Science and Technology Facilities Council

Implicit-factorization preconditioners for non-symmetric problems

Technical Report 2047353-TN-04

Maksims Abaļenkovs* Vassil Alexandrov* Anton Lebedev* Emre Sahin* Sue Thorne**

July 2021

1 Introduction

In this report, we extend the class of constraint preconditioners from symmetric problems to non-symmetric problems. We consider the theoretical properties and demonstrate their effectiveness on a set of test problems inspired by the Hasegawa-Wakatani problem.

2 Constraint-style preconditioners

Let us assume that

$$\mathscr{A} = \begin{pmatrix} H & C \\ B & -D \end{pmatrix}, \tag{1}$$

where $H \in \mathbb{R}^{n\times n}$, $B, C^T \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$ subject to $m \le n$. We always assume that $\mathscr{A}$ is non-singular. We consider the use of a preconditioner of the form

$$\mathscr{P} = \begin{pmatrix} G & C \\ B & -D \end{pmatrix}, \tag{2}$$

where $G \in \mathbb{R}^{n\times n}$.

2.1 Constraint-style preconditioners: symmetric case

The case when $D = 0$, $B = C^T$ and $H = H^T$ was analysed by Keller, Gould and Wathen [6]:

Theorem 2.1. *Let $\mathscr{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ be a symmetric and indefinite matrix of the form*

$$\mathscr{A} = \begin{pmatrix} H & B^T \\ B & 0 \end{pmatrix},$$

where $H \in \mathbb{R}^{n\times n}$ is symmetric and $B \in \mathbb{R}^{m\times n}$ is of full rank. Assume Z is an $n \times (n-m)$ basis for the nullspace of B. Preconditioning $\mathscr{A}$ by a matrix of the form

$$\mathscr{P} = \begin{pmatrix} G & B^T \\ B & 0 \end{pmatrix},$$

where $G \in \mathbb{R}^{n\times n}$ is symmetric, and $B \in \mathbb{R}^{m\times n}$ is as above, implies that the matrix $\mathscr{P}^{-1}\mathscr{A}$ has

*The authors are with the Hartree Centre, STFC Daresbury Laboratory, Sci-Tech Daresbury, Keckwick, Daresbury, Warrington, WA4 4AD, UK.

**Sue Thorne is with the Hartree Centre, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK. Email contact: sue.thorne@stfc.ac.uk

1. *an eigenvalue at 1 with multiplicity* $2m$;
2. $n-m$ *eigenvalues* λ *which are defined by the generalized eigenvalue problem*

$$Z^T HZx_z = \lambda Z^T GZx_z. \tag{3}$$

This accounts for all of the eigenvalues.

Assume, in addition, that $Z^T GZ$ *is positive definite. Then* $\mathcal{P}^{-1}\mathcal{A}$ *has the following* $m+i+j$ *linearly independent eigenvectors:*

1. m *eigenvectors of the form* $\left[0^T, y^T\right]^T$ *corresponding to the eigenvalue 1 of* $\mathcal{P}^{-1}\mathcal{A}$;
2. i $(0 \le i \le n)$ *eigenvectors of the form* $\left[w^T, y^T\right]^T$ *corresponding to the eigenvalue 1 of* $\mathcal{P}^{-1}\mathcal{A}$*, where the components* w *arise from the generalized eigenvalue problem* $Hw = Gw$;
3. j $\left(0 \le j \le n-m\right)$ *eigenvectors of the form* $\left[x_z^T, 0^T, y^T\right]^T$ *corresponding to the eigenvalues of* $\mathcal{P}^{-1}\mathcal{A}$ *note equal to 1, where the components* x_z *arise from the generalized eigenvalue problem* $Z^T HZx_z = \lambda Z^T GZx_z$ *with* $\lambda \ne 1$.

