

Two-level Nystrom-Schur preconditioner for sparse symmetric positive definite matrices

Article

Accepted Version

Al Daas, H., Rees, T. and Scott, J. ORCID:
<https://orcid.org/0000-0003-2130-1091> (2021) Two-level
Nystrom-Schur preconditioner for sparse symmetric positive
definite matrices. SIAM Journal on Scientific Computing, 43
(6). A3837-A3861. ISSN 1095-7197 doi:
<https://doi.org/10.1137/21M139548X> Available at
<https://centaur.reading.ac.uk/99495/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

To link to this article DOI: <http://dx.doi.org/10.1137/21M139548X>

Publisher: Society for Industrial and Applied Mathematics

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

TWO-LEVEL NYSTRÖM–SCHUR PRECONDITIONER FOR SPARSE SYMMETRIC POSITIVE DEFINITE MATRICES*

HUSSAM AL DAAS[†], TYRONE REES[†], AND JENNIFER SCOTT^{†‡}

Abstract. Randomized methods are becoming increasingly popular in numerical linear algebra. However, few attempts have been made to use them in developing preconditioners. Our interest lies in solving large-scale sparse symmetric positive definite linear systems of equations where the system matrix is preordered to doubly bordered block diagonal form (for example, using a nested dissection ordering). We investigate the use of randomized methods to construct high quality preconditioners. In particular, we propose a new and efficient approach that employs Nyström’s method for computing low rank approximations to develop robust algebraic two-level preconditioners. Construction of the new preconditioners involves iteratively solving a smaller but denser symmetric positive definite Schur complement system with multiple right-hand sides. Numerical experiments on problems coming from a range of application areas demonstrate that this inner system can be solved cheaply using block conjugate gradients and that using a large convergence tolerance to limit the cost does not adversely affect the quality of the resulting Nyström–Schur two-level preconditioner.

Key words. Randomized methods, Nyström’s method, Low rank, Schur complement, Deflation, Sparse symmetric positive definite systems, Doubly bordered block diagonal form, Block Conjugate Gradients, Preconditioning.

1. Introduction. Large scale linear systems of equations arise in a wide range of real-life applications. Since the 1970s, sparse direct methods, such as LU, Cholesky, and LDLT factorizations, have been studied in depth and library quality software is available (see, for example, [9] and the references therein). However, their memory requirements and the difficulties in developing effective parallel implementations can limit their scope for solving extremely large problems, unless they are used in combination with an iterative approach. Iterative methods are attractive because they have low memory requirements and are simpler to parallelize. In this work, our interest is in using the conjugate gradient (CG) method to solve large sparse symmetric positive definite (SPD) systems of the form

$$(1.1) \quad Ax = b,$$

where $A \in \mathbb{R}^{n \times n}$ is SPD, $b \in \mathbb{R}^n$ is the given right-hand side, and x is the required solution. The solution of SPD systems is ubiquitous in scientific computing, being required in applications as diverse as least-squares problems, non-linear optimization subproblems, Monte-Carlo simulations, finite element analysis, and Kalman filtering. In the following, we assume no additional structure beyond a sparse SPD system.

It is well known that the approximate solution x_k at iteration k of the CG method satisfies

$$(1.2) \quad \|x_\star - x_k\|_A \leq 2\|x_\star - x_0\|_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k,$$

where x_\star is the exact solution, x_0 is the initial guess, $\|\cdot\|_A$ is the A -norm, and $\kappa(A) = \lambda_{\max}/\lambda_{\min}$ is the spectral condition number (λ_{\max} and λ_{\min} denote the largest and

*Submitted to the editors January 28, 2021.

[†]STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX, UK (hussam.al-daas@stfc.ac.uk, tyrone.rees@stfc.ac.uk, jennifer.scott@stfc.ac.uk).

[‡]School of Mathematical, Physical and Computational Sciences, University of Reading, Reading RG6 6AQ, UK

40 smallest eigenvalues of A). The rate of convergence also depends on the distribution
 41 of the eigenvalues (as well as on b and x_0): eigenvalues clustered away from the
 42 origin lead to rapid convergence. If $\kappa(A)$ is large and the eigenvalues of A are evenly
 43 distributed, the system needs to be preconditioned to enhance convergence. This
 44 can be done by applying a linear operator \mathcal{P} to (1.1), where $\mathcal{P} \in \mathbb{R}^{n \times n}$ is chosen so
 45 that the spectral condition number of $\mathcal{P}A$ is small and applying \mathcal{P} is inexpensive. In
 46 some applications, knowledge of the provenance of A can help in building an efficient
 47 preconditioner. Algebraic preconditioners do not assume such knowledge, and include
 48 incomplete Cholesky factorizations, block Jacobi, Gauss–Seidel, and additive Schwarz;
 49 see, for example, [36]. These are referred to as *one-level* or *traditional* preconditioners
 50 [7, 43]. In general, algebraic preconditioners bound the largest eigenvalues of $\mathcal{P}A$ but
 51 encounter difficulties in controlling the smallest eigenvalues, which can lie close to the
 52 origin, hindering convergence.

53 Deflation strategies have been proposed to overcome the issues related to small
 54 eigenvalues. As explained in [25], the basic idea behind deflation is to “hide” certain
 55 parts of the spectrum of the matrix from the CG method, such that the CG iteration
 56 “sees” a system that has a much smaller condition number than the original matrix.
 57 The part of the spectrum that is hidden from CG is determined by the deflation
 58 subspace and the improvement in the convergence rate of the deflated CG method is
 59 dependent on the choice of this subspace. In the ideal case, the deflation subspace
 60 is the invariant subspace spanned by the eigenvectors associated with the smallest
 61 eigenvalues of A and the convergence rate is then governed by the “effective” spectral
 62 condition number associated with the remaining eigenvalues (that is, the ratio of the
 63 largest eigenvalue to the smallest remaining eigenvalue). The idea was first introduced
 64 in the late 1980s [8, 33], and has been discussed and used by a number of researchers
 65 [2, 3, 10, 14, 22, 23, 27, 32, 40, 41, 45, 46]. However, in most of these references,
 66 the deflation subspaces rely on the underlying partial differential equation and its
 67 discretization, and cannot be applied to more general systems or used as “black box”
 68 preconditioners. Algebraic two-level preconditioners have been proposed in [4, 11,
 69 15, 30, 43, 44]. Recently, a two-level Schur complement preconditioner based on the
 70 power series approximation was proposed in [50].

71 In recent years, the study of randomized methods has become an active and
 72 promising research area in the field of numerical linear algebra (see, for example,
 73 [16, 31] and the references therein). The use of randomized methods to build
 74 preconditioners has been proposed in a number of papers, including [14, 18]. The
 75 approach in [14] starts by reordering the system matrix A to a 2×2 doubly
 76 bordered block diagonal form, which can be achieved using a nested dissection
 77 ordering. The Schur complement system must then be solved. Starting from
 78 a first-level preconditioner \mathcal{P} , a deflation subspace is constructed via a low rank
 79 approximation. Although deflation can be seen as a low rank correction, using
 80 randomized methods to estimate the low rank term is not straightforward because
 81 the deflation subspace is more likely to be associated with the invariant subspace
 82 corresponding to the smallest eigenvalues of the preconditioned matrix, and not to
 83 its dominant subspace. In section 2, we review the ingredients involved in building
 84 our two-level preconditioner. This includes Nyström’s method for computing a low
 85 rank approximation of a matrix [12, 16, 34, 47, 48], basic ideas behind deflation
 86 preconditioners, and the two-level Schur complement preconditioners presented in
 87 [14, 27]. In section 3, we illustrate the difficulties in constructing these two-level
 88 preconditioners by analysing the eigenvalue problems that must be solved. We show
 89 that these difficulties are mainly associated with the clustering of eigenvalues near

Identifier	n	$nnz(A)$	$\kappa(A)$	n_Γ	2D/3D	Application	Source
bcstkt38	8,032	355,460	5.5e+16	2,589	2D	Structural problem	SSMC
ela2d	45,602	543,600	1.5e+8	4,288	2D	Elasticity problem	FF++
ela3d	9,438	312,372	4.5e+5	4,658	3D	Elasticity problem	FF++
msc10848	10,848	1,229,776	1.0e+10	4,440	3D	Structural problem	SSMC
nd3k	9,000	3,279,690	1.6e+7	1,785	3D	Not available	SSMC
s3rmt3m3	5,357	207,123	2.4e+10	2,058	2D	Structural problem	SSMC

TABLE 1

Set of test matrices. n and $nnz(A)$ denote the order of A and the number of nonzero entries in A disregarding, $\kappa(A)$ is the spectral condition number, n_Γ is the order of the Schur complement (2.11). SSMC refers to SuiteSparse Matrix Collection [5]. FF++ refers to FreeFem++ [17].

90 the origin. Motivated by this analysis, in section 4 we propose reformulating the
91 approximation problem.

92 The new formulation leads to well-separated eigenvalues that lie away from
93 the origin, and this allows randomized methods to be used to compute a deflation
94 subspace. Our approach guarantees a user-defined upper bound on the expected value
95 of the spectral condition number of the preconditioned matrix. Numerical results for
96 our new preconditioner and comparisons with other approaches are given in section 5.
97 Concluding remarks are made in section 6.

98 Our main contributions are:

- 99 • an analysis of the eigenvalue problems and solvers presented in [14, 27];
- 100 • a reformulation of the eigenvalue problem so that it be efficiently solving using
101 randomized methods;
- 102 • a new two-level preconditioner for symmetric positive definite systems that
103 we refer to as a two-level Nyström–Schur preconditioner;
- 104 • theoretical bounds on the expected value of the spectral condition number of
105 the preconditioned system.

106 *Test environment.* In this study, to demonstrate our theoretical and practical
107 findings, we report on numerical experiments using the test matrices given in Table 1.
108 This set was chosen to include 2D and 3D problems having a range of densities and
109 with relatively large spectral condition numbers. In the Appendix, results are given
110 for a much larger set of matrices. For each test, the entries of the right-hand side
111 vector f are taken to be random numbers in the interval $[0, 1]$. All experiments are
112 performed using Matlab 2020b.

113 *Notation.* Throughout this article, matrices are denoted using uppercase letters;
114 scalars and vectors are lowercase. The pseudo inverse of a matrix C is denoted by C^\dagger
115 and its transpose is given by C^\top . $\Lambda(M)$ denotes the spectrum of the matrix M and
116 $\kappa(M)$ denotes its condition number. $\Lambda_k = \text{diag}(\lambda_1, \dots, \lambda_k)$ denotes a $k \times k$ diagonal
117 matrix with entries on the diagonal equal to $\lambda_1, \dots, \lambda_k$. \tilde{S} (with or without a subscript
118 or superscript) is used as an approximation to a Schur complement matrix. \mathcal{P} (with
119 or without a subscript) denotes a (deflation) preconditioner. \mathcal{M} (with or without
120 a subscript) denotes a two-level (deflation) preconditioner. Matrices with an upper
121 symbol such as \tilde{Z} , \hat{Z} , and \check{Z} denote approximations of the matrix Z . Euler’s constant
122 is denoted by e .

123 **2. Background.** We start by presenting a brief review of Nyström’s method for
124 computing a low rank approximation to a matrix and then recalling key ideas behind
125 two-level preconditioners; both are required in later sections.

