note that both the preconditioner and the matrix have the same factor. In the generalized eigenvalue problem it would drop out and we get the very same eigenvalue problem as we had for the AC-preconditioner. If we forget the ν and consider right preconditioning, the preconditioner in [12] applied to the augmented system is equivalent to an AC-preconditioner applied to the original saddle point problem. This explains why the results we obtain in Section 4.2 for the Oseen equations are similar to the results in [12].

Instead of (15) we can consider the ES-preconditioner with $+\omega$ as parameter as a preconditioner for the augmented system. We have a similar factorization:

$$\begin{pmatrix} A+\omega BB^{\mathrm{T}} & B \\ 0 & I/\omega \end{pmatrix} = \begin{pmatrix} I & \omega B \\ 0 & I \end{pmatrix} \begin{pmatrix} A+\omega BB^{\mathrm{T}} & 0 \\ 0 & I/\omega \end{pmatrix} = T(\omega) \hat{K}_{\mathrm{GD}}(\omega)$$

Now this preconditioner and the augmented system have a factor $T(\omega)$ in common. Using this preconditioner as a left-preconditioner for the augmented system is equivalent to solving the non-augmented system with a GD-preconditioner.

3.3. Remarks on the condition number

If one of the preconditioners $\hat{K}_*$ in this paper is used in a Krylov subspace method, the convergence will depend on the distribution of the eigenvalues of the preconditioned matrix over the complex plane. In case of symmetric positive definite A, the distribution can be measured by the spectral condition number κ, which is the quotient of the largest and smallest eigenvalue in absolute value.

In case of the preconditioners in this paper the condition number appears not to be a good estimate. This is caused by the large number of eigenvalues 1. In itself it is already advantageous

DOI: 10.1002/fld

when eigenvalues coincide, since the number of iterations of a Krylov subspace method is at most equal to the number of different eigenvalues (assuming they are simple). However, if these coinciding eigenvalues are also well separated from the others they become extreme eigenvalues, of which the corresponding subspace is found first by the subspace method, and after that, these eigenvalues do not play a role anymore in the speed of convergence, see [19, Sections 5.3, 6.2]. Therefore, we will use an effective spectral condition number $\tilde{\kappa}$, instead of the real spectral condition number κ. The number $\tilde{\kappa}$ is the quotient of the largest and smallest eigenvalue in the absolute value *not equal to one*.

For the ES-preconditioner we have

$$\tilde{\kappa}(\hat{K}_{\mathrm{ES}}^{-1}(\omega)K) = \frac{\omega\mu_{\max}}{\omega\mu_{\min}} = \frac{\mu_{\max}}{\mu_{\min}} = \kappa(C) \tag{16}$$

The non-unit eigenvalues of the preconditioned matrices $\hat{K}_{\mathrm{AC}}^{-1}K$ and $\hat{K}_{\mathrm{GD}}^{-1}K$ are the same except for a minus sign. Hence, their condition numbers are equal. Since, the function $x/(1+x)$ is monotonously increasing, the largest eigenvalue of the preconditioned matrix corresponds to the largest eigenvalue $\mu_{\max}$ and the smallest to the smallest eigenvalue $\mu_{\min}$. The effective condition number of both methods is

$$\tilde{\kappa}(\hat{K}_{\mathrm{AC}}^{-1}K) = \tilde{\kappa}(\hat{K}_{\mathrm{GD}}^{-1}K) = \frac{\omega\mu_{\max}}{1+\omega\mu_{\max}} \cdot \frac{1+\omega\mu_{\min}}{\omega\mu_{\min}} = \frac{\mu_{\max}}{\mu_{\min}} \cdot \frac{1+\omega\mu_{\min}}{1+\omega\mu_{\max}} \leqslant \kappa(C) \tag{17}$$

The eigenvalue analysis already showed that the preconditioners are good for large ω. This analysis shows that for any positive value of ω, the effective condition number of the preconditioners is better than that of the ES-preconditioner at least in the case of symmetric and positive definite A.

3.4. Solving grad-div added systems

An important issue for the AC- and GD-preconditioner is the solution of grad-div added systems. The alternative preconditioners have better spectral properties than the ES- and WS-preconditioner, but they require the solution of systems involving $A + \omega BB^{\mathrm{T}}$. This system is more difficult to solve than A itself. The gain in the outer iterations could get lost in the solution of systems in the inner iterations.

The solution of systems with A is not an issue at all. In the applications in this paper the matrix A is of convection–diffusion type with possibly a Coriolis force. For these type of systems a lot of (algebraic) multigrid methods and incomplete LU factorizations are available that provide good preconditioners [20, 21]. With the addition of ωBB^{T} to A it becomes questionable whether these methods still can be applied. Therefore, in this section we will discuss ways to solve the grad-div added systems.

In general, the addition of ωBB^{T} to A will create new entries in the matrix. The increase of non-zeros depends on the underlying equations and their discretization. For the problems we discuss in the next section the number of non-zeros is almost doubled by grad-div adding.