The case when when $B = C^T$, $H = H^T$ and D symmetric and positive definite has been analysed by a number of different authors [2, 3, 4] and can be summarised in the following theorems:

Theorem 2.2. *Let* $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ *be a symmetric and indefinite matrix of the form*

$$\mathcal{A} = \begin{pmatrix} H & B^T \\ B & -D \end{pmatrix},$$

where $H \in \mathbb{R}^{n\times n}$ *is symmetric,* $B \in \mathbb{R}^{m\times n}$ *is of full rank and* $D \in \mathbb{R}^{m\times m}$ *is symmetric and positive definite. Preconditioning* $\mathcal{A}$ *by a matrix of the form*

$$\mathcal{P} = \begin{pmatrix} G & B^T \\ B & -D \end{pmatrix},$$

where $G \in \mathbb{R}^{n\times n}$ *is symmetric, and* $B \in \mathbb{R}^{m\times n}$ *and* $D \in \mathbb{R}^{m\times m}$ *are as above, implies that the matrix* $\mathcal{P}^{-1}\mathcal{A}$ *has*

1. *an eigenvalue at 1 with multiplicity* m;
2. n *eigenvalues* λ *which are defined by the generalized eigenvalue problem*

$$\left(H + B^T D^{-1} B\right) x = \lambda \left(G + B^T D^{-1} B\right) x. \tag{4}$$

This accounts for all of the eigenvalues.

Dollar *et al.* [4] have extended Theorem 2.2 to the case when D is symmetric and positive semi-definite:

Theorem 2.3. *Let* $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ *be a symmetric and indefinite matrix of the form*

$$\mathcal{A} = \begin{pmatrix} H & B^T \\ B & -D \end{pmatrix},$$

where $H \in \mathbb{R}^{n\times n}$ *is symmetric,* $B \in \mathbb{R}^{m\times n}$ *is of full rank and* $D \in \mathbb{R}^{m\times m}$ *is symmetric and positive semi-definite with rank* l*, where* $0 < l < m$*. Assume that* D *is factored as* $D = ESE^T$*, where* $E \in \mathbb{R}^{m\times l}$ *and* $S \in \mathbb{R}^{l\times l}$ *is nonsingular,* $F \in \mathbb{R}^{m\times(m-l)}$ *is a vasis for the nullspace of* E^T *and* $\begin{bmatrix} E & F \end{bmatrix}$ *is orthogonal. Let the columns of* $N \in \mathbb{R}^{n\times(n-m+l)}$ *span the nullspace of* $F^T B$*. Preconditioning* $\mathcal{A}$ *by a matrix of the form*

$$\mathcal{P} = \begin{pmatrix} G & B^T \\ B & -D \end{pmatrix},$$

where $G \in \mathbb{R}^{n\times n}$ *is symmetric, and* $B \in \mathbb{R}^{m\times n}$ *and* $D \in \mathbb{R}^{m\times m}$ *are as above, implies that the matrix* $\mathcal{P}^{-1}\mathcal{A}$ *has*

1. *an eigenvalue at 1 with multiplicity* $2m-l$;
2. $n-m+l$ *eigenvalues* λ *which are defined by the generalized eigenvalue problem*

$$N^T \left(H + B^T ES^{-1} E^T B\right) Nz = \lambda N^T \left(G + B^T ES^{-1} E^T B\right) Nz. \tag{5}$$

This accounts for all of the eigenvalues.

2.2 Constraint-style preconditioners: nonsymmetric case

We will now extend Theorems 2.1 and 2.2 to the non-symmetric case.

2.3 D non-singular

Theorem 2.4. *Let $\mathscr{A} \in \mathbb{R}^{(n+m)\times(n+m)}$, $m \le n$, be a matrix of the form*

$$\mathscr{A} = \begin{pmatrix} H & C \\ B & -D \end{pmatrix},$$

where $H \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{n\times m}$ are of full rank and $D \in \mathbb{R}^{m\times m}$ is non-singular. Preconditioning $\mathscr{A}$ by a matrix of the form

$$\mathscr{P} = \begin{pmatrix} G & C \\ B & -D \end{pmatrix},$$

where $G \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{m\times m}$ are as above, implies that the matrix $\mathscr{P}^{-1}\mathscr{A}$ has

1. *an eigenvalue at 1 with multiplicity m;*
2. *n eigenvalues λ which are defined by the generalized eigenvalue problem*

$$\left(H + CD^{-1}B\right)x = \lambda\left(G + CD^{-1}B\right)x. \tag{6}$$

This accounts for all of the eigenvalues.