126 **2.1. Nyström’s method.** Given a matrix G , the Nyström approximation of a
 127 SPSD matrix B is defined to be

$$128 \quad (2.1) \quad BG(G^\top BG)^\dagger(BG)^\top.$$

129 We observe that there are a large number of variants based on different choices of
 130 G (for example, [16, 28, 31]). For $q \geq 0$, the q -power iteration Nyström method is
 131 obtained by choosing

$$132 \quad (2.2) \quad G = B^q \Omega,$$

133 for a given (random) starting matrix Ω . Note that, in practice, for stability it is
 134 normally necessary to orthonormalize the columns between applications of B .

135 The variant of Nyström’s method we employ is outlined in [Algorithm 2.1](#). It gives
 136 a near-optimal low rank approximation to B and is particularly effective when the
 137 eigenvalues of B decay rapidly after the k -th eigenvalue [16, 31]. It requires only one
 138 matrix-matrix product with B (or $q + 1$ products if (2.2) is used). The rank of the
 139 resulting approximation is $\min(r, k)$, where r is the rank of D_1 , see Step 5.

Algorithm 2.1 Nyström’s method for computing a low rank approximation to a SPSD matrix.

Input: A SPSD matrix $B \in \mathbb{R}^{n \times n}$, the required rank $k > 0$, an oversampling parameter $p \geq 0$ such that $k, p \ll n$, and a threshold ε .

Output: $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{U}_k^\top \approx B$ where \tilde{U}_k is orthonormal $\tilde{\Sigma}_k$ is diagonal with non negative entries.

- 1: Draw a random matrix $G \in \mathbb{R}^{n \times (k+p)}$.
 - 2: Compute $F = BG$.
 - 3: Compute the QR factorization $F = QR$.
 - 4: Set $C = G^\top F$.
 - 5: Compute the EVD $C = V_1 D_1 V_1^\top + V_2 D_2 V_2^\top$, where D_1 contains all the eigenvalues that are at least ε .
 - 6: Set $T = R V_1 D_1^{-1} (R V_1)^\top$.
 - 7: Compute the EVD $T = W E W^\top$.
 - 8: Set $\tilde{U} = QW$, $\tilde{U}_k = \tilde{U}(:, 1:k)$, $\tilde{\Sigma} = E(1:k, 1:k)$, and $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{U}_k^\top$.
-

140 Note that, if the eigenvalues are ordered in descending order, the success of
 141 Nyström’s method is closely related to the ratio of the $(k + 1)$ th and the k th
 142 eigenvalues. If the ratio is approximately equal to one, q must be large to obtain
 143 a good approximation [37].

144 **2.2. Introduction to two-level preconditioners.** Consider the linear system
 145 (1.1). As already noted, deflation techniques are typically used to shift isolated
 146 clusters of small eigenvalues to obtain a tighter spectrum and a smaller condition
 147 number. Such changes have a positive effect on the convergence of Krylov subspace
 148 methods. Consider the general (left) preconditioned system

$$149 \quad (2.3) \quad \mathcal{P}Ax = \mathcal{P}b, \quad \mathcal{P} \in \mathbb{R}^{n \times n}.$$

150 Given a projection subspace matrix $Z \in \mathbb{R}^{n \times k}$ of full rank and $k \ll n$, define the
 151 nonsingular matrix $E = Z^\top AZ \in \mathbb{R}^{k \times k}$ and the matrix $Q = ZE^{-1}Z^\top \in \mathbb{R}^{n \times n}$. The
 152 deflation preconditioner $\mathcal{P}_{\text{DEF}} \in \mathbb{R}^{n \times n}$ is defined to be [10]

$$153 \quad (2.4) \quad \mathcal{P}_{\text{DEF}} = I - AQ.$$

154 It is straightforward to show that \mathcal{P}_{DEF} is a projection matrix and $\mathcal{P}_{\text{DEF}}A$ has k zero
 155 eigenvalues (see [44] for basic properties of \mathcal{P}_{DEF}). To solve (1.1), we write

$$156 \quad x = (I - \mathcal{P}_{\text{DEF}}^\top)x + \mathcal{P}_{\text{DEF}}^\top x.$$

158 Since Q is symmetric, $\mathcal{P}_{\text{DEF}}^\top = I - QA$, and so

$$159 \quad x = QAx + \mathcal{P}_{\text{DEF}}^\top x = Qb + \mathcal{P}_{\text{DEF}}^\top x,$$

161 and we only need to compute $\mathcal{P}_{\text{DEF}}^\top x$. We first find y that satisfies the deflated system

$$162 \quad (2.5) \quad \mathcal{P}_{\text{DEF}}Ay = \mathcal{P}_{\text{DEF}}b,$$

163 then (due to the identity $A\mathcal{P}_{\text{DEF}}^\top = \mathcal{P}_{\text{DEF}}A$) we have that $\mathcal{P}_{\text{DEF}}^\top y = \mathcal{P}_{\text{DEF}}^\top x$. We therefore
 164 obtain the unique solution $x = Qb + \mathcal{P}_{\text{DEF}}^\top y$. The deflated system (2.5) is singular and
 165 can only be solved using CG if it is consistent [24], which is the case here since the
 166 same projection is applied to both sides of a consistent nonsingular system (1.1).
 167 The deflated system can also be solved using a preconditioner, giving a two-level
 168 preconditioner for the original system.

169 Tang *et al.* [44] illustrate that rounding errors can result in erratic and slow
 170 convergence of CG using \mathcal{P}_{DEF} . They thus also consider an adapted deflation
 171 preconditioner

$$172 \quad (2.6) \quad \mathcal{P}_{\text{A-DEF}} = I - QA + Q,$$

173 that combines $\mathcal{P}_{\text{DEF}}^\top$ with Q . In exact arithmetic, both \mathcal{P}_{DEF} and $\mathcal{P}_{\text{A-DEF}}$ used with
 174 CG generate the same iterates. However, numerical experiments [44] show that the
 175 latter is more robust and leads to better numerical behavior of CG¹.

176 Let $\lambda_n \geq \dots \geq \lambda_1 > 0$ be the eigenvalues of A with associated normalized
 177 eigenvectors v_n, \dots, v_1 . For the ideal deflation preconditioner, $\mathcal{P}_{\text{ideal}}$, the deflation
 178 subspace is the invariant subspace spanned by the eigenvectors associated with the
 179 smallest eigenvalues. To demonstrate how $\mathcal{P}_{\text{ideal}}$ modifies the spectrum of the deflated
 180 matrix, set $Z_k = [v_1, \dots, v_k]$ to be the $n \times k$ matrix whose columns are the eigenvectors
 181 corresponding to the smallest eigenvalues. It follows that $E = Z^\top AZ$ is equal to
 182 $\Lambda_k = \text{diag}(\lambda_1, \dots, \lambda_k)$ and the preconditioned matrix is given by

$$183 \quad \mathcal{P}_{\text{ideal}}A = A - Z_k\Lambda_k Z_k^\top.$$

185 Since Z_k is orthonormal and its columns span an invariant subspace, the spectrum
 186 of $\mathcal{P}_{\text{ideal}}A$ is $\{\lambda_n, \dots, \lambda_{k+1}, 0\}$. Starting with x_0 such that $Z_k^\top r_0 = 0$ (r_0 is the
 187 initial residual), for $l \geq 0$, $Z_k^\top (\mathcal{P}_{\text{ideal}}A)^l r_0 = 0$ and $Z_k^\top A^l r_0 = 0$. Hence the search
 188 subspace generated by the preconditioned CG (PCG) method lies in the invariant
 189 subspace spanned by v_n, \dots, v_{k+1} , which is orthogonal to the subspace spanned by
 190 the columns of Z_k . Consequently, the effective spectrum of the operator that PCG
 191 sees is $\{\lambda_n, \dots, \lambda_{k+1}\}$ and the associated *effective spectral condition number* is

$$192 \quad \kappa_{\text{eff}}(\mathcal{P}_{\text{ideal}}A) = \lambda_n / \lambda_{k+1}.$$

193 Using similar computations, the ideal adapted deflated system is given by:

$$194 \quad (2.7) \quad \mathcal{P}_{\text{A-ideal}} = A - Z_k\Lambda_k^{-1}Z_k^\top + Z_k Z_k^\top.$$

¹In [44], \mathcal{P}_{DEF} and $\mathcal{P}_{\text{A-DEF}}$ are termed $\mathcal{P}_{\text{DEF1}}$ and $\mathcal{P}_{\text{A-DEF2}}$, respectively

195 Furthermore, the spectrum of the operator that PCG sees is $\{\lambda_n, \dots, \lambda_{k+1}, 1, \dots, 1\}$
 196 and the associated effective spectral condition number is

$$197 \quad \kappa_{\text{eff}}(\mathcal{P}_{A\text{-ideal}}A) = \max\{1, \lambda_n\} / \min\{1, \lambda_{k+1}\}.$$

198 In practice, only an approximation of the ideal deflation subspace spanned by the
 199 columns of Z_k is available. Kahl and Rittich [25] analyze the deflation preconditioner
 200 using $\tilde{Z}_k \approx Z_k$ and present an upper bound on the corresponding effective spectral
 201 condition number of the deflated matrix $\kappa(\mathcal{P}A)$. Their bound [25, Proposition 4.3],
 202 which depends on $\kappa(A)$, $\kappa_{\text{eff}}(\mathcal{P}_{\text{ideal}}A)$, and the largest principal angle θ between \tilde{Z}_k
 203 and Z_k , is given by

$$204 \quad (2.8) \quad \kappa(\mathcal{P}A) \leq \left(\sqrt{\kappa(A)} \sin \theta + \sqrt{\kappa_{\text{eff}}(\mathcal{P}_{\text{ideal}}A)} \right)^2,$$

205 where $\sin \theta = \|Z_k Z_k^\top - \tilde{Z}_k \tilde{Z}_k^\top\|_2$.

206 **2.3. Schur Complement Preconditioners.** This section reviews the Schur
 207 complement preconditioner with a focus on two-level variants that were introduced in
 208 [14, 27].

209 One-level preconditioners may not provide the required robustness when used with
 210 a Krylov subspace method because they typically fail to capture information about
 211 the eigenvectors corresponding to the smallest eigenvalues. To try and remedy this, in
 212 their (unpublished) report, Grigori *et al.* [14] and, independently, Li *et al.* [27] propose
 213 a two-level preconditioner based on using a block factorization and approximating the
 214 resulting Schur complement.

215 Applying graph partitioning techniques (for example, using the METIS package
 216 [26, 29]), A can be symmetrically permuted to the 2×2 doubly bordered block diagonal
 217 form

$$218 \quad (2.9) \quad P^\top A P = \begin{pmatrix} A_I & A_{I\Gamma} \\ A_{\Gamma I} & A_\Gamma \end{pmatrix},$$

219 where $A_I \in \mathbb{R}^{n_I \times n_I}$ is a block diagonal matrix, $A_\Gamma \in \mathbb{R}^{n_\Gamma \times n_\Gamma}$, $A_{\Gamma I} \in \mathbb{R}^{n_\Gamma \times n_I}$ and
 220 $A_{I\Gamma} = A_{\Gamma I}^\top$. For simplicity of notation, we assume that A is of the form (2.9) (and
 221 omit the permutation P from the subsequent discussion).

222 The block form (2.9) induces a block LDLT factorization

$$223 \quad (2.10) \quad A = \begin{pmatrix} I & \\ A_{\Gamma I} A_I^{-1} & I \end{pmatrix} \begin{pmatrix} A_I & \\ & S_\Gamma \end{pmatrix} \begin{pmatrix} I & A_I^{-1} A_{I\Gamma} \\ & I \end{pmatrix},$$

224 where

$$225 \quad (2.11) \quad S_\Gamma = A_\Gamma - A_{\Gamma I} A_I^{-1} A_{I\Gamma}$$

226 is the Schur complement of A with respect to A_I . Provided the blocks within A_I
 227 are small, they can be factorized cheaply in parallel using a direct algorithm (see,
 228 for example, [38]) and thus we assume that solving linear systems with A_I is not
 229 computationally expensive. However, the SPD Schur complement S_Γ is typically
 230 large and significantly denser than A_Γ (its size increases with the number of blocks
 231 in A_I) and, in large-scale practical applications, it may not be possible to explicitly
 232 assemble or factorize it.