A second remark: if the matrix A is an M-matrix, which means it is symmetric positive definite and diagonally dominant, the matrix $A + \omega BB^{\mathrm{T}}$ will still be symmetric and positive definite (as long as $\omega > 0$), but it will loose diagonal dominance; there will be positive off-diagonal entries. For M-matrices a lot of methods are available. In general they cannot be applied without any concern to the matrix $A + \omega BB^{\mathrm{T}}$. However, in case of the test problems we describe in the next section, we will see that for moderate values of ω some of these methods can be applied.

DOI: 10.1002/fld

Furthermore, we want to point out that the matrices A and $A+\omega BB^{\mathrm{T}}$ are spectrally equivalent. The matrix $A^{-1}(A+\omega BB^{\mathrm{T}})$ has $n-m$ eigenvalues 1 and the other m eigenvalues are $1+\omega\mu_i$. The effective condition number is

$$\tilde{\kappa}(A^{-1}(A+\omega BB^{\mathrm{T}}))=\frac{1+\omega\mu_{\max}}{1+\omega\mu_{\min}}\leqslant\kappa(C)$$

If this last number is small, the matrix A can be used as preconditioner for $A+\omega BB^{\mathrm{T}}$. Instead of the matrix A itself we can also use a preconditioner for A as a preconditioner for $A+\omega BB^{\mathrm{T}}$.

Finally, we have to remark that the theory about the solution of grad-div added systems is not well developed yet. Several papers struggle with the question what is the best way to solve this kind of systems. Olshanskii and Reusken [14] used a standard multigrid method, Dohrmann and Lehoucq [17] and Gartling and Dohrmann [18] used a balancing domain decomposition by constraints (BDDC) preconditioner and Benzi and Olshanskii [12] used a multigrid method with a special type of Gauss–Seidel smoother. The last multigrid method seems the most promising for our applications. At least for the Oseen equations the convergence is independent of the mesh size. For all preconditioners it holds that the results deteriorate with increasing ω.

Summarizing we can state that multigrid methods exist for the solution of grad-div added systems that work fine for modest values of ω. Unfortunately we do not have these codes, so in the next section we will use a robust standard incomplete LU-factorization for the grad-div added systems. Maybe an incomplete LU-factorization is not the best method, but for modest values it suffices to illustrate the possibilities of the preconditioners.

The research for a better and more general approach for grad-div added systems especially for larger values of ω is ongoing.

4. NUMERICAL RESULTS

In this section, we show a comparison between the preconditioners for two different saddle point problems. All computations are performed in MATLAB (7.1.0.183 (R14) Service Pack 3) on a PC with two 2.4 GHz AMD Opteron processors and 7.6 GB memory.

4.1. Stokes flow in a driven cavity

The first problem is the two-dimensional Stokes equation in a driven cavity. The following set of equations has to be solved on the unit square Ω:

$$\begin{aligned}-\nu\Delta u+\nabla p&=0\\ \nabla\cdot u&=0\end{aligned}\tag{18}$$

where $u(x,y)$ is the velocity field and $p(x,y)$ the pressure field; the parameter ν controls the amount of viscosity. The non-trivial solution is determined by the boundary conditions that are zero on three sides of the unit square. At the upper boundary ($y=1$) we have a horizontal velocity $u(x,1)=1$.

We can get rid of the parameter ν by defining a new pressure variable $\bar{p}=p/\nu$. If the first equation is divided by ν, we can substitute p by $\bar{p}$ and the parameter ν is gone. So we may assume that $\nu=1$.

DOI: 10.1002/fld

The equations are discretized on a uniform staggered grid (a Marker-and-Cell or C-grid) with mesh size h using standard second-order finite differences, which results in a system of linear equations $Kx = b$, where K is of saddle point form. It is well known that in this case the Schur complement $C = BA^{-1}B^{\mathrm{T}} \sim I$, hence, the condition number $\kappa(C)$ is independent of the mesh size.

In Table I we show the number of iterations in BICGSTAB [22] needed to obtain an accuracy of 10^{-6} for the preconditioned Stokes matrix. We applied the preconditioners mentioned in this paper and varied the parameter ω in case of the artificial compressibility and grad-div preconditioner. Moreover, we used several mesh sizes. Equations with A or $A + \omega BB^{\mathrm{T}}$ are solved *via* an exact LU factorization.

The results are as we expected from the eigenvalue analysis. The condition number $\kappa(C)$ is independent of the mesh size and so is the number of iterations using the ES-, GD- and AC-preconditioners. For the SIMPLER and WS-preconditioner the number of iterations slightly increases when the mesh size decreases, so there is some grid dependence. The SIMPLE preconditioner cannot be considered as a serious alternative, because the number of iterations grows rapidly with the grid size.