Proof. The eigenvalues of $\mathscr{P}^{-1}\mathscr{A}$ may be derived by considering the generalized eigenvalue problem

$$\begin{pmatrix} H & C \\ B & -D \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} G & C \\ B & -D \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{7}$$

Premultiplying (7) by the non-singular matrix

$$\begin{pmatrix} I & CD^{-1} \\ 0 & -D^{-1} \end{pmatrix}$$

gives the equivalent generalized eigenvalue problem

$$\begin{pmatrix} H + CD^{-1}B & 0 \\ -D^{-1}B & I \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} G + CD^{-1}B & 0 \\ -D^{-1}B & I \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$$

Thus, there are m eigenvalues equal to 1 and the remaining n eigenvalues are defined by the generalized eigenvalue problem

$$\left(H + CD^{-1}B\right)x = \lambda\left(G + CD^{-1}B\right)x. \tag{8}$$

□

2.4 $D = 0$

Theorem 2.5. *Let $\mathscr{A} \in \mathbb{R}^{(n+m)\times(n+m)}$, $m \le n$, be a matrix of the form*

$$\mathscr{A} = \begin{pmatrix} H & C \\ B & 0 \end{pmatrix},$$

where $H \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{n\times m}$ are of full rank. Let the columns of $Z_B \in \mathbb{R}^{n\times(n-m)}$ span the nullspace of B and the columns of $Z_C \in \mathbb{R}^{n\times(n-m)}$ span the nullspace of C^T. Preconditioning $\mathscr{A}$ by a matrix of the form

$$\mathscr{P} = \begin{pmatrix} G & C \\ B & 0 \end{pmatrix},$$

where $G \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{n\times m}$ are as above, implies that the matrix $\mathscr{P}^{-1}\mathscr{A}$ has

1. *$2m$ eigenvalues of equal to 1;*

2. *the remaining $n-m$ eigenvalues, λ, are defined by the generalized eigenvalue problem*

$$Z_C^T H Z_B x_z = \lambda Z_C^T G Z_B x_z. \tag{9}$$

This accounts for all of the eigenvalues.

Proof. The eigenvalues of $\mathscr{P}^{-1}\mathscr{A}$ may be derived by considering the generalized eigenvalue problem

$$\begin{pmatrix} H & C \\ B & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} G & C \\ B & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}, \tag{10}$$

where $\lambda \in \mathbb{C}$, $\lambda \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}^m$. Let

$$B = U_B \begin{pmatrix} \Sigma_B & 0 \end{pmatrix}\begin{pmatrix} Y_B^T \\ Z_B^T \end{pmatrix}, \quad C^T = U_C \begin{pmatrix} \Sigma_C & 0 \end{pmatrix}\begin{pmatrix} Y_C^T \\ Z_C^T \end{pmatrix}$$

be the singular-value decompositions of B and C with $Y_B, Y_C \in \mathbb{R}^{n\times m}$. Note that $Z_B, Z_C \in \mathbb{R}^{n\times(n-m)}$ span the nullspace of B and C^T, respectively.