233 Preconditioners may be derived by approximating S_Γ^{-1} . An approximate block
234 factorization of A^{-1} is

$$235 \quad M^{-1} = \begin{pmatrix} I & -A_I^{-1}A_{I\Gamma I} \\ & I \end{pmatrix} \begin{pmatrix} A_I^{-1} & \\ & \tilde{S}^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -A_{\Gamma I}A_I^{-1} & I \end{pmatrix},$$

237 where $\tilde{S}^{-1} \approx S_\Gamma^{-1}$. If M^{-1} is employed as a preconditioner for A then the
238 preconditioned system is given by

$$239 \quad (2.12) \quad M^{-1}A = \begin{pmatrix} I & A_I^{-1}A_{I\Gamma}(I - \tilde{S}^{-1}S_\Gamma) \\ & \tilde{S}^{-1}S_\Gamma \end{pmatrix},$$

240 with $\Lambda(M^{-1}A) = \{\lambda \in \Lambda(\tilde{S}^{-1}S_\Gamma)\} \cup \{1\}$. Thus, to bound the condition number
241 $\kappa(M^{-1}A)$, we need to construct \tilde{S}^{-1} so that $\kappa(\tilde{S}^{-1}S_\Gamma)$ is bounded. Moreover, (2.12)
242 shows that applying the preconditioner requires the efficient solution of linear systems
243 with $\tilde{S}^{-1}S_\Gamma$ and A_I , the latter being relatively inexpensive. We therefore focus on
244 constructing preconditioners \tilde{S}^{-1} for linear systems of the form

$$245 \quad (2.13) \quad S_\Gamma w = f.$$

246 Consider the first-level preconditioner obtained by setting

$$247 \quad (2.14) \quad \tilde{S}_1^{-1} := A_\Gamma^{-1}.$$

248 Assume for now that we can factorize A_Γ , although in practice it may be very large
249 and a recursive construction of the preconditioner may then be needed (see [49]). Let
250 the eigenvalues of the generalized eigenvalue problem

$$251 \quad (2.15) \quad S_\Gamma z = \lambda \tilde{S}_1^{-1} z$$

be $\lambda_{n_\Gamma} \geq \dots \geq \lambda_1 > 0$. From (2.11), $\lambda_{n_\Gamma} \leq 1$ and so

$$\kappa(\tilde{S}_1^{-1}S_\Gamma) = \frac{\lambda_{n_\Gamma}}{\lambda_1} \leq \frac{1}{\lambda_1}.$$

252 As this is unbounded as λ_1 approaches zero, we seek to add a low rank term to
253 “correct” the approximation and shift the smallest k eigenvalues of $\tilde{S}_1^{-1}S_\Gamma$. Let
254 $\Lambda_k = \text{diag}\{\lambda_1, \dots, \lambda_k\}$ and let $Z_k \in \mathbb{R}^{n_\Gamma \times k}$ be the matrix whose columns are
255 the corresponding eigenvectors. Without loss of generality, we assume Z_k is A_Γ -
256 orthonormal. Let the Cholesky factorization of A_Γ be

$$257 \quad (2.16) \quad A_\Gamma = R_\Gamma^\top R_\Gamma$$

258 and define

$$259 \quad (2.17) \quad \tilde{S}_2^{-1} := A_\Gamma^{-1} + Z_k(\Lambda_k^{-1} - I)Z_k^\top.$$

260 \tilde{S}_2^{-1} is an additive combination of the first-level preconditioner \tilde{S}_1^{-1} and an adapted
261 deflation preconditioner associated with the subspace spanned by the columns of
262 $U_k = R_\Gamma Z_k$, which is an invariant subspace of $R_\Gamma^{-1}S_\Gamma R_\Gamma^{-\top}$. Substituting U_k into
263 (2.17) and using (2.16),

$$264 \quad (2.18) \quad \tilde{S}_2^{-1} = R_\Gamma^{-1}(I + U_k(\Lambda_k^{-1} - I)U_k^\top)R_\Gamma^{-\top}.$$

265 Setting $Q = U_k \Lambda_k^{-1} U_k^\top$ in (2.6) gives

$$266 \quad \mathcal{P}_{\text{A-DEF}} = R_\Gamma \tilde{S}_2^{-1} R_\Gamma^\top.$$

Now $\tilde{S}_2^{-1} S_\Gamma = R_\Gamma^{-1} \mathcal{P}_{\text{A-DEF}} R_\Gamma^{-\top} S_\Gamma$ and $\mathcal{P}_{\text{A-DEF}} R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ are spectrally equivalent and $\Lambda(\tilde{S}_2^{-1} S_\Gamma) = \{\lambda_{n_\Gamma}, \lambda_{n_\Gamma-1}, \dots, \lambda_{k+1}\} \cup \{1\}$. It follows that

$$\kappa(\tilde{S}_2^{-1} S_\Gamma) = \frac{\lambda_{n_\Gamma}}{\lambda_{k+1}} \leq \frac{1}{\lambda_{k+1}}.$$

268 Grigori *et al.* [14] note that (2.15) is equivalent to the generalized eigenvalue
269 problem

$$270 \quad (2.19) \quad (A_\Gamma - S_\Gamma)z = A_{\Gamma I} A_I^{-1} A_{I\Gamma} z = \sigma A_\Gamma z, \quad \sigma = 1 - \lambda.$$

271 Setting $u = R_\Gamma z$ and defining

$$272 \quad (2.20) \quad H = R_\Gamma^{-\top} A_{\Gamma I} A_I^{-1} A_{I\Gamma} R_\Gamma^{-1},$$

273 (2.19) becomes

$$274 \quad (2.21) \quad Hu = \sigma u.$$

275 Thus, the smallest eigenvalues λ of (2.15) are transformed to the largest eigenvalues
276 σ of problems (2.19) and (2.21). Grigori *et al.* employ a randomized algorithm to
277 compute a low rank eigenvalue decomposition (EVD) of H that approximates its
278 largest eigenvalues and vectors, which are multiplied by R_Γ^{-1} to obtain approximate
279 eigenvectors of $A_\Gamma^{-1} S_\Gamma$.

280 In [27], Li *et al.* write the inverse of the Schur complement S_Γ as:

$$\begin{aligned} 281 \quad (2.22) \quad S_\Gamma^{-1} &= (A_\Gamma - A_{\Gamma I} A_I^{-1} A_{I\Gamma})^{-1} \\ &= (R_\Gamma^\top R_\Gamma - A_{\Gamma I} A_I^{-1} A_{I\Gamma})^{-1} \\ 282 \quad &= R_\Gamma^{-1} (I - H)^{-1} R_\Gamma^{-\top}, \end{aligned}$$

where the symmetric positive semidefinite (SPSD) matrix H is given by (2.20). Since
 $I - H = R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ is SPD, the eigenvalues $\sigma_1 \geq \dots \geq \sigma_{n_\Gamma}$ of H belong to $[0, 1]$.
Let the EVD of H be

$$H = U \Sigma U^\top,$$

283 where U is orthonormal and $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_{n_\Gamma}\}$. It follows that

$$\begin{aligned} 284 \quad (2.23) \quad S_\Gamma^{-1} &= R_\Gamma^{-1} (I - U \Sigma U^\top)^{-1} R_\Gamma^{-\top} \\ &= R_\Gamma^{-1} U (I - \Sigma)^{-1} U^\top R_\Gamma^{-\top} \\ &= R_\Gamma^{-1} \left(I + U \left((I - \Sigma)^{-1} - I \right) U^\top \right) R_\Gamma^{-\top} \\ 285 \quad &= A_\Gamma^{-1} + R_\Gamma^{-1} U \left((I - \Sigma)^{-1} - I \right) U^\top R_\Gamma^{-\top}. \end{aligned}$$

286 If H has an approximate EVD of the form

$$287 \quad H \approx U \tilde{\Sigma} U^\top, \quad \tilde{\Sigma} = \text{diag}\{\tilde{\sigma}_1, \dots, \tilde{\sigma}_{n_\Gamma}\},$$

289 then an approximation of S_Γ^{-1} is

$$290 \quad (2.24) \quad \tilde{S}^{-1} = A_\Gamma^{-1} + R_\Gamma^{-1}U \left((I - \tilde{\Sigma})^{-1} - I \right) U^\top R_\Gamma^{-\top}.$$

291

292 The simplest selection of $\tilde{\Sigma}$ is the one that ensures the k largest eigenvalues of $(I - \tilde{\Sigma})^{-1}$
 293 match the largest eigenvalues of $(I - \Sigma)^{-1}$. Li *et al.* set $\tilde{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_k, \theta, \dots, \theta)$,
 294 where $\theta \in [0, 1]$. The resulting preconditioner can be written as

$$295 \quad (2.25) \quad \tilde{S}_\theta^{-1} = \frac{1}{1 - \theta} A_\Gamma^{-1} + Z_k \left((I - \Sigma_k)^{-1} - \frac{1}{1 - \theta} I \right) Z_k^\top,$$

296 where $\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k)$ and the columns of $Z_k = R_\Gamma^{-1}U_k$ are the eigenvectors
 297 corresponding to the k largest eigenvalues of H . In [27], it is shown that $\kappa(\tilde{S}_\theta^{-1}S) =$
 298 $(1 - \sigma_{n_\Gamma})/(1 - \theta)$, which takes its minimum value for $\theta = \sigma_{k+1}$.

299 In the next section, we analyse the eigenvalue problems that need to be solved
 300 to construct the preconditioners (2.17) and (2.25). In particular, we show that the
 301 approaches presented in [14, 27] for tackling these problems are inefficient because of
 302 the eigenvalue distribution.

303 3. Analysis of $Hu = \sigma u$.