As one can see, even for ω equal to one, the number of iterations for the GD- and AC-preconditioner is smaller than for the ES-preconditioner. According to (17) the effective condition numbers of the GD- and AC-preconditioned matrices are smaller than that of the ES-preconditioned matrix. The last number is equal to the condition number of C, that is independent of the grid size. Indeed, BICGSTAB converges almost two times faster for the GD- and AC-preconditioner. With increasing ω, the effective condition numbers of GD and AC get even closer to 1 and the number of iterations decreases further, as we expected from the analysis.

We have to remark that in Table I we only measured the number of iterations. The amount of work per iteration per preconditioner varies quite a lot. For example, one iteration with SIMPLER requires a lot more computations than one iteration with the AC-preconditioner, that in turn is more expensive than one iteration with the ES-preconditioner. Therefore, in Table II we attempt to make a fairer comparison between the methods. We compare the cpu-times for the construction of the preconditioner and the solution of the largest Stokes problem. We do not use exact solves for the subsystems A and $A+\omega BB^{\mathrm{T}}$, but instead we solve these system with GMRES using MATLAB's LUINC factorization as a preconditioner. The drop tolerance for the incomplete factorization is 3×10^{-4} in case of A and 10^{-4} in case of $A + \omega BB^{\mathrm{T}}$. The tolerances are chosen such that both factorizations have approximately the same amount of non-zeros per row. Because of the nested iterative scheme we have to use a flexible Krylov method in the outer iteration. In the experiments we used FGMRES [23]. There is no need to solve the inner iterations very accurately so there we stop as soon as an accuracy of 10^{-3} is reached. The accuracy for the outer iteration is 10^{-6}.

Table I. Number of iterations in BICGSTAB (accuracy 10^{-6}) with several preconditioners for the Stokes equation in a driven cavity.

Method	SI	SR	WS	ES	GD	GD	GD	AC	AC	AC
ω	—	—	1	1	1	16	256	1	16	256
$h = 1/32$	48	8	15	7	5	3	3	4	2	2
1/64	111	12	18	7	5	3	3	4	2	2
1/128	243	14	20	7	5	3	2	4	2	2
1/256	>300	22	23	7	5	3	2	4	2	2

Note: Subsystems are solved exactly.

DOI: 10.1002/fld

Table II. Number of iterations in FGMRES (accuracy 10^{-6}) and cpu-times required for the solution of the Stokes equation with mesh size $h = 1/256$.

Method	SR	WS	ES	GD	AC
Value of ω	—	1	1	16	16
Iterations in FGMRES	30	26	13	5	4
Construction time (s)	18	11	11	22	22
Solve time (s)	1822	456	214	171	113

Note: Subsystems are solved with GMRES up to an accuracy of 10^{-3} using as a preconditioner MATLAB's LUINC with drop tolerance 3×10^{-4} (WS/ES) or 10^{-4} (GD/AC).

The table shows that the grad-div and artificial compressibility preconditioners can compete in practice with the ES- and WS-preconditioners. The construction of the first preconditioners is more expensive because we have to compute an incomplete LU factorization for the grad-div added matrix that has more non-zeros than the original one. Note that even with the relative large value $\omega = 16$ we are able to compute a good incomplete factorization for the matrix $A + \omega B B^{\mathrm{T}}$. So the problems sketched in Section 3.4 appear to be not too serious in case of the Stokes equation. The loss in time in the factorization phase of the GD- and AC-preconditioner is compensated by a much faster convergence of the outer iteration in the solution phase.

Finally, we remark that the table illustrates that SIMPLER is much more expensive to apply than the other methods. It requires approximately the same amount of iterations as the WS-preconditioner, but it is roughly four times more expensive per iteration.

4.2. Oseen equations in a driven cavity

The second test is a benchmark problem which is proposed by Elman *et al.* [10] and used by Benzi and Olshanskii [12]. It is again a driven cavity problem, but now for the Oseen equations that we described in Equation (3). The area Ω is the unit square. The boundary conditions are

$$\begin{aligned} &u_1 = u_2 = 0 \quad \text{for } x = 0, \quad x = 1 \text{ and } y = 0 \\ &u_1 = 1, \quad u_2 = 0 \quad \text{for } y = 1 \end{aligned} \tag{19}$$

the wind is

$$w = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix} \tag{20}$$

This wind contains a single recirculation on the domain Ω. For a more detailed description of the problem see [10]. We solve the problem on a regular grid with a Marker-and-Cell or C-grid discretization using central differences.

We do not show the results of the SI-, SR-, WS- and ES-preconditioner because they perform quite bad on the Oseen equations. The best preconditioner in this case is the KLW-preconditioner so we will compare that one with the artificial compressibility preconditioner.

The results with the KLW-preconditioner for our discretization can be found in Table III. The number of iterations is independent of the mesh size, but slightly depends on the viscosity. This agrees with the results found in [10, 12].

DOI: 10.1002/fld

Table III. Number of iterations in GMRES (accuracy 10^{-6}) for the KLW-preconditioner for the Oseen equations in a driven cavity.