If we substitute in $x = Y_B x_Y + Z_B x_Z$ into (10) and premultiply the equation by the nonsingular matrix

$$\begin{pmatrix} Y_C^T & 0 \\ Z_C^T & 0 \\ 0 & I \end{pmatrix},$$

where Y_B and Y_C are n by m matrices whose columns span the range space of B^T and C, respectively, then we obtain

$$\begin{pmatrix} Y_C^T H Y_B & Y_C^T H Z_B & Y_C^T C \\ Z_C^T H Y_B & Z_C^T H Z_B & 0 \\ B Y_B & 0 & 0 \end{pmatrix}\begin{pmatrix} x_Y \\ x_Z \\ y \end{pmatrix} = \lambda \underbrace{\begin{pmatrix} Y_C^T G Y_B & Y_C^T G Z_B & Y_C^T C \\ Z_C^T G Y_B & Z_C^T G Z_B & 0 \\ B Y_B & 0 & 0 \end{pmatrix}}_{\tilde{\mathscr{P}}}\begin{pmatrix} x_Y \\ x_Z \\ y \end{pmatrix}. \tag{11}$$

If we pre-multiply (11) by $\tilde{\mathscr{P}}^{-1}$, then we obtain an equivalent eigenvalue problem of the form

$$\begin{pmatrix} I & 0 & 0 \\ \Theta_1 & (Z_C^T G X_B)^{-1} Z_C^T H Z_B & 0 \\ \Theta_2 & \Theta_3 & I \end{pmatrix}\begin{pmatrix} x_Y \\ x_Z \\ y \end{pmatrix} = \lambda \begin{pmatrix} x_Y \\ x_Z \\ y \end{pmatrix}, \tag{12}$$

where the exact definition of Θ_1, Θ_2 and Θ_3 are not important for the proof. Hence, $\mathscr{P}^{-1}\mathscr{A}$ has $2m$ eigenvalues equal to 1 and the remaining eigenvalues are defined by the eigenvalue problem generalized eigenvalue problem (9).

□

We note that when $\mathscr{A}$ and $\mathscr{P}$ are no-longer symmetric, some of the non-unitary eigenvalues may be complex.

3 Implicit-factorization constraint preconditioners

In [4], the authors derive a number of factorizations for generating symmetric constraint preconditioners. In the following, we will assume that the rows and columns of H have been ordered in such a manner that we can partition $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{n\times m}$, $G \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{n\times n}$ as

$$B = \begin{pmatrix} B_1 & B_2 \end{pmatrix}, \tag{13}$$

$$C = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}, \tag{14}$$

$$G = \begin{pmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{pmatrix}, \tag{15}$$

$$H = \begin{pmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{pmatrix}, \tag{16}$$

where $B_1 \in \mathbb{R}^{m\times m}$ and $C_1 \in \mathbb{R}^{m\times m}$ are non-singular, $G_{1,1} \in \mathbb{R}^{m\times m}$ and $H_{1,1} \in \mathbb{R}^{m\times m}$. For coupled multi-physics problems, this ordering is implicitly available through the nature of the problems. As in [4], we form factors of the form

$$\begin{aligned} L &= \begin{pmatrix} L_{1,1} & L_{1,2} & L_{1,3} \\ L_{2,1} & L_{2,2} & L_{2,3} \\ L_{3,1} & L_{3,2} & L_{3,3} \end{pmatrix}, \\ N &= \begin{pmatrix} N_{1,1} & N_{1,2} & N_{1,3} \\ N_{2,1} & N_{2,2} & N_{2,3} \\ N_{3,1} & N_{3,2} & N_{3,3} \end{pmatrix}, \\ R &= \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{pmatrix} \end{aligned}$$

set some of the sub-blocks to zero whilst assuming other sub-blocks are invertible and relatively easy to solve with, and the sub-blocks are such that the product LNR forms a non-symmetric constraint preconditioner of the form

$$\mathscr{P} = \begin{pmatrix} G & C \\ B & -D \end{pmatrix}.$$

Without loss of generality, we fix $L_{1,3}$, $L_{2,2}$, $L_{2,3}$, $R_{2,2}$, $R_{3,1}$ and $R_{3,2}$ to be non-zero with $L_{2,2}$ and $R_{2,2}$ both non-singular. We use a Matlab script (see Appendix A) to generate all 62 possible implicit-factorization constraint preconditioners. We note that if $B = C^T$, $R_{3,1} = B_1$, $R_{3,2} = B_2$, $L_{1,3} = B_1^T$ and $L_{2,3} = B_2^T$, then we obtain the families given in [4].