304 **3.1. Use of the Lanczos method.** Consider the eigenproblem:

$$305 \quad \text{Given } \varepsilon > 0, \text{ find all the eigenpairs } (\lambda, z) \in \mathbb{R} \times \mathbb{R}^{n_\Gamma} \text{ such that}$$

$$306 \quad S_\Gamma z = \lambda A_\Gamma z, \quad \lambda < \varepsilon.$$

307 This can be rewritten as:

$$308 \quad (3.1) \quad \text{Given } \varepsilon > 0, \text{ find all the eigenpairs } (\lambda, z) \in \mathbb{R} \times \mathbb{R}^{n_\Gamma} \text{ such that}$$

$$309 \quad (I - H)u = \lambda u, \quad z = R_\Gamma^{-1}u, \quad \lambda < \varepsilon,$$

310 where R_Γ and H are given by (2.16) and (2.20). Consider also the eigenproblem:

$$311 \quad (3.2) \quad \text{Given } \varepsilon > 0, \text{ find all the eigenpairs } (\sigma, u) \in \mathbb{R} \times \mathbb{R}^{n_\Gamma} \text{ such that}$$

$$312 \quad Hu = \sigma u, \quad \sigma > 1 - \varepsilon.$$

As already observed, each eigenpair (λ, z) of (3.1) corresponds to the eigenpair $(1 - \lambda, R_\Gamma z)$ of (3.2). Consider using the Lanczos method to solve these eigenproblems. The Krylov subspace at iteration j generated for (3.1) is

$$K_j((I - H), v_1) = \text{span}(v_1, (I - H)v_1, \dots, (I - H)^{j-1}v_1),$$

while the subspace generated for (3.2) is

$$K_j(H, v_1) = \text{span}(v_1, Hv_1, \dots, H^{j-1}v_1).$$

It is clear that, provided the same starting vector v_1 is used, $K_j((I - H), v_1)$ and $K_j(H, v_1)$ are identical. Suppose that $[\mathcal{V}_j, v_{j+1}]$ is the output of the Lanczos basis of the Krylov subspace, then the subspace relations that hold at iteration j are

$$(I - H)\mathcal{V}_j = \mathcal{V}_j T_j + v_{j+1} h_j^\top,$$

$$H\mathcal{V}_j = \mathcal{V}_j(I - T_j) - v_{j+1}h_j^\top,$$

313 where $T_j \in \mathbb{R}^{j \times j}$ is a symmetric tridiagonal matrix and $h_j \in \mathbb{R}^j$. The eigenpair
 314 (λ, z) (respectively, (σ, u)) corresponding to the smallest (respectively, largest)
 315 eigenvalue in (3.1) (respectively, (3.2)) is approximated by the eigenpair $(\tilde{\lambda}, R_\Gamma^{-1}\mathcal{V}_j\tilde{u})$
 316 (respectively, $(\tilde{\sigma}, \mathcal{V}_j\tilde{u})$) corresponding to the smallest (respectively, largest) eigenvalue
 317 of T_j (respectively, $I - T_j$). To overcome memory constraints, the Lanczos procedure
 318 is typically restarted after a chosen number of iterations, at each restart discarding
 319 the non convergent part of the Krylov subspace [42]. Hence, starting with the same
 320 v_1 and performing the same number of iterations per cycle, in exact arithmetic the
 321 accuracy obtained when solving (3.1) and (3.2) is identical.

322 Having shown that the convergence of Lanczos' method for solving (3.1) and (3.2)
 is the same, we focus on (3.2). In Figure 1, for each of our test matrices in Table 1

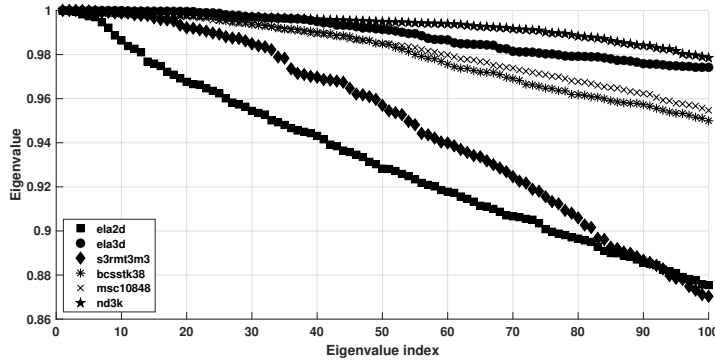


FIG. 1. Largest 100 eigenvalues of $H = R_\Gamma^{-\top} A_{\Gamma I} A_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$ associated with our test matrices computed to an accuracy of 10^{-8} using the Krylov-Schur method [42].

323 we plot the 100 largest eigenvalues of the matrix H given by (2.20). We see that the
 324 largest eigenvalues (which are the ones that we require) are clustered near one and
 325 they do not decay rapidly. As there are a significant number of eigenvalues in the
 326 cluster, computing the largest k (for $k = O(10)$) and the corresponding eigenvectors
 327 with sufficient accuracy using the Lanczos method is challenging. Similar distributions
 328 were observed for the larger test set that we report on in the Appendix, particularly
 329 for problems for which the one-level preconditioner \tilde{S}_1 was found to perform poorly,
 330 which is generally the case when $\kappa(A)$ is large. Table 2 reports the Lanczos iteration
 331 counts (it_{Lan}) for computing the $k = 20$ and 40 largest eigenpairs (that is, the number
 332 of linear systems that are solved in the Lanczos method). In addition, we present the
 333 PCG iteration count (it_{PCG}) for solving the linear system (2.13) using the first-level
 334 preconditioner $\tilde{S}_1 = A_\Gamma^{-1}$ and the two-level preconditioner \tilde{S}_2 given by (2.17). We
 335 see that, in terms of the total iteration count, the first-level preconditioner is the
 336 more efficient option. It is of interest to consider whether relaxing the convergence
 337 tolerance ε_{Lan} in the Lanczos method can reduce the total iteration count for \tilde{S}_2 .
 338 Table 3 illustrates the effect of varying ε_{Lan} for problem el3d (results for the other test
 339 problems are consistent). Although it_{Lan} decreases as ε_{Lan} increases, it_{PCG} increases
 340 and the total count still exceeds the 175 PCG iterations required by the first-level
 341 preconditioner \tilde{S}_1 .
 342

343 As already observed, in [49] a recursive (multilevel) scheme is proposed to
 344 help mitigate the computational costs of building and applying the preconditioner.

Identifier	\tilde{S}_1	\tilde{S}_2					
		$k = 20$			$k = 40$		
		it_{PCG}	it_{Lan}	it_{PCG}	total	it_{Lan}	it_{PCG}
bcsstk38	584	797	122	919	730	67	797
el2d	914	1210	231	1441	982	120	1102
el3d	174	311	37	348	389	27	416
msc10848	612	813	116	929	760	63	823
nd3k	603	1796	143	1939	1349	105	1454
s3rmt3m3	441	529	70	599	480	37	517

TABLE 2

The Lanczos iteration count (it_{Lan}) and the iteration count for PCG (it_{PCG}). The convergence tolerance for the Lanczos method and PCG is 10^{-6} . The size of the Krylov subspace per cycle is $2k$.

ε_{Lan}	$k = 20$			$k = 40$		
	it_{Lan}	it_{PCG}	total	it_{Lan}	it_{PCG}	total
0.1	50	131	181	80	101	181
0.08	50	131	181	100	85	185
0.06	60	121	181	100	85	185
0.04	82	100	182	120	71	191
0.02	127	64	201	207	37	244
0.01	169	41	210	259	32	291
0.005	213	38	251	316	29	345
0.001	247	37	284	372	28	400

TABLE 3

Problem *el3d* and two-level preconditioner \tilde{S}_2 : sensitivity of the number of the Lanczos iteration count (it_{Lan}) and the iteration count for PCG (it_{PCG}) to the convergence tolerance ε_{Lan} . The PCG convergence tolerance is 10^{-6} . The size of the Krylov subspace per cycle is $2k$.

345 Nevertheless, the Lanczos method is still used, albeit with reduced costs for applying
346 the operator matrices.

347 **3.2. Use of Nyström’s method.** As suggested in [14], an alternative approach
348 to approximating the dominant subspace of H is to use a randomized method,
349 specifically a randomized eigenvalue decomposition. Because H is SPSD, Nyström’s
350 method can be use. Results are presented in Table 4 for problem *el3d* (results for our
351 other test examples are consistent with these). Here p is the oversampling parameter
352 and q is the power iteration parameter. These show that, as with the Lanczos method,
353 Nyström’s method struggles to approximate the dominant eigenpairs of H . Using
354 $k = 20$ (respectively, 40) exact eigenpairs, PCG using \tilde{S}_2 requires 37 (respectively,
355 28) iterations. To obtain the same iteration counts using vectors computed using
356 Nyström’s method requires the oversampling parameter to be greater than 2000,
357 which is clearly prohibitive. Using the power iteration improves the quality of the
358 approximate subspace. However, the large value of q needed to decrease the PCG
359 iteration count means a large number of linear systems must be solved with A_Γ , in
360 addition to the work involved in the orthogonalization that is needed between the
361 power iterations to maintain stability. Indeed, it is sufficient to look at Figure 1 to
362 predict this behaviour for any randomized method applied to H . The lack of success
363 of existing strategies motivates us, in the next section, to reformulate the eigenvalue

364 problem to one with a spectrum that is easy to approximate.

p	$k = 20$	$k = 40$	q	$k = 20$	$k = 40$
100	171	169	0	172	171
200	170	165	20	121	87
400	165	161	40	86	48
800	155	146	60	68	34
1600	125	111	80	55	30
3200	55	45	100	46	29

TABLE 4

PCG iteration counts for problem *el3d* using the two-level preconditioner \tilde{S}_2 constructed using a rank k approximation of $H = R_\Gamma^{-\top} A_{\Gamma I} A_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$. The PCG convergence tolerance is 10^{-6} . Nyström's method applied to H with the oversampling parameter $p \geq 100$ and the power iteration parameter $q = 0$ (left) and with $p = 0$ and $q \geq 0$ (right).

365 **4. Nyström–Schur two-level preconditioner.** In this section, we propose
 366 reformulating the eigenvalue problem to obtain a new one such that the desired
 367 eigenvectors correspond to the largest eigenvalues and these eigenvalues are well
 368 separated from the remaining eigenvalues: this is what is needed for randomized
 369 methods to be successful.

370 **4.1. Two-level preconditioner for S_Γ .** Applying the Sherman Morrison
 371 Woodbury identity [13, 2.1.3], the inverse of the Schur complement S_Γ (2.11) can
 372 be written as:

$$373 \quad (4.1) \quad \begin{aligned} S_\Gamma^{-1} &= A_\Gamma^{-1} + A_\Gamma^{-1} A_{\Gamma I} (A_I - A_{I\Gamma} A_\Gamma^{-1} A_{\Gamma I})^{-1} A_{I\Gamma} A_\Gamma^{-1} \\ 374 \quad &= A_\Gamma^{-1} + A_\Gamma^{-1} A_{\Gamma I} S_I^{-1} A_{I\Gamma} A_\Gamma^{-1}, \end{aligned}$$

375 where

$$376 \quad (4.2) \quad S_I = A_I - A_{I\Gamma} A_\Gamma^{-1} A_{\Gamma I}$$

377 is the Schur complement of A with respect to A_I . Using the Cholesky factorization
 378 (2.16), we have

$$379 \quad (4.3) \quad R_\Gamma S_\Gamma^{-1} R_\Gamma^\top = I + R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1}.$$

380 Note that if (λ, u) is an eigenpair of $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$, then $(\frac{1}{\lambda} - 1, u)$ is an eigenpair of
 381 $R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$. Therefore, the cluster of eigenvalues of $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ near the
 382 origin (which correspond to the cluster of eigenvalues of H near 1) correspond to
 383 very large and highly separated eigenvalues of $R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$. Hence, using
 384 randomized methods to approximate the dominant subspace of $R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$
 385 can be an efficient way of computing a deflation subspace for $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$. Now
 386 assume that we have a low rank approximation

$$387 \quad (4.4) \quad R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1} \approx \check{U}_k \check{\Sigma}_k \check{U}_k^\top,$$

388 where $\check{U}_k \in \mathbb{R}^{n_\Gamma \times k}$ is orthonormal and $\check{\Sigma}_k \in \mathbb{R}^{k \times k}$ is diagonal. Combining (4.3) and
 389 (4.4), we can define a preconditioner for $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ to be

$$390 \quad (4.5) \quad \mathcal{P}_1 = I + \check{U}_k \check{\Sigma}_k \check{U}_k^\top.$$

391 The preconditioned matrix $\mathcal{P}_1 R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ is spectrally equivalent to $R_\Gamma^{-1} \mathcal{P}_1 R_\Gamma^{-\top} S_\Gamma$.
 392 Therefore, the preconditioned system can be written as

$$393 \quad (4.6) \quad \mathcal{M}_1 S_\Gamma = R_\Gamma^{-1} \mathcal{P}_1 R_\Gamma^{-\top} S_\Gamma = \left(A_\Gamma^{-1} + \check{Z}_k \check{\Sigma}_k \check{Z}_k^\top \right) S_\Gamma,$$

394 where $\check{Z}_k = R_\Gamma^{-1} \check{U}_k$. If (4.4) is obtained using a truncated EVD denoted by $U_k \Sigma_k U_k^\top$,
 395 then $\check{U}_k = U_k$ and the subspace spanned by the columns of U_k is an invariant subspace
 396 of $R_\Gamma S_\Gamma^{-1} R_\Gamma^\top$ and of its inverse $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$. Furthermore, using the truncated EVD,
 397 (4.5) is an adapted deflation preconditioner for $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$. Indeed, as the columns of
 398 U_k are orthonormal eigenvectors, we have from (4.3) that $R_\Gamma S_\Gamma^{-1} R_\Gamma^\top U_k = U_k (I + \Sigma_k)$.
 399 Hence $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} U_k = U_k (I + \Sigma_k)^{-1}$ and the preconditioned matrix is

$$\begin{aligned} 400 \quad \mathcal{P}_{\text{A-DEF}} R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} &= R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} + U_k \Sigma_k (I + \Sigma_k)^{-1} U_k^\top \\ 401 &= R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} + U_k ((I + \Sigma_k) - I) (I + \Sigma_k)^{-1} U_k^\top \\ 402 &= R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} - U_k (I + \Sigma_k)^{-1} U_k^\top + U_k U_k^\top, \end{aligned}$$

404 which has the same form as the ideal adapted preconditioned matrix (2.7).