	Viscosity v				
h	1/20	1/40	1/80	1/160	1/320
1/16	17	19	21	24	26
1/32	17	19	21	22	24
1/64	18	20	21	23	25
1/128	19	20	22	23	25
1/256	18	19	22	23	25

Note: Subsystems are solved exactly.

Table IV. Number of iterations in GMRES (accuracy 10^{-6}) for the artificial compressibility preconditioner with $\omega = 1$ for the Oseen equations in a driven cavity.

	Viscosity v				
h	1/20	1/40	1/80	1/160	1/320
1/16	6	6	6	6	6
1/32	6	6	6	6	6
1/64	5	5	5	6	6
1/128	5	5	5	5	5
1/256	4	4	4	5	5

Note: Subsystems are solved exactly.

Table V. Number of iterations in GMRES (accuracy 10^{-6}) for the artificial compressibility preconditioner with several values of ω for the Oseen equations in a driven cavity on the largest grid $h = \frac{1}{256}$.

	Viscosity v				
ω	1/20	1/40	1/80	1/160	1/320
4	3	3	3	3	3
1	4	4	4	5	5
1/4	8	8	9	9	9
1/16	14	17	21	23	25

Note: Subsystems are solved exactly.

In Table IV one finds the results of preconditioning with the artificial compressibility preconditioner for the Oseen equations. For the parameter ω we chose the value 1. Clearly, the number of iterations is independent of grid size and viscosity for this value of ω.

The results we obtain are comparable with the ones in [18] on a slightly different Oseen problem. In that paper the number of outer iteration seems to increase a little with the Reynolds number. We do not observe this in Table IV. This difference could be explained by the choice of the parameter ω.

We show the dependency on ω in Table V. The table contains the number of iterations for different values of ω and v. Only for the smallest value $\omega = \frac{1}{16}$ the number of iterations depends

DOI: 10.1002/fld

slightly on ν. For even smaller values of ω (for example $\frac{1}{64}$) the number of iterations increases more rapidly with decreasing viscosity and eventually the method will fail to converge. This is not a surprise, because for these values the preconditioner becomes more and more similar to the ES-preconditioner which is known to be not appropriate for the Oseen equation with small viscosity.

The number of iterations does not tell the whole story. In both Tables III and IV we solved the subsystems with A and $\hat{C}_{\mathrm{KLW}}$ or $A+\omega BB^{\mathrm{T}}$ exactly. In general, the last system will be more difficult to solve. Therefore, in Table VI we compare the cpu-times for the largest problem ($h=\frac{1}{256}$) and for the two extreme values of ν. When we have to apply the preconditioner we solve the subsystems with GMRES using the incomplete LU factorization of MATLAB as preconditioner, which is certainly not the best for these problems, but here it suffices. We chose the drop tolerance such that all incomplete factorizations have approximately the same amount of fill per row.

In Table VII we show the dependency of the cpu-times on the grid size. The grid is refined with a factor 2 in each direction. For the larger problems the solve time grows with a factor bigger than 4. Neither KLW nor AC combined with LUINC is ideal in the sense that the amount of work scales linearly with the problem size. In Table VII we see that the increase in time is caused mainly by the number of inner iterations. This is observed in [18] as well. Here, we can blame LUINC. We already mentioned that there exist better (multigrid) methods to solve A (see [9]) and $A+\omega BB^{\mathrm{T}}$ (see [12]), but, although the LUINC-factorization is not the best, the scaling results are reasonable.

Table VI. Number of iterations in FGMRES (accuracy 10^{-6}) and cpu-times required to solve the Oseen equations for $h=\frac{1}{256}$. In all cases $\omega=1$.

Method	KLW	KLW	AC	AC
Viscosity ν	1/20	1/320	1/20	1/320
Iterations in FGMRES	26	48	13	11
Construction time (s)	11	11	18	12
Solve time (s)	148	277	66	147

Note: Subsystems are solved with GMRES up to an accuracy of 10^{-3} using MATLAB's LUINC factorization with drop tolerance 3×10^{-4} (KLW) and 10^{-4} (AC).

Table VII. Number of outer iterations in FGMRES (accuracy 10^{-6}), average number of inner iterations in GMRES (accuracy 10^{-3}) and cpu-times required to solve the Oseen equations for $\nu=\frac{1}{80}$ for different grid sizes. In all cases $\omega=1$.

	KLW				AC			
Method h	Outer iter	Inner iter	Construction time	Solve time	Outer iter	Inner iter	Construction time	Solve time
16	23	1	0.02	0.19	6	1	0.02	0.08
32	23	2	0.05	0.74	7	2	0.09	0.46
64	23	3	0.25	5.27	7	3	0.50	2.11
128	24	4	1.67	33.4	9	5	2.46	10.6
256	28	6	11.4	186	10	9	13.7	75.5

Note: Subsystems are solved using MATLAB's LUINC factorization with drop tolerance 1×10^{-4} (KLW) and 3×10^{-4} (AC) as preconditioner.

DOI: 10.1002/fld

The timing results show that in case of the Oseen equation the artificial compressibility preconditioner is not only theoretical but also in practice a serious alternative to the KLW-preconditioner.