Some of the non-symmetric implicit factorizations are only suitable for the case $D = 0$, for example

$$\begin{aligned} L &= \begin{pmatrix} L_{1,1} & 0 & L_{1,3} \\ L_{2,1} & L_{2,2} & L_{2,3} \\ L_{3,1} & 0 & 0 \end{pmatrix}, \\ N &= \begin{pmatrix} 0 & 0 & N_{1,3} \\ 0 & N_{2,2} & N_{2,3} \\ N_{3,1} & N_{3,2} & N_{3,3} \end{pmatrix}, \\ R &= \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ 0 & R_{2,2} & 0 \\ R_{3,1} & R_{3,2} & 0 \end{pmatrix} \end{aligned}$$

subject to

$$\begin{aligned} L_{3,1}N_{1,3}R_{3,1} &= B_1, \\ B_1R_{3,1}^{-1}R_{3,2} &= B_2, \\ L_{1,3}N_{3,1}R_{1,3} &= C_1, \\ L_{2,3}L_{1,3}^{-1}C_1 &= C_2 \end{aligned}$$

produces

$$\begin{aligned} G_{1,1} &= L_{1,1}N_{1,3}R_{3,1} + L_{1,3}N_{3,3}R_{3,1} + L_{1,3}N_{3,1}R_{1,1}, \\ G_{1,2} &= L_{1,1}L_{3,1}^{-1}B_2 + L_{1,3}N_{3,3}R_{3,1}B_1^{-1}B_2 + C_1R_{1,3}^{-1}R_{1,2} + L_{1,3}N_{3,2}R_{2,2}, \\ G_{2,1} &= L_{2,1}L_{3,1}^{-1}B_1 + L_{2,2}N_{2,3}R_{3,1} + C_2C_1^{-1}L_{1,3}N_{3,3}R_{3,1} + C_2R_{1,3}^{-1}R_{1,1}, \\ G_{2,2} &= L_{2,2}N_{2,2}R_{2,2} + C_2C_1^{-1}L_{1,3}N_{3,2}R_{2,2} + L_{2,1}L_{3,1}^{-1}B_2 + C_2C_1^{-1}L_{1,3}N_{3,3}R_{3,1}B_1^{-1}B_2 \\ &\quad + L_{2,2}N_{2,3}R_{3,1}B_1^{-1}B_2. \end{aligned}$$

There are also some that are only suitable for non-singular D, for example,

$$
\begin{aligned}
L &= \begin{pmatrix} L_{1,1} & 0 & L_{1,3} \\ L_{2,1} & L_{2,2} & L_{2,3} \\ L_{3,1} & 0 & 0 \end{pmatrix}, \\
N &= \begin{pmatrix} 0 & 0 & N_{1,3} \\ 0 & N_{2,2} & N_{2,3} \\ N_{3,1} & N_{3,2} & N_{3,3} \end{pmatrix}, \\
R &= \begin{pmatrix} R_{1,1} & R_{1,2} & 0 \\ 0 & R_{2,2} & 0 \\ R_{3,1} & R_{3,2} & R_{3,3} \end{pmatrix}
\end{aligned}
$$

subject to

$$
\begin{aligned}
L_{3,1}N_{1,3}R_{3,1} &= B_1, \\
B_1 R_{3,1}^{-1} R_{3,2} &= B_2, \\
L_{3,1}N_{1,3}R_{33} &= D, \\
-\left(L_{1,1}N_{1,3} + L_{1,3}N_{3,3}\right) R_{3,1} &= C_1 D^{-1} B_1, \\
-\left(L_{2,1}N_{1,3} + L_{2,3}N_{3,3} + L_{2,2}N_{2,3}\right) R_{3,1} R_{3,1} &= C_1 D^{-1} B_1
\end{aligned}
$$