405 Note that given the matrix \check{U}_k in the approximation (4.4), then following
 406 subsection 2.2, we can define a deflation preconditioner for $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$. Setting
 407 $E_k = \check{U}_k^\top R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} \check{U}_k$ and $Q = \check{U}_k E^{-1} \check{U}_k^\top$, the deflation preconditioner is

$$408 \quad (4.7) \quad \mathcal{P}_{1\text{-A-DEF}} = I - Q R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1} + Q.$$

409 The preconditioned Schur complement $\mathcal{P}_{1\text{-A-DEF}} R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ is spectrally similar to
 410 $R_\Gamma^{-1} \mathcal{P}_{1\text{-A-DEF}} R_\Gamma^{-\top} S_\Gamma$ and thus

$$411 \quad (4.8) \quad \mathcal{M}_{1\text{-A-DEF}} = R_\Gamma^{-1} \mathcal{P}_{1\text{-A-DEF}} R_\Gamma^{-\top}$$

412 is a two-level preconditioner for S_Γ .

413 **4.2. Lanczos versus Nyström.** The two-level preconditioner (4.8) relies on
 414 computing a low-rank approximation (4.4). We now consider the difference between
 415 using the Lanczos and Nyström methods for this.

416 Both methods require the application of $R_\Gamma^{-\top} A_{\Gamma I} S_I^{-1} A_{I\Gamma} R_\Gamma^{-1}$ to a set of $k + p$
 417 vectors, where $k > 0$ is the required rank and $p \geq 0$. Because explicitly computing
 418 the SPD matrix $S_I = A_I - A_{I\Gamma} A_\Gamma^{-1} A_{\Gamma I}$ and factorizing it is prohibitively expensive,
 419 applying S_I^{-1} must be done using an iterative solver.

420 The Lanczos method builds a Krylov subspace of dimension $k + p$ in order to
 421 compute a low-rank approximation. Therefore, $k + p$ linear systems must be solved,
 422 each with one right-hand side, first for R_Γ , then for S_I , and then for R_Γ^\top . However,
 423 the Nyström method requires the solution of only one linear system with $k + p$ right-
 424 hand sides for R_Γ , then for S_I , and then for R_Γ^\top . This allows the use of matrix-matrix
 425 operations rather than less efficient matrix-vector operations. Moreover, as we will
 426 illustrate in section 5, block Krylov subspace methods, such as block CG [35], for
 427 solving the system with S_I yield faster convergence than their classical counterparts.
 428 When the Nyström method is used, we call the resulting preconditioner (4.8) the
 429 *Nyström-Schur preconditioner*.

430 **4.3. Avoiding computations with R_Γ .** For large scale problems, computing
 431 the Cholesky factorization $A_\Gamma = R_\Gamma^\top R_\Gamma$ is prohibitive and so we would like to avoid

432 computations with R_Γ . We can achieve this by using an iterative solver to solve linear
 433 systems with A_Γ . Note that this is possible when solving the generalized eigenvalue
 434 problem (2.15). Because A_Γ is typically well conditioned, so too is R_Γ . Thus, we can
 435 reduce the cost of computing the Nyström–Schur preconditioner by approximating
 436 the SPSD matrix $A_{\Gamma I} S_I^{-1} A_{I\Gamma}$ (or even by approximating S_I^{-1}). Of course, this needs
 437 to be done without seriously adversely affecting the preconditioner quality. Using an
 438 approximate factorization

$$439 \quad (4.9) \quad A_{\Gamma I} S_I^{-1} A_{I\Gamma} \approx \widetilde{W}_k \widetilde{\Sigma}_k \widetilde{W}_k^\top,$$

440 an alternative deflation preconditioner is

$$441 \quad \mathcal{P}_2 = I + R_\Gamma^{-\top} \widetilde{W}_k \widetilde{\Sigma}_k \widetilde{W}_k^\top R_\Gamma^{-1},$$

$$442 \quad = R_\Gamma^{-\top} \left(A_\Gamma + \widetilde{W}_k \widetilde{\Sigma}_k \widetilde{W}_k^\top \right) R_\Gamma^{-1}.$$

444 The preconditioned Schur complement $\mathcal{P}_2 R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ is spectrally similar to
 445 $R_\Gamma^{-1} \mathcal{P}_2 R_\Gamma^{-\top} S_\Gamma$ and, setting $\widetilde{Z}_k = A_\Gamma^{-1} \widetilde{W}_k$, we have

$$446 \quad (4.10) \quad \mathcal{M}_2 S_\Gamma = R_\Gamma^{-1} \mathcal{P}_2 R_\Gamma^{-\top} S_\Gamma = (A_\Gamma^{-1} + \widetilde{Z}_k \widetilde{\Sigma}_k \widetilde{Z}_k^\top) S_\Gamma.$$

447 Thus $\mathcal{M}_2 = A_\Gamma^{-1} + \widetilde{Z}_k \widetilde{\Sigma}_k \widetilde{Z}_k^\top$ is a variant of the Nyström–Schur preconditioner for S_Γ
 448 that avoids computing R_Γ .

449 Alternatively, assuming we have an approximate factorization

$$450 \quad (4.11) \quad S_I^{-1} \approx \widehat{V}_k \widehat{\Sigma}_k \widehat{V}_k^\top,$$

yields

$$\mathcal{P}_3 = I + R_\Gamma^{-\top} A_{\Gamma I} \widehat{V}_k \widehat{\Sigma}_k \widehat{V}_k^\top A_{I\Gamma} R_\Gamma^{-1}.$$

451 Again, $\mathcal{P}_3 R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ is spectrally similar to $R_\Gamma^{-1} \mathcal{P}_3 R_\Gamma^{-\top} S_\Gamma$ and, setting $\widehat{Z}_k =$
 452 $A_\Gamma^{-1} A_{\Gamma I} \widehat{V}_k$, we have

$$453 \quad (4.12) \quad \mathcal{M}_3 S_\Gamma = R_\Gamma^{-1} \mathcal{P}_3 R_\Gamma^{-\top} S_\Gamma = (A_\Gamma^{-1} + \widehat{Z}_k \widehat{\Sigma}_k \widehat{Z}_k^\top) S_\Gamma,$$

454 which gives another variant of the Nyström–Schur preconditioner. In a similar way
 455 to defining $\mathcal{M}_{1\text{-A-DEF}}$ (4.7), we can define $\mathcal{M}_{2\text{-A-DEF}}$ and $\mathcal{M}_{3\text{-A-DEF}}$. Note that $\mathcal{M}_{2\text{-A-DEF}}$
 456 and $\mathcal{M}_{3\text{-A-DEF}}$ also avoid computations with R_Γ .

457 **4.4. Nyström–Schur preconditioner.** Algorithm 4.1 presents the
 458 construction of the Nyström–Schur preconditioner \mathcal{M}_2 ; an analogous derivation
 459 yields the variant \mathcal{M}_3 . Step 3 is the most expensive step, that is, solving the $n_I \times n_I$
 460 SPD linear system

$$461 \quad (4.13) \quad S_I X = F,$$

462 where $F \in \mathbb{R}^{n_I \times (k+p)}$ and $S_I = A_I - A_{I\Gamma} A_\Gamma^{-1} A_{\Gamma I}$. Using an iterative solver requires a
 463 linear system solve with A_Γ on each iteration. Importantly for efficiency, the number
 464 of iterations can be limited by employing a large relative tolerance when solving
 465 (4.13) without adversely affecting the performance of the resulting preconditioner.
 466 Numerical experiments in section 5 illustrate this robustness.

467 Observe that applying \mathcal{M}_2 to a vector requires the solution of a linear system
 468 with A_Γ and a low rank correction; see Step 12.

Algorithm 4.1 Construction of the Nyström–Schur preconditioner (4.10)

Input: A in block form (2.9), $k > 0$ and $p \geq 0$ ($k, p \ll n_\Gamma$) and $\varepsilon > 0$.

Output: Two-level preconditioner for the $n_\Gamma \times n_\Gamma$ Schur complement S_Γ .

- 1: Draw a random matrix $G \in \mathbb{R}^{n_\Gamma \times (k+p)}$.
 - 2: Compute $F = A_\Gamma G$.
 - 3: Solve $S_\Gamma X = F$.
 - 4: Compute $Y = A_\Gamma X$.
 - 5: Compute $Y = QR$.
 - 6: Set $C = G^\top Y$.
 - 7: Compute the EVD $C = V_1 D_1 V_1^\top + V_2 D_2 V_2^\top$, where D_1 contains all the eigenvalues that are at least ε .
 - 8: Set $T = R V_1 D_1^{-1} V_1^\top R^\top$.
 - 9: Compute the EVD $T = W E W^\top$.
 - 10: Set $\tilde{U} = YW(:, 1:k)$, $\Sigma = E(1:k, 1:k)$.
 - 11: Solve $A_\Gamma Z = \tilde{U}$.
 - 12: Define the preconditioner $\mathcal{M}_2 = A_\Gamma^{-1} + Z \Sigma Z^\top$.
-

469 **4.5. Estimation of the Spectral Condition Number.** In this section, we
 470 provide an expectation of the spectral condition number of S_Γ preconditioned by
 471 the Nyström–Schur preconditioner. Saibaba [37] derives bounds on the angles
 472 between the approximate singular vectors computed using a randomized singular
 473 value decomposition and the exact singular vectors of a matrix. It is straightforward
 474 to derive the corresponding bounds for the Nyström method. Let Π_M denote
 475 the orthogonal projector on the space spanned by the columns of the matrix M .
 476 Let (λ_j, u_j) , $j = 1, \dots, k$, be the dominant eigenpairs of $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$. Following
 477 the notation in Algorithm 2.1, the angle $\theta_j = \angle(u_j, \tilde{U})$ between the approximate
 478 eigenvectors $\tilde{U} \in \mathbb{R}^{n_\Gamma \times (k+p)}$ of $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ and the exact eigenvector $u_j \in \mathbb{R}^{n_\Gamma}$
 479 satisfies

$$480 \quad (4.14) \quad \sin \angle(u_j, \tilde{U}) = \|u_j - \Pi_{\tilde{U}} u_j\|_2 \leq \gamma_{j,k}^{q+1} c,$$