4.3. Thermohaline ocean circulation

One of the applications of the sketched preconditioners is the numerical simulation of three-dimensional thermohaline ocean circulation. The equations of thermohaline ocean circulation involve six variables, namely the flow velocity in longitudinal, latitudinal and vertical (depth) direction (u, v, w), pressure (p), temperature (T) and salinity (S). The word *thermohaline* means that the flow is driven by temperature and heat fluxes. We have six conservation laws: conservation of momentum in the three directions and conservation of mass, heat and salinity. For a detailed description of the equations and the precise parameter setting see [24].

The longitudinal and latitudinal momentum equations are similar, the same holds for the conservation of salinity and heat, therefore, a natural clustering of variables is (u, v) and (S, T). Because of the Coriolis force that is involved we have to discretize on a spherical Lorenz grid, which means that we have a B-grid in the horizontal (longitudinal and latitudinal) direction and a C-grid in the vertical (depth) direction. A motivation for this special grid is given in [25]. The discretized system has the following structure:

$$\begin{pmatrix} A_{uv} & 0 & G_{uv} & 0 \\ 0 & 0 & G_w & B_{\mathrm{ST}} \\ D_{uv} & D_w & 0 & 0 \\ B_{uv} & B_w & 0 & A_{\mathrm{ST}} \end{pmatrix} \begin{pmatrix} x_{uv} \\ x_w \\ x_p \\ x_{\mathrm{ST}} \end{pmatrix} = \begin{pmatrix} b_{uv} \\ b_w \\ b_p \\ b_{\mathrm{ST}} \end{pmatrix} \tag{21}$$

The four rows represent, respectively, conservation of momentum in horizontal direction, conservation of momentum in vertical direction (which simplifies to the hydrostatic pressure equation), conservation of mass and conservation of heat and salinity.

The submatrix A_{uv} consists of two convection–diffusion equations that are coupled by the Coriolis force. The coupling is skew-symmetric, which makes the problem more difficult. Another cause of troubles is the zero diagonal block in the vertical momentum equation. This block is empty because the equation is dominated by the balance between gravity and pressure.

We designed a preconditioner for the equation. It is based on depth integration of the pressure, a transformation of the continuity equation and an asymmetric rearrangement of rows and columns. We will not describe it here in detail. In this paper, it is enough to know that at some point we have to solve a depth-averaged saddle point problem. Let M_{uv} denote the depth averaging operator for the velocities and M_p the depth averaging operator for the pressure. We can then define the depth average matrices $A = M_{uv} A_{uv} M_{uv}^{\mathrm{T}}$, $B_1 = M_{uv} G_{uv} M_p^{\mathrm{T}}$ and $B_2 = M_p D_{uv} M_{uv}^{\mathrm{T}}$. Equation (21) contains a saddle point problem that is revealed if we remove the second and fourth columns and rows. Depth averaging of that saddle point problem results in a new saddle point problem and that is the one we will consider

$$\begin{pmatrix} A & B_1 \\ B_2 & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_{uv} \\ \bar{x}_p \end{pmatrix} = \begin{pmatrix} \bar{b}_{uv} \\ \bar{b}_p \end{pmatrix} \tag{22}$$

One of the properties of the resulting saddle point equation is that A is a convection–diffusion equation involving a Coriolis force. In the Appendix, we show *via* Fourier analysis that the

DOI: 10.1002/fld

Coriolis force does not necessarily destroy the property that the Schur complement $C = B_2 A^{-1} B_1$ is independent of the mesh, though the condition number may be substantially larger.

As a test problem we isolate a saddle point problem from a developed flow in a rectangular box in the North Atlantic (286–350° longitude, 10–74° latitude). With a developed flow we mean that wind, temperature, salinity and Coriolis force play an important role in the solution. In other words we deal with a real ocean flow problem. We extract saddle point problems for varying mesh sizes. The problem has a horizontal Ekman number $E_H = 4 \times 10^{-5}$ and a Reynolds number $Re = 2.5$. The low Reynolds number is because of the relative low resolution, which forces us to use a large horizontal viscosity parameter.

In Table VIII one finds the number of iterations for some of the preconditioners we treated in this paper. First of all we have to remark that we do not show the results for the ES- and WS-preconditioner. Both are not able to produce an acceptable preconditioner. The maximum number of iterations (300) in BICGSTAB is exceeded without getting near the desired accuracy. Furthermore, the KLW-preconditioner cannot be used. It is not defined for this type of problem. Because of the presence of a dominant skew-symmetric Coriolis force it is no longer clear what kind of A_p we should choose such that Equation (6) is a good approximation.