produces

$$
\begin{aligned}
G_{1,1} &= -C_1 D^{-1} B_1 + L_{1,3}N_{3,1}R_{1,1}, \\
G_{1,2} &= -C_1 D^{-1} B_2 + \left(G_{1,1} + C_1 D^{-1} B_1\right) R_{1,1}^{-1} R_{1,2} + L_{1,3}N_{3,2}R_{2,2}, \\
G_{2,1} &= -C_2 D^{-1} B_1 + L_{2,3}N_{3,1}R_{1,1}, \\
G_{2,2} &= L_{2,2}N_{2,2}R_{2,2} + L_{2,3}N_{3,2}R_{2,2} + C_2 C^{-1} B_2 + L_{2,3}N_{3,1}R_{1,2}.
\end{aligned}
$$

4 Numerical tests

We will consider a test problem inspired by the 2D problem known as the Hasegawa-Wakatani problem, which is similar to incompressible fluid dynamics:

$$
\begin{aligned}
\frac{\partial n}{\partial t} &= -\{\phi, n\} + \alpha(\phi - n) - \kappa \frac{\partial \phi}{\partial z} + D_n \nabla_\perp^2 n \\
\frac{\partial \omega}{\partial t} &= -\{\omega, n\} + \alpha(\omega - n) + D_\omega \nabla_\perp^2 \omega \\
\nabla^2 \phi &= \omega\,.
\end{aligned}
$$

Here n is the plasma number density, $\omega := \vec{b}_0 \cdot \nabla \times \vec{v}$ is the vorticity with $\vec{v}$ being the $\vec{E} \times \vec{B}$ drift velocity in a constant magnetic field and $\vec{b}_0$ is the unit vector in the direction of the equilibrium magnetic field. The operator $\{\cdot,\cdot\}$ is the Poisson bracket.

The discretized version of the problem is described in [1] but we will consider a split implicit-explicit method where the Jacobian that needs solving at each Newton iteration is of the following form:

$$
J = \begin{pmatrix} A & 0 & B \\ 0 & C & E \\ -M & K & 0 \end{pmatrix}, \tag{17}
$$

where the constituent matrices are the following

$$
\begin{aligned}
A &= M + \Delta t\,(-D_\omega K)\,, \\
B &= \alpha \Delta t M, \\
C &= \Delta t\,(-\alpha M)\,, \\
E &= M + \Delta t\,(\alpha M - D_n K)\,.
\end{aligned}
$$

Here K and M are the stiffness and mass matrices, respectively. Note that we have permuted the rows and columns so the matrix will not directly map to that given in [1]. We tried to use BOUT++[5] directly to solve the Hasegawa-Wakatani problem and test our preconditioners but using PETSc with a constraint and preconditioner resulted in runtime errors, which might be due to the manner that PETSc was installed on the Hartree Centre's Scafell Pike cluster. Instead, we took advantage of the situation and created mass and stiffness matrices that use a finite-element discretization instead of finite difference. We used the same values of constants as used within the BOUT++ implementation and set Δt to be equal to the inverse of the number of rows in M.

We will compare the following preconditioning strategies:

- A block-diagonal preconditioner
$$P_D = \begin{pmatrix} A & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & I \end{pmatrix};$$
- A constraint preconditioner with $G = I$
$$P_1 = \begin{pmatrix} I & 0 & B \\ 0 & I & E \\ -M & K & 0 \end{pmatrix};$$
- A constraint preconditioner with $G_{2,2} = I$ and the remainder of G zero:
$$P_2 = \begin{pmatrix} 0 & 0 & B \\ 0 & I & E \\ -M & K & 0 \end{pmatrix};$$
- A constraint preconditioner with $G_{2,2} = C$ and the remainder of G zero:
$$P_3 = \begin{pmatrix} 0 & 0 & B \\ 0 & C & E \\ -M & K & 0 \end{pmatrix};$$
- An implicit-factorization constraint preconditioner with:
$$\begin{aligned} L &= \begin{pmatrix} -I & 0 & I \\ -\frac{D_\omega D_n \Delta t}{\alpha} K M^{-1} K M^{-1} & I & \frac{1}{\alpha \Delta t}\left((1+\alpha\gamma)M - \gamma D_n K\right) M^{-1} \\ I & 0 & 0 \end{pmatrix}, \\ N &= \begin{pmatrix} 0 & 0 & -M \\ 0 & -\frac{1}{\alpha\Delta t}\left((1+\alpha\gamma)M - \gamma D_n K\right) M^{-1} K & \frac{D_\omega(1+\alpha\Delta t)}{\alpha} K \\ \alpha \Delta t M & K & -\gamma D_\omega K \end{pmatrix}, \\ R &= \begin{pmatrix} 0 & -\frac{D_\omega}{\alpha} M^{-1} K M^{-1} K & I \\ 0 & I & 0 \\ I & -M^{-1} K & 0 \end{pmatrix}. \end{aligned}$$

Note that, with the exception of preconditioner P_1, we do not explicitly form the preconditioner and we instead apply them by exploiting the block structures. In Tables 1 and 2, we report the number of iterations to reduce the relative residual by a factor of 10^{-6} and the times for solving our test problems using Matlab's GMRES function with no restarting. Note that the preconditioners have not been optimized with respect to time so these values are only indicative. Preconditioners P_1 and P_5 produce the best iteration counts but we note that for larger problems, factoring P_1 via a direct method will become extremely expensive. Additionally, alternative choices for the blocks in the implicit factorization preconditioner may increase the number of iterations but make the preconditioner much cheaper to apply. For example, solves with the mass matrix can be well-approximated using the Chebyshev semi-iteration [7] and solves involving the stiffness matrix may be approximated with a multigrid method: this was very successfully done within the symmetric constraint preconditioner context for PDE-constrained problems [8].

In Tables 1 and 2, we report the number of iterations to reduce the relative residual by a factor of 10^{-6} and the times for solving our test problems using Matlab's GMRES function with restarting set to 10. Here, preconditioner P_2 failed to converge but we see similar results to non-restarted GMRES for preconditioners P_1 and P_5. Note that by using the restarted version of GMRES, we were able to solve larger problems.

n	m	P_D	P_1	P_2	P_3	P_4
450	225	57	3	43	61	2
1922	961	103	2	86	122	2
7938	3969	192	2	172	239	2

Table 1: Number of preconditioned GMRES iterations to reduce the relative residual by a factor of 10^{-6}.

n	m	P_D	P_1	P_2	P_3	P_4
450	225	0.029	0.020	0.037	0.051	0.015
1922	961	0.21	0.077	0.17	0.31	0.27
7938	3969	2.18	0.37	0.37	3.97	8.63

Table 2: Time (in seconds) for preconditioned GMRES to reduce the relative residual by a factor of 10^{-6}.

n	m	P_D	P_1	P_2	P_3	P_4
450	225	404	3	-	15	2
1922	961	725	2	-	649	2
7938	3969	2078	2	-	2255	2
32258	16129	4514	2	-	7875	2

Table 3: Number of preconditioned GMRES(10) iterations to reduce the relative residual by a factor of 10^{-6}.

n	m	P_D	P_1	P_2	P_3	P_4
450	225	0.17	0.020	-	0.12	0.015
1922	961	1.19	0.083	-	1.58	0.27
7938	3969	20.3	0.37	-	33.4	8.70
32258	16129	214	1.83	-	622	324

Table 4: Time (in seconds) for preconditioned GMRES(10) to reduce the relative residual by a factor of 10^{-6}.

5 Conclusions

We conclude by observing that our results demonstrate the effectiveness of using non-symmetric constraint preconditioners. By careful selection of the constraint preconditioner, we have shown that they can be applied in an operator-based manner either by using very simple choices of G or by using an implicit-factorization. The next step will be to incorporate these preconditioners into BOUT++ and Nektar++ [9] to see how they perform within a non-linear simulation.