481 where q is the power iteration count (recall (2.2)), $\gamma_{j,k}$ is the gap between $\lambda_j^{-1} - 1$
 482 and $\lambda_{k+1}^{-1} - 1$ given by

$$483 \quad (4.15) \quad \gamma_{j,k} = (\lambda_{k+1}^{-1} - 1) / (\lambda_j^{-1} - 1),$$

484 and c has the expected value

$$485 \quad (4.16) \quad \mathbb{E}(c) = \sqrt{\frac{k}{p-1} + \frac{e\sqrt{(k+p)(n_\Gamma - k)}}{p}},$$

486 where k is the required rank and $p \geq 2$ is the oversampling parameter. Hence,

$$487 \quad (4.17) \quad \mathbb{E}(\sin \angle(u_j, \tilde{U})) = \mathbb{E}(\|u_j - \Pi_{\tilde{U}} u_j\|_2) \leq \gamma_{j,k}^{q+1} \mathbb{E}(c).$$

488 Note that if $\lambda_j \leq 1/2$ then $\gamma_{j,k} \leq 2\lambda_j / \lambda_{k+1}$ ($j = 1, \dots, k$).

PROPOSITION 4.1. *Let the EVD of the SPD matrix $I - H = R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ be*

$$\begin{bmatrix} U_\perp & U_k \end{bmatrix} \begin{bmatrix} \Lambda_\perp & \\ & \Lambda_k \end{bmatrix} \begin{bmatrix} U_\perp^\top \\ U_k^\top \end{bmatrix},$$

489 where $\Lambda_\perp \in \mathbb{R}^{(n_\Gamma-k) \times (n_\Gamma-k)}$ and $\Lambda_k \in \mathbb{R}^{k \times k}$ are diagonal matrices with the eigenvalues
 490 $(\lambda_i)_{k \geq i \geq 1}$ and $(\lambda_i)_{n_\Gamma \geq i \geq k+1}$, respectively, in decreasing order. Furthermore, assume
 491 that $\lambda_k \leq 1/2$. Let the columns of $\tilde{U} \in \mathbb{R}^{n_\Gamma \times (k+p)}$ be the approximate eigenvectors of
 492 $I - H$ computed using the Nystrom method and let

$$493 \quad \mathcal{P} = I - (I - H)\tilde{U}E^{-1}\tilde{U}^\top \quad \text{with} \quad E = \tilde{U}^\top(I - H)\tilde{U},$$

494 be the associated deflation preconditioner. Then, the effective condition number of the
 495 two-level preconditioner $\mathcal{P}(I - H) = \mathcal{P}R_\Gamma^{-\top}S_\Gamma R_\Gamma^{-1}$ satisfies

$$496 \quad (4.18) \quad \mathbb{E} \left(\sqrt{\kappa_{\text{eff}}(\mathcal{P}(I - H))} \right) \leq c_1 \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{k+1}}},$$

497 where c_1^2 is independent of the spectrum of $I - H$ and can be bounded by a polynomial
 498 of degree 3 in k .

499 *Proof.* Let $x \in \mathbb{R}^{n_\Gamma}$. Since u_1, \dots, u_{n_Γ} form an orthogonal basis of \mathbb{R}^{n_Γ} , there
 500 exists $\alpha_1, \dots, \alpha_{n_\Gamma} \in \mathbb{R}$ such that $x = \sum_{i=1}^{n_\Gamma} \alpha_i u_i$. In [25, Theorem 3.4], Kahl and
 501 Rittich show that, if for some positive constant c_K , \tilde{U} satisfies

$$502 \quad (4.19) \quad \|x - \Pi_{\tilde{U}}x\|_2^2 \leq c_K \frac{\|x\|_{I-H}^2}{\|I - H\|_2},$$

then the effective condition number of $\mathcal{P}(I - H)$ satisfies

$$\kappa_{\text{eff}}(\mathcal{P}(I - H)) \leq c_K.$$

503 Let $t \leq k$ and consider

$$\begin{aligned} 504 \quad \|x - \Pi_{\tilde{U}}x\|_2 &= \left\| \sum_{i=1}^{n_\Gamma} \alpha_i u_i - \Pi_{\tilde{U}} \sum_{i=1}^{n_\Gamma} \alpha_i u_i \right\|_2 \\ 505 \quad &\leq \left\| \sum_{i=t+1}^{n_\Gamma} (I - \Pi_{\tilde{U}}) \alpha_i u_i \right\|_2 + \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}}u_i\|_2 \\ 506 \quad &\leq \left\| \sum_{i=t+1}^{n_\Gamma} \alpha_i u_i \right\|_2 + \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}}u_i\|_2. \end{aligned}$$

508 The last inequality is obtained using the fact that $I - \Pi_{\tilde{U}}$ is an orthogonal projector.
 509 Now bound each term on the right separately. We have

$$\begin{aligned} 510 \quad \left\| \sum_{i=t+1}^{n_\Gamma} \alpha_i u_i \right\|_2 &\leq \frac{1}{\sqrt{\lambda_{t+1}}} \left\| \sum_{i=t+1}^{n_\Gamma} \sqrt{\lambda_{t+1}} \alpha_i u_i \right\|_2 \leq \frac{1}{\sqrt{\lambda_{t+1}}} \left\| \sum_{i=t+1}^{n_\Gamma} \sqrt{\lambda_i} \alpha_i u_i \right\|_2 \\ 511 \quad &\leq \frac{1}{\sqrt{\lambda_{t+1}}} \sum_{i=t+1}^{n_\Gamma} \lambda_i \alpha_i^2 = \frac{1}{\sqrt{\lambda_{t+1}}} \|x - \Pi_{U_t}x\|_{I-H} = \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \frac{\|x - \Pi_{U_t}x\|_{I-H}}{\sqrt{\|I - H\|_2}}. \blacksquare \end{aligned}$$

513 From (4.15), $\gamma_{i,k} \leq 1$ for $i = 1, \dots, t$, thus,

$$\begin{aligned}
 514 \quad \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}} u_i\|_2 &\leq \sum_{i=1}^t |\alpha_i| \gamma_{i,k}^{q+1} c \leq c \gamma_{t,k}^{q+\frac{1}{2}} \sum_{i=1}^t |\alpha_i| \sqrt{\gamma_{i,k}} \\
 515 \quad &= c \gamma_{t,k}^{q+\frac{1}{2}} \sqrt{\lambda_{k+1}^{-1} - 1} \sum_{i=1}^t |\alpha_i| \frac{1}{\sqrt{\lambda_i^{-1} - 1}} \\
 516 \quad &\leq c \gamma_{t,k}^{q+\frac{1}{2}} \frac{1}{\sqrt{\lambda_{k+1}}} \sum_{i=1}^t |\alpha_i| \frac{1}{\sqrt{\lambda_i^{-1} - 1}}.
 \end{aligned}$$

518 Assuming that $\lambda_i \leq 1/2$ for $i = 1, \dots, t$, we have

$$\begin{aligned}
 519 \quad \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}} u_i\|_2 &\leq \sqrt{2} c \gamma_{t,k}^{q+\frac{1}{2}} \frac{1}{\sqrt{\lambda_{k+1}}} \sum_{i=1}^t |\alpha_i| \frac{1}{\sqrt{\lambda_i^{-1}}} \\
 520 \quad &\leq \sqrt{2} c \gamma_{t,k}^{q+\frac{1}{2}} \frac{1}{\sqrt{\lambda_{k+1}}} \sum_{i=1}^t |\alpha_i| \sqrt{\lambda_i}.
 \end{aligned}$$

522 Using the fact that the l_1 and l_2 norms are equivalent, we have

$$\begin{aligned}
 523 \quad \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}} u_i\|_2 &\leq c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}} \frac{1}{\sqrt{\lambda_{k+1}}} \sqrt{\sum_{i=1}^t \alpha_i^2 \lambda_i} \\
 524 \quad &= c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}} \frac{1}{\sqrt{\lambda_{k+1}}} \|\Pi_{U_t} x\|_{I-H} \\
 525 \quad &= c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}} \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{k+1}}} \frac{\|\Pi_{U_t} x\|_{I-H}}{\sqrt{\|I-H\|_2}}.
 \end{aligned}$$

527 Since $\lambda_k \geq \lambda_t$ we have

$$528 \quad \sum_{i=1}^t |\alpha_i| \|u_i - \Pi_{\tilde{U}} u_i\|_2 \leq c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}} \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \frac{\|\Pi_{U_t} x\|_{I-H}}{\sqrt{\|I-H\|_2}}.$$

530 It follows that

$$\begin{aligned}
 531 \quad \|x - \Pi_{\tilde{U}} x\|_2 &\leq \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \frac{\|x - \Pi_{U_t} x\|_{I-H}}{\sqrt{\|I-H\|_2}} + c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}} \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \frac{\|\Pi_{U_t} x\|_{I-H}}{\sqrt{\|I-H\|_2}} \\
 532 \quad &\leq \sqrt{2} \max(c \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}}, 1) \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \frac{\|x\|_{I-H}}{\sqrt{\|I-H\|_2}}.
 \end{aligned}$$

534 Hence (4.19) is satisfied and we have

$$535 \quad \kappa_{\text{eff}}(\mathcal{P}(I-H)) \leq 2 \max(2c^2 t \gamma_{t,k}^{2q+1}, 1) \frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}.$$

536 Thus,

$$537 \quad \mathbb{E} \left(\sqrt{\kappa_{\text{eff}}(\mathcal{P}(I-H))} \right) \leq \sqrt{2} \max(\mathbb{E}(c) \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}}, 1) \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}}.$$

538 Since t is chosen arbitrarily between 1 and k we have

$$539 \quad (4.20) \quad \mathbb{E} \left(\sqrt{\kappa_{\text{eff}}(\mathcal{P}(I-H))} \right) \leq \sqrt{2} \min_{1 \leq t \leq k} \left(\max \left(\mathbb{E}(c) \sqrt{2t} \gamma_{t,k}^{q+\frac{1}{2}}, 1 \right) \sqrt{\frac{\lambda_{n_\Gamma}}{\lambda_{t+1}}} \right).$$

540 Because $\mathbb{E}(c)$ can be bounded by a polynomial of degree 1 in k and $\gamma_{t,k} \leq 1$,
 541 $\max(4t\gamma_{t,k}^{2q+1}(\mathbb{E}(c))^2, 2)$ can be bounded by a polynomial of degree 3 in k independent
 542 of the spectrum of $I-H$. \square

543 Note that, in practice, when the problem is challenging, a few eigenvalues of
 544 $R_\Gamma^{-\top} S_\Gamma R_\Gamma^{-1}$ are close to the origin. This is reflected in a rapid and exponential
 545 decay of the values of the entries of $\Lambda^{-1} - I$. **Figure 2** depicts the bound obtained
 546 in **Proposition 4.1** for different values of k and q for problem s3rmt3m3.

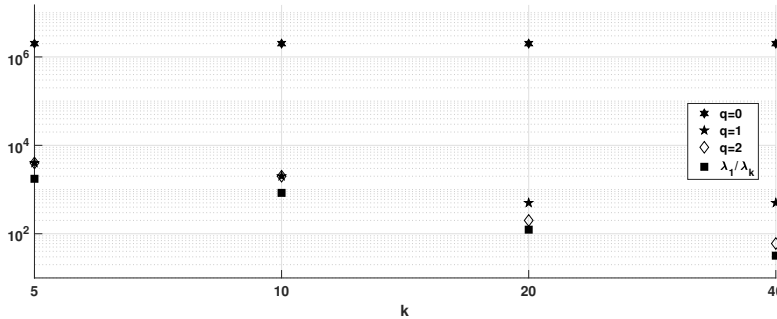


FIG. 2. Problem s3rmt3m3: Values of the bound (4.20) on $(\mathbb{E}(\sqrt{\kappa_{\text{eff}}(\mathcal{P}(I-H))})^2$ for a range of values of k and q .