If we apply the SIMPLE and SIMPLER preconditioners to the saddle point problem from ocean circulation we get very bad results: the maximum number of iterations is exceeded before we reach the desired accuracy. However, we can improve the performance of the preconditioners. As we pointed out before, the Coriolis force is very important in the equations. It is essentially a coupling between u and v and after discretization it will create an off-diagonal entry. The SIMPLE(R) preconditioner uses the diagonal of A and, therefore, it does not see the importance of Coriolis force. We can tackle this problem by a simple modification. Assume that we order u and v such that the horizontal velocities that belong to the same grid point are clustered. Now if we take the 2×2-block-diagonal of A instead of the normal diagonal, we include the discretization of the Coriolis force. We will call the resulting preconditioners modified SIMPLE(R) or MSI and MSR for short.

The preconditioner proposed by Maxim and Olshanskii [13] for the rotation form of the Navier–Stokes equations could be applied to the ocean problem as well. It is a variant of the KLW-preconditioner that we described in Section 2.4. It uses a different approximation to the Schur complement that involves a 2×2-block-diagonal approximation to A similar to the one we use in the MSI- and MSR-preconditioner. The numerical results we obtain for the preconditioner in [13] are similar to that of MSI, what is not a surprise if one takes into account the similarity between

Table VIII. Average number of iterations in BICGSTAB (accuracy 10^{-6}) for the modified SIMPLE(R), grad-div and artificial compressibility preconditioner for a depth-averaged saddle point problem from ocean circulation.

Method	MSI	MSR	GD	GD	GD	AC	AC	AC
ω	—	—	16	64	256	16	64	256
$n = 768$	16	7	67	30	11	33	12	5
3072	33	23	96	26	13	34	13	5
6912	78	34	78	30	12	33	12	5
12 288	116	47	80	24	12	31	13	5
19 200	158	67	45	21	9	23	9	4

Note: Subsystems are solved exactly.

DOI: 10.1002/fld

Table IX. Number of iterations in FGMRES (accuracy 10^{-6}) and cpu-times required for the solution of the largest depth-averaged saddle point problem from ocean circulation ($n = 19\,200$).

Method	MSI	MSR	GD	AC
Value of ω	—	—	16	16
Iterations	155	83	254	40
Construction time	1.0	1.0	1.9	1.9
Solve time	41	36	152	24

Note: Subsystems are solved with GMRES (accuracy 10^{-3}) using MATLABS's LUINC with drop tolerance 3e−4 as preconditioner.

Theorems 1 and 4. Because these preconditioners are so closely related that we only show the numerical results for the MSI-preconditioner.

In Table VIII we compare the results for different preconditioners. The problem size in the table is the size of the depth-averaged saddle point problem of Equation (22). The reader should keep in mind that the underlying ocean problem as described in (21) is at least 32 times bigger. The solution of the full ocean equations requires a number of solves of the depth-averaged saddle point problem.

We have to choose a relative large value for ω to get acceptable results for the grad-div and artificial compressibility preconditioner. This is probably due to the Coriolis force and bad scaling of the original matrix. For the values shown in the table, the number of iterations seems to depend only slightly on the problem size. For the modified SIMPLE and SIMPLER preconditioners the dependence is a lot stronger.

Finally, we show in Table IX the cpu-times for the preconditioners on the largest saddle point problem if we solve the subsystems with GMRES using MATLAB's LUINC factorization as a preconditioner. It seems that the grad-div and artificial compressibility preconditioners are more sensitive to the replacement of exact solves by approximations than the modified SIMPLE and SIMPLER method. The number of iterations using the grad-div preconditioner is even five times bigger than in the case of exact solution of subsystems. This is possibly due to the asymmetry of the saddle point problem. The artificial compressibility preconditioner is the fastest in this case, although the modified SIMPLE and SIMPLER preconditioner perform quite well too.

5. SUMMARY AND DISCUSSION

In this paper, we treated two preconditioners for the saddle point problem: one based on the augmented Lagrangian and another on artificial compressibility. An eigenvalue analysis shows that small condition numbers are achievable. This analysis is supported by the numerical experiments for the Stokes equation, the Oseen equation and a saddle point problem from an ocean model. In all cases the artificial compressibility preconditioner performs better than the preconditioners SIMPLE(R), ES, WS and KLW.

One might consider the presence of the parameter ω in the preconditioners as a disadvantage. However, the advantage of this parameter is that we can shift work from inner to outer iterations and back. If we increase the value of ω the condition number of the preconditioned matrix will be closer to 1 and the number of outer iterations will decrease. Meanwhile the grad-div added matrix

DOI: 10.1002/fld

will be more difficult to solve so the number of inner iterations will increase. The parameter should be chosen such that there is a balance in work in the inner and outer iteration. In general, if the entries in A and B are scaled such that they are of the same magnitude, $\omega = \mathcal{O}(1)$ works fine. Our experience is that near this value the increase or decrease of ω with a factor of 2 will shift the amount of work, but the overall cpu-time will be almost the same. Hence, fortunately the choice is not very critical.

As we mentioned in Section 3.4 the question of a robust preconditioner for grad-div added matrices is not fully solved. We sketched several possibilities. However, the numerical experiments show that in practice we can apply the methods for A to $A + \omega BB^{\mathrm{T}}$ as well, at least for modest values of ω. Nevertheless, the solution of these systems needs more study especially for larger values of ω.