References

[1] V. Alexandrov, A. Lebedev, E. Sahin, and S. Thorne. Linear systems of equations and preconditioners relating to the NEPTUNE Programme: a brief overview. Technical Report 2047353-TN-02, UKAEA, 2021.

[2] O. Axelsson and M. Neytcheva. Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical linear algebra with applications*, 10(1-2):3–31, 2003.

[3] L. Bergamaschi, J. Gondzio, and G. Zilli. Preconditioning indefinite systems in interior point methods for optimization. *Computational Optimization and Applications*, 28(2):149–171, 2004.

[4] H. S. Dollar, N. I. Gould, W. H. Schilders, and A. J. Wathen. Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems. *SIAM Journal on Matrix Analysis and Applications*, 28(1):170–189, 2006.

[5] B. Dudson, P. Hill, and J. Parker. BOUT++. online repository, 2020. URL `http://boutproject.github.io`.

[6] C. Keller, N. Gould, and A. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, 21:1300–1317, 2000.

[7] T. Rees and A. J. Wathen. Optimal solvers for pde-constrained optimizationchebyshev semi-iteration in preconditioning for problems including the mass matrix. *Electronic Transactions on Numerical Analysis*, 34, 2008.

[8] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for pde-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010. doi: 10.1137/080727154. URL `https://doi.org/10.1137/080727154`.

[9] S. Sherwin, M. Kirby, C. Cantwell, and D. Moxey. Nektar++. online, 2021. URL `https://www.nektar.info`.

Appendix A: Matlab Script

```
% Generates non-symmetric implicit-factorization constraint preconditioner
% families
format compact
Ll = [sym('l11'),sym('l12'), sym('l13'),sym('l21'),sym('l22'),...
    sym('l23'),sym('l31'),sym('l32'),sym('l33') ];
Rr = [sym('r11'),sym('r12'), sym('r13'),sym('r21'),sym('r22'),...
    sym('r23'),sym('r31'),sym('r32'),sym('r33') ]
Mm = [sym('m11'),sym('m12'), sym('m13'),sym('m21'),sym('m22'),...
    sym('m23'),sym('m31'),sym('m32'),sym('m33') ]
total = 0;

for i=1:5
    for j=1:3
        for k=1:5
            L = [Ll(1:3);Ll(4:6);Ll(7:9)];
            R = [Rr(1:3);Rr(4:6);Rr(7:9)];
            M = [Mm(1:3);Mm(4:6);Mm(7:9)];
            switch i
                case 1
                    L(1,1:2)=0; L(2,1)=0;
                case 2
                    L(1,1:2)=0; L(3,2)=0;
                case 3
                    L(1,2)=0; L(3,1:2)=0;
                case 4
                    L(2,1)=0; L(3,1:2)=0;
                case 5
                    L(1,2)=0; L(3,2:3)=0;
            end

            switch j
                case 1
                    M(3,2:3)=0; M(2,3)=0;
                case 2
                    M(1,1:2)=0; M(2,1)=0;
                case 3
                    M(1,2)=0; M(2,1)=0; M(2,3)=0; M(3,2)=0;
            end

            switch k
                case 1
                    R(1:2,1)=0; R(1,2)=0;
                case 2
                    R(1:2,1)=0; R(2,3)=0;
                case 3
                    R(2,1)=0; R(2:3,3)=0;
                case 4
                    R(2,1)=0; R(1:2,3)=0;
                case 5
                    R(1,2)=0; R(1:2,3)=0;
            end
            p = 1;
            F = L*M*R;
            if ((F(1,3)==0) | (F(2,3)==0) | (F(3,1)==0) | (F(3,2)==0))
```

```
                p=0;
            end

            if (p==1)
                total = total+1;
               % [i,j,k]
               factor = total
                struct=[L,M,R]
                F
            end
        end
    end
end
total
```