547 **5. Numerical Experiments.** We use 64 subdomains (i.e., A_I is a 64-block
 548 diagonal matrix) for each of our test matrices with the exception of one problem. The
 549 matrix nd3k is much denser than the others, and we use only two blocks (to reduce
 550 the runtime). For comparison purposes, we include results for the Schur complement
 551 preconditioners \tilde{S}_1 and \tilde{S}_2 given by (2.14) and (2.17), respectively. As demonstrated
 552 in subsection 3.1, the latter is too costly to be practical, however, its performance
 553 is the ideal since it guarantees the smallest spectral condition number for a fixed
 554 deflation subspace. Therefore, the quality of the Nyström–Schur preconditioner will
 555 be measured in terms of how close its performance is to that of \tilde{S}_2 and the reduction in
 556 iteration it gives compared to \tilde{S}_1 . For a given problem, the right-hand side vector is the
 557 same for all the tests: it is generated randomly with entries from the standard normal
 558 distribution. The relative convergence tolerance for PCG is 10^{-6} . Unless otherwise
 559 specified, the parameters within Nyström’s method (Algorithm 2.1) are rank $k = 20$,
 560 oversampling $p = 0$, and power iteration $q = 0$. To ensure fair comparisons, the
 561 random matrices generated in different runs of the Nyström algorithm use the same
 562 seed. We employ the Nyström–Schur variant \mathcal{M}_2 (4.10) (recall that its construction
 563 does not require the Cholesky factors of A_Γ). The relative convergence tolerance used
 564 when solving the SPD system (4.13) is $\varepsilon_{S_I} = 0.1$. This system (4.13) is preconditioned
 565 by the block diagonal matrix A_I . We denote by it_{S_I} the number of block PCG
 566 iterations required to solve (4.13) during the construction of the Nyström–Schur
 567 preconditioners (it is zero for \tilde{S}_1 and \tilde{S}_2), and by it_{PCG} the PCG iteration count

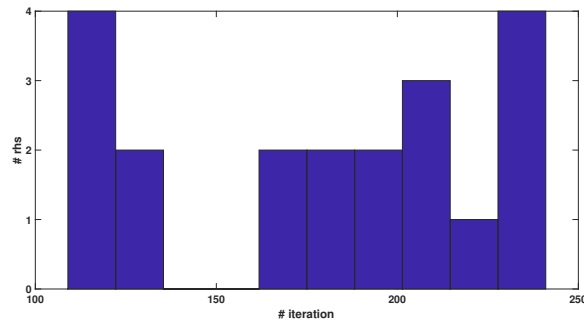


FIG. 3. Histogram of the PCG iteration counts for (4.13) for problem *bcsstk38*. The number of right hand sides for which the iteration count is between $[k, k + 10)$, $k = 100, \dots, 240$, is given.

Identifier	Classic		Block	
	<i>iters</i>	<i>it_{PCG}</i>	<i>iters</i>	<i>it_{PCG}</i>
<i>bcsstk38</i>	238	186	46	173
<i>el2d</i>	549	261	72	228
<i>el3d</i>	95	56	24	52
<i>msc10848</i>	203	194	47	166
<i>nd3k</i>	294	191	32	178
<i>s3rmt3m3</i>	403	157	37	98

TABLE 5

A comparison of the performance of classic and block PCG. *iters* denotes the iteration count for solving (4.13) (details in the text) and *it_{PCG}* is the iteration count for solving (2.13).

568 for solving (2.13). The total number of iterations is $it_{\text{total}} = it_{S_I} + it_{\text{PCG}}$. We use the
569 code [1] to generate the numerical experiments.

570 **5.1. Linear system with S_I .** We start by considering how to efficiently
571 compute an approximate solution of (4.13).

572 **5.1.1. Block and classic CG.** The system (4.13) has $k + p$ right hand sides.
573 The number of iterations required by PCG to solve each right hand side is different
574 and the variation can be large; this is illustrated in Figure 3 for problem *bcsstk38*.
575 Here we report the number of right hand sides for which the iteration count lies in
576 the interval $[k, k + 10)$, $k = 100, \dots, 240$. For example, there are 4 right hand sides
577 for which the count is between 110 and 119. Similar behaviour was observed for our
578 other test problems.

579 Table 5 reports the iteration counts for the classical PCG method and the
580 breakdown-free block PCG method [21, 35]. For PCG, *iters* is the largest PCG
581 iteration count over the $k + p$ right hand sides. For the block method, $iters = it_{S_I}$
582 is the number of block PCG iterations. As expected from the theory, the block method
583 significantly reduces the (maximum) iteration count. For our examples, it also leads
584 to a modest reduction in the iteration count *it_{PCG}* for solving (2.13).

585 **5.1.2. Impact of tolerance ε_{S_I} .** We now study the impact of the convergence
586 tolerance ε_{S_I} used when solving (4.13) on the quality of the Nyström-Schur
587 preconditioner. In Table 6, we present results for three test problems that illustrate
588 the (slightly) different behaviors we observed. The results demonstrate numerically

Identifier	ε_{S_I}	\mathcal{M}_2		\tilde{S}_1	\tilde{S}_2
		it_{S_I}	it_{PCG}		
el2d	0.8	1	500+	914	231
	0.5	68	228		
	0.3	70	228		
	0.1	72	228		
	0.01	78	228		
el3d	0.8	1	173	174	37
	0.5	2	171		
	0.3	22	52		
	0.1	24	52		
	0.01	27	52		
nd3k	0.8	32	178	603	143
	0.5	32	178		
	0.3	32	178		
	0.1	32	178		
	0.01	33	178		

TABLE 6

The effects of the convergence tolerance ε_{S_I} on the quality of the Nyström–Schur preconditioner.

Identifier	\mathcal{M}_1	$\mathcal{M}_{1-A-DEF}$	\mathcal{M}_2	$\mathcal{M}_{2-A-DEF}$	\mathcal{M}_3	$\mathcal{M}_{3-A-DEF}$	\tilde{S}_1	\tilde{S}_2
bcsstk38	218	218	219	219	360	313	584	122
el2d	266	267	300	300	282	282	914	231
el3d	73	72	76	75	78	76	174	37
msc10848	206	205	213	211	216	222	612	116
nd3k	205	205	210	210	211	211	603	143
s3rmt3m3	127	127	135	134	161	153	441	70

TABLE 7

Comparison of it_{total} for the variants of the Nyström–Schur preconditioner and \tilde{S}_1 and \tilde{S}_2 . $\varepsilon_{S_I} = 0.1$.

589 that a large tolerance can be used without affecting the quality of the preconditioner.
590 Indeed, using $\varepsilon_{S_I} = 0.3$ leads to a preconditioner whose efficiency is close to that of the
591 ideal (but impractical) two-level preconditioner \tilde{S}_2 . The use of a large ε_{S_I} to limit it_{S_I}
592 is crucial in ensuring low construction costs for the Nyström–Schur preconditioners.

593 **5.2. Type of preconditioner.** We next compare the performances of the
594 variants \mathcal{M}_i and $\mathcal{M}_{i-A-DEF}$ ($i = 1, 2, 3$) of the Nyström–Schur preconditioner presented
595 in section 4. In Table 7, we report the total iteration count it_{total} . All the variants
596 have similar behaviors and have a significantly smaller count than the one-level
597 preconditioner \tilde{S}_1 .

598 **5.3. Varying the rank and the oversampling parameter.** We now look
599 at varying the rank k within the Nyström algorithm and demonstrate numerically
600 that the efficiency of the preconditioner is robust with respect to the oversampling
601 parameter p . For problem s3rmt3m3, Table 8 compares the iteration counts for \mathcal{M}_2
602 with that of the ideal two-level preconditioner \tilde{S}_2 for k ranging from 5 to 320. For \tilde{S}_1 ,
603 the iteration count is 441. This demonstrates the effectiveness of the Nyström–Schur
604 preconditioner in reducing the iteration count. Increasing the size of the deflation
605 subspace (the rank k) steadily reduces the iteration count required to solve the S_I

	k	5	10	20	40	80	160	320
\mathcal{M}_2	it_{S_I}	97	57	37	23	16	11	8
	it_{PCG}	244	203	98	53	30	20	14
\tilde{S}_2	it_{PCG}	212	153	70	37	22	13	9

TABLE 8

Problem *s3rmt3m3*: Impact of the rank k on the iteration counts ($p = 0$).

p	0	5	10	20	40
it_{S_I}	37	31	28	23	20
it_{PCG}	98	86	79	77	74

TABLE 9

Problem *s3rmt3m3*: Impact of the oversampling parameter p on the iteration counts ($k = 20$).

606 system (4.13). For the same test example, Table 9 presents the iteration counts for
 607 a range of values of the oversampling parameter p (here $k = 20$). We observe that
 608 the counts are relatively insensitive to p but, as p increases, it_{PCG} reduces towards
 609 the lower bound of 70 PCG iterations required by \tilde{S}_2 . Similar behavior was noticed
 610 for our other test examples. Although increasing k and p improves the efficiency
 611 of the Nyström–Schur preconditioner, this comes with extra costs during both the
 612 construction of the preconditioner and its application. Nevertheless, the savings from
 613 the reduction in the iteration count and the efficiency in solving block linear systems
 614 of equations for moderate block sizes (for example, $k = 40$) typically outweigh the
 615 increase in construction costs.

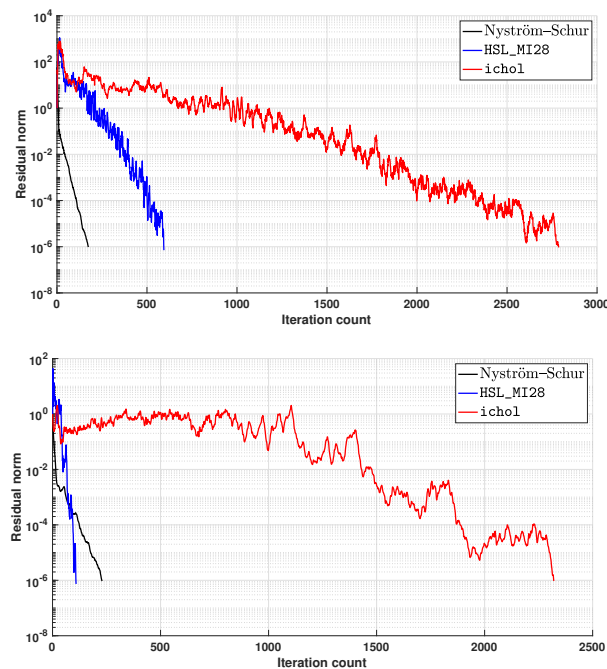
616 **5.4. Comparisons with incomplete Cholesky factorization**
 617 **preconditioners.** Finally, we compare the Nyström–Schur preconditioner with
 618 two incomplete Cholesky factorization preconditioners applied to original system.
 619 The first is the Matlab variant `ichol` with the global diagonal shift set to 0.1 and
 620 default values for other parameters and the second is the Matlab interface to the
 621 incomplete Cholesky (IC) factorization preconditioner `HSL_MI28` [39] from the HSL
 622 library [20] using the default parameter settings. IC preconditioners are widely used
 623 but their construction is often serial, potentially limiting their suitability for very
 624 large problems (see [19] for an IC preconditioner that can be parallelised). In terms
 625 of iteration counts, the Nyström–Schur and the `HSL_MI28` preconditioners are clearly
 626 superior to the simple `ichol` preconditioner, with neither consistently offering the
 627 best performance. Figure 4 presents the residual norm history for PCG. This is
 628 confirmed by the results in the Appendix for our large test set. The residual norm for
 629 \mathcal{M}_2 decreases monotonically while for the IC preconditioners we observe oscillatory
 630 behaviour.