One of the advantages of the described preconditioners is that they perform well on the three different test problems without concern. The eigenvalue analysis shows that the preconditioners will always perform well on a saddle point problem that has the structure of (1) as long as we choose ω large enough. This certainly does not hold for the preconditioners from literature that we described in this paper. The KLW-preconditioner is only defined for the Oseen equations; the WS- and ES-preconditioners perform bad on Oseen and ocean equations; the SIMPLE(R) preconditioners do not perform well on the Oseen equations and we have to modify the methods before we can apply them to the ocean equations.

Both eigenvalue analysis and numerical experiments show that the grad-div and artificial compressibility preconditioners might be good alternatives for existing preconditioners in the solution of saddle point problems in fluid flows.

APPENDIX: FOURIER ANALYSIS FOR STOKES WITH CORIOLIS

In Section 4.3 we stated that the eigenvalues of the Schur complement $C = B^{\mathrm{T}}A^{-1}B$ in case of the saddle point problem from the ocean equations are independent of the mesh size. Here, we will motivate that *via* Fourier analysis on the Stokes equation with a Coriolis force, because in the ocean flow equations the diffusion terms dominate the convection terms.

We start with the continuous case. The Stokes equation with a Coriolis force yields a saddle point problem of the form (1) where A is a discrete version of

$$\begin{pmatrix} \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} & \Omega \\ -\Omega & \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} \end{pmatrix}$$

Here, Ω represents the ratio of the Coriolis force and the viscosity; we assume Ω to be constant. The matrix B is the discretization of

$$\begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}$$

DOI: 10.1002/fld

If we define

$$\begin{pmatrix} u \\ v \\ p \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \\ p_0 \end{pmatrix} \mathrm{e}^{\mathrm{i}f_1 x + \mathrm{i}f_2 y}$$

the symbols of the continuous form of A and B become

$$\hat{A} = \begin{pmatrix} f_1^2 + f_2^2 & \Omega \\ -\Omega & f_1^2 + f_2^2 \end{pmatrix} \quad \text{and} \quad \hat{B} = \begin{pmatrix} \mathrm{i}f_1 \\ \mathrm{i}f_2 \end{pmatrix}$$

So the inverse of $\hat{A}$ is

$$\hat{A}^{-1} = \frac{1}{(f_1^2 + f_2^2)^2 + \Omega^2} \begin{pmatrix} f_1^2 + f_2^2 & -\Omega \\ \Omega & f_1^2 + f_2^2 \end{pmatrix}$$

Hence $\hat{C} = \hat{B}^* \hat{A}^{-1} \hat{B}$ becomes (split $\hat{A}^{-1}$ in symmetric and skew symmetric part):

$$\hat{C} = \frac{(f_1^2 + f_2^2)^2}{(f_1^2 + f_2^2)^2 + \Omega^2}$$

The largest eigenvalue is bounded by 1, and the smallest by $f_{\min}^4/\Omega^2$. The minimum frequency $f_{\min}$ is determined by the domain and the boundary conditions where we exclude the frequency zero, which yields a smallest eigenvalue zero of $\hat{C}$. That is not relevant if the right-hand side is in the image of the operator. So the condition number of the continuous variant of C based on Fourier analysis is bounded by $\Omega^2/f_{\min}^4$. Hence, it is finite for all $f_1, f_2 > f_{\min}$.

For the discrete case on a B-grid (for a motivation see [25]) the symbols of A and B are

$$\tilde{A} = \begin{pmatrix} \frac{4}{h^2}\left(\sin^2\left(\frac{f_1 h}{2}\right) + \sin^2\left(\frac{f_2 h}{2}\right)\right) & -\Omega \\ \Omega & \frac{4}{h^2}\left(\sin^2\left(\frac{f_1 h}{2}\right) + \sin^2\left(\frac{f_2 h}{2}\right)\right) \end{pmatrix}$$

$$\tilde{B} = \begin{pmatrix} \frac{2i}{h}\cos\left(\frac{f_2 h}{2}\right)\sin\left(\frac{f_1 h}{2}\right) \\ \frac{2i}{h}\cos\left(\frac{f_1 h}{2}\right)\sin\left(\frac{f_2 h}{2}\right) \end{pmatrix}$$

which leads in an analogous way to the symbol

$$\tilde{C} = \frac{\left(\sin^2\left(\frac{f_1 h}{2}\right) + \sin^2\left(\frac{f_2 h}{2}\right)\right)\left(\cos^2\left(\frac{f_2 h}{2}\right)\sin^2\left(\frac{f_1 h}{2}\right) + \cos^2\left(\frac{f_1 h}{2}\right)\sin^2\left(\frac{f_2 h}{2}\right)\right)}{\left(\sin^2\left(\frac{f_1 h}{2}\right) + \sin^2\left(\frac{f_2 h}{2}\right)\right)^2 + \Omega^2}$$

DOI: 10.1002/fld

In this case the largest eigenvalue is bounded by $4/(4+\Omega^2)$ which is by itself less than one. For the smallest eigenvalue we have two cases from which we have to take the minimum. The first one is the same as that of the continuous case and the second occurs if the cosines in the numerator are close to zero. In any case the minimum eigenvalue is at least the ratio of the minimum of the numerator divided by the maximum of the denominator. The latter is simply $4+\Omega^2$ and an elementary analysis shows that the sought minimum of the numerator occurs if $f_1 = f_2$ and f_1 is closest to π/h (i.e. the highest frequency that can be represented on the grid). The numerator will then have a value approximately $(\pi - f_{\max}h)^2$, hence the condition number in the discrete case is

$$\max(\Omega^2/f_{\min}^4, 4/(\pi - f_{\max}h)^2)$$

Now, if the mesh size decreases $f_{\max}$ will tend to π/h, we exclude here the sawtooth Fourier component which gives a smallest eigenvalue zero. Hence, the second part in the max expression will dominate the first if the mesh size becomes small. However, in ocean models we have quite large mesh sizes and currently we do not encounter problems due to a dominating second part.

ACKNOWLEDGEMENTS

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

REFERENCES

1. Saad Y. *Iterative Methods for Sparse Linear Systems*. SIAM: Philadelphia, PA, 2003.
2. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
3. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
4. Vuik C, Saghir A. The Krylov accelerated SIMPLE(R) method for incompressible flow. *Report 02-01*, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, 2002.
5. Li C, Vuik C. Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow. *Numerical Linear Algebra with Applications* 2004; **11**(5–6):511–523.
6. Li C, Vuik C. Some results on the eigenvalue analysis of a SIMPLER preconditioned matrix. *Report 03-08*, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, 2003.
7. Silvester DJ, Wathen AJ. Fast iterative solution of stabilised Stokes systems. II. Using general block preconditioners. *SIAM Journal on Numerical Analysis* 1994; **31**(5):1352–1367.
8. Elman HC, Silvester DJ. Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations. *SIAM Journal on Scientific Computing* 1996; **17**(1):33–46.
9. Kay D, Loghin D, Wathen AJ. A preconditioner for the steady-state Navier–Stokes equations. *SIAM Journal on Scientific Computing* 2002; **24**(1):237–256 (electronic).
10. Elman HC, Silvester DJ, Wathen AJ. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations. *Numerische Mathematik* 2002; **90**(4):665–688.
11. Glowinski R, Le Tallec P. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM Studies in Applied Mathematics, vol. 9. SIAM: Philadelphia, PA, 1989.
12. Benzi M, Olshanskii MA. An augmented Lagrangian-based approach to the oseen problem. Submitted, preprint available on internet.
13. Olshanskii MA. A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: a stabilization issue and iterative methods. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**(47–48):5515–5536.
14. Olshanskii MA, Reusken A. Grad-div stabilization for Stokes equations. *Mathematics of Computation* 2004; **73**(248):1699–1718 (electronic).
15. Axelsson O. Preconditioning of indefinite problems by regularization. *SIAM Journal on Numerical Analysis* 1979; **16**(1):58–69.

DOI: 10.1002/fld

16. de Niet AC, Wubs FW. Two preconditioners for the saddle point equation. *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering* (ECCOMAS), Jyväskylä, 2004. See http://www.mit.jyu.fi/eccomas2004/
17. Dohrmann CR, Lehoucq RB. A primal-based penalty preconditioner for elliptic saddle point systems. *SIAM Journal on Numerical Analysis* 2006; **44**(1):270–282 (electronic).
18. Gartling DK, Dohrmann CR. Quadratic finite elements and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(13–16):1692–1708.
19. van der Vorst HA. *Iterative Krylov Methods for Large Linear Systems*. Cambridge Monographs on Applied and Computational Mathematics, vol. 13. Cambridge University Press: Cambridge, 2003.
20. Botta EFF, Wubs FW. Matrix renumbering ILU: an effective algebraic multilevel ILU preconditioner for sparse matrices. *SIAM Journal on Matrix Analysis and Applications* 1999; **20**(4):1007–1026; *Sparse and Structured Matrices and their Applications*. Coeur d'Alene: ID, 1996.
21. Elman HC, Silvester DJ, Wathen AJ. *Finite Elements and Fast Iterative Solvers*: *With Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press: New York, 2005.
22. van der Vorst HA. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**(2):631–644.
23. Saad Y. A flexible inner–outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* 1993; **14**(2):461–469.
24. Wilbert W, Dijkstra HA, Öksüzoğlu H, Wubs FW, de Niet AC. A fully-implicit model of the global ocean circulation. *Journal of Computational Physics* 2003; **192**:452–470.
25. Wubs FW, de Niet AC, Dijkstra HA. The performance of implicit ocean models on B- and C-grids. *Journal of Computational Physics* 2006; **211**:210–228.

DOI: 10.1002/fld