631 Because our implementation of the Nyström–Schur preconditioner is in Matlab,
 632 we are not able to provide performance comparisons in terms of computation times.
 633 Having demonstrated the potential of our two-level Nyström–Schur preconditioner,
 634 one of our objectives for the future is to develop an efficient (parallel) implementation
 635 in Fortran that will be included within the HSL library. This will allow users to
 636 test out the preconditioner and to assess the performance of both constructing and
 637 applying the preconditioner. Our preliminary work on this is encouraging.

Identifier	\mathcal{M}_2		HSL_MI28	icho1
	it_{S_I}	it_{PCG}		
bcsstk38	46	173	593	2786
ela2d	72	228	108	2319
ela3d	24	52	36	170
msc10848	47	166	145	784
nd3k	32	178	102	1231
s3rmt3m3	37	98	610	2281

TABLE 10

PCG iteration counts for the Nyström–Schur preconditioner \mathcal{M}_2 (with $k = 20$) and the IC preconditioners HSL_MI28 and icho1.

FIG. 4. PCG residual norm history for test examples *bcsstk38* (top) and *ela2d* (bottom).

638 **6. Concluding comments.** In this paper, we have investigated using
639 randomized methods to construct efficient and robust preconditioners for use with
640 CG to solve large-scale SPD linear systems. The approach requires an initial
641 ordering to doubly bordered block diagonal form and then uses a Schur complement
642 approximation. We have demonstrated that by carefully posing the approximation
643 problem we can apply randomized methods to construct high quality preconditioners,
644 which gives an improvement over previously proposed methods that use low rank
645 approximation strategies. We have presented a number of variants of our new
646 Nyström–Schur preconditioner. During the preconditioner construction, we must
647 solve a smaller linear system with multiple right-hand sides. Our numerical
648 experiments have shown that a small number of iterations of block CG are needed
649 to obtain an approximate solution that is sufficient to construct an effective
650 preconditioner.

651 Currently, the construction and application of our Nyström–Schur preconditioners
 652 requires the solution of linear systems with the block matrix A_Γ (2.9). Given
 653 the promising results presented in this paper, in the future we plan to investigate
 654 employing a recursive approach, following ideas given in [49]. This will only require
 655 the solution of systems involving a much smaller matrix and will lead to a practical
 656 approach for very large-scale SPD systems. A parallel implementation of the
 657 preconditioner will also be developed.

658 **Appendix A. Extended numerical experiments.** Here we present results
 659 for a larger test set. The problems are given in Table 11. We selected all the SPD
 660 matrices in the SuiteSparse Collection with n lying between 5K and 100K, giving us a
 661 set of 71 problems. For each problem, we ran PCG with the \tilde{S}_1 , \mathcal{M}_2 , \tilde{S}_2 and HSL_MI28
 662 preconditioners. In all the tests, we use 64 subdomains. For \mathcal{M}_2 , we used $k = 20$
 663 and set $p = q = 0$. Iteration counts are given in the table, whilst performance profiles
 664 [6] are presented in Figure 5. In recent years, performance profiles have become a
 665 popular and widely used tool for providing objective information when benchmarking
 666 algorithms. The performance profile takes into account the number of problems solved
 667 by an algorithm as well as the cost to solve it. It scales the cost of solving the problem
 668 according to the best solver for that problem. In our case, the performance cost is
 669 the iteration count (for \mathcal{M}_2 , we sum the counts it_{S_1} and it_{PCG}). Note that we do
 670 not include \tilde{S}_2 in the performance profiles because it is an ideal but impractical two-
 671 level preconditioner and, as such, it always outperforms \mathcal{M}_2 . The performance profile
 672 shows that on the problems where \tilde{S}_1 struggles, there is little to choose between the
 overall quality of \mathcal{M}_2 and HSL_MI28.

Identifier	\tilde{S}_1		\mathcal{M}_2		\tilde{S}_2	HSL_MI28	$\kappa(A)$
	it_{S_1}	it_{PCG}	it_{S_1}	it_{PCG}			
aft01	118	19	45	31	17	9e+18	
apache1	667	122	291	192	72	3e+06	
bcstkt17	349	46	55	48	59	1e+10	
bcstkt18	136	40	77	45	26	6e+11	
bcstkt25	†	92	660	453	254	1e+13	
bcstkt36	451	64	214	169	†	1e+12	
bcstkt38	584	46	171	122	593	6e+16	
bodyy6	182	53	163	129	5	9e+04	
cant	†	57	228	396	933	5e+10	
cfid1	209	30	72	50	274	1e+06	
consph	185	47	177	136	50	3e+07	
gridgena	426	90	377	298	66	6e+05	
gyro	†	55	346	518	319	4e+09	
gyro_k	†	55	346	518	319	3e+09	
gyro_m	165	16	34	22	17	1e+07	
m_t1	867	85	247	187	†	3e+11	
minsurfo	15	3	15	13	3	8e+01	
mssl0848	612	47	168	116	145	3e+10	
mssl23052	479	69	220	175	†	1e+12	
nasasrb	1279	135	496	421	†	1e+09	
nd3k	1091	56	301	230	102	5e+07	
nd6k	1184	108	325	248	116	6e+07	
oilpan	647	67	122	72	507	4e+09	
olafu	1428	69	489	757	557	2e+12	
pdb1HYS	869	89	83	274	483	2e+12	
vanbody	†	287	1106	769	†	4e+03	
ct20stif	1296	90	232	281	†	2e+14	
nd12k	1039	155	337	265	111	2e+08	
nd24k	1093	165	386	268	120	2e+08	
s1rmq4m1	154	19	50	32	33	5e+06	
s1rmt3m1	192	24	59	39	18	3e+08	
s2rmq4m1	231	28	54	41	39	4e+08	
s2rmt3m1	260	31	64	45	33	3e+11	
s3dkq4m2	†	148	339	236	610	6e+11	

Identifier	\tilde{S}_1		\mathcal{M}_2		\tilde{S}_2	HSL_MI28	$\kappa(A)$
	it_{S_1}	it_{PCG}	it_{S_1}	it_{PCG}			
s3dkk3m2	†	164	338	270	1107	3e+10	
s3rmq4m1	356	31	80	58	472	4e+10	
s3rmt3m1	434	36	101	64	413	4e+10	
s3rmt3m3	441	37	101	70	610	3e+00	
ship_001	1453	367	600	368	1177	6e+09	
smt	399	59	112	72	95	1e+09	
thermal1	169	30	62	47	30	4e+01	
Pres_Poisson	92	13	29	19	32	3e+06	
crankseg_1	92	16	49	33	34	9e+18	
crankseg_2	89	17	47	32	38	8e+06	
Kuu	81	16	44	31	10	3e+04	
bodyy5	72	19	67	57	4	9e+03	
Dubcova2	62	11	32	23	14	1e+04	
cbuckle	55	9	51	39	47	7e+07	
fv3	50	12	31	21	8	4e+03	
Dubcova1	39	8	24	15	7	2e+03	
bodyy4	34	8	29	24	4	1e+03	
jnlbrng1	22	4	21	19	4	1e+02	
bundle1	13	3	8	5	5	1e+04	
t2dalu_e	12	3	12	11	3	3e+07	
obstclae	12	3	12	12	3	4e+01	
torsion1	12	3	12	12	3	8e+03	
wathen100	12	3	12	11	3	2e+07	
wathen120	12	3	12	11	3	2e+07	
fv1	7	2	7	7	3	1e+01	
fv2	7	2	7	7	3	1e+01	
shallow_water2	7	40	7	7	3	3e+12	
shallow_water1	5	20	5	5	2	1e+01	
Muu	6	1	6	6	2	1e+02	
qa8fm	6	1	6	6	2	1e+02	
crystm02	6	1	6	5	2	4e+02	
crystm03	6	1	6	5	2	4e+02	
finan512	5	1	5	5	3	9e+01	
ted_B_unscaled	3	1	3	4	2	4e+05	
ted_B	2	1	3	3	2	2e+11	
Trefethen_20000b	3	1	2	2	3	1e+05	
Trefethen_20000	4	1	2	2	3	2e+05	

TABLE 11

PCG iteration counts for SPD matrices from the SuiteSparse Collection with n ranging between 5K and 100K.

673

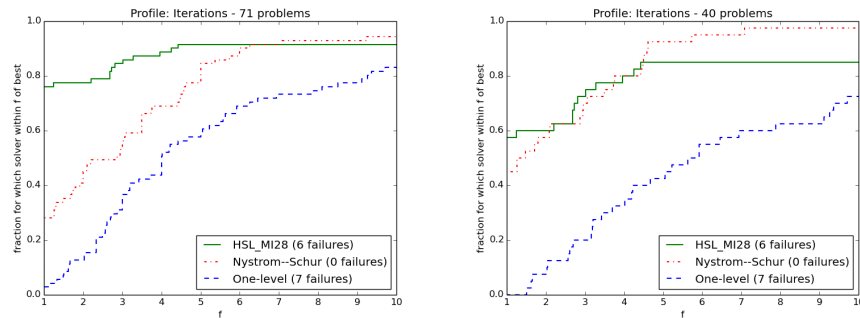


FIG. 5. Iteration count performance profile for the large test set. The 40 problems used in the right hand plot are the subset for which the \tilde{S}_1 (one-level) iteration count exceeded 100.

674

REFERENCES

- 675 [1] H. AL DAAS, *haldaas/Nystrom-Schur-Preconditioner: version reproducing paper numerical*
676 *experiments*, June 2021, <https://doi.org/10.5281/zenodo.4957301>.
- 677 [2] H. AL DAAS AND L. GRIGORI, *A class of efficient locally constructed preconditioners based on*
678 *coarse spaces*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 66–91.
- 679 [3] H. AL DAAS, L. GRIGORI, P. JOLIVET, AND P.-H. TOURNIER, *A multilevel Schwarz*
680 *preconditioner based on a hierarchy of robust coarse spaces*, SIAM Journal on Scientific
681 *Computing*, (2021), pp. A1907–A1928.
- 682 [4] H. AL DAAS, P. JOLIVET, AND J. A. SCOTT, *A robust algebraic domain decomposition*
683 *preconditioner for sparse normal equations*, 2021, <https://arxiv.org/abs/2107.09006>.
- 684 [5] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Transactions
685 *on Mathematical Software*, 38 (2011), pp. 1–28.
- 686 [6] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*,
687 *Mathematical Programming*, 91 (2002), pp. 201–213.
- 688 [7] V. DOLEAN, P. JOLIVET, AND F. NATAF, *An introduction to domain decomposition methods*,
689 *Society for Industrial and Applied Mathematics*, Philadelphia, PA, 2015. Algorithms,
690 *theory, and parallel implementation*.
- 691 [8] Z. DOSTÁL, *Conjugate gradient method with preconditioning by projector*, International Journal
692 *of Computer Mathematics*, 23 (1988), pp. 315–323.
- 693 [9] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices, Second*
694 *Edition*, Oxford University Press, London, 2017.
- 695 [10] J. FRANK AND C. VUIK, *On the construction of deflation-based preconditioners*, SIAM Journal
696 *on Scientific Computing*, 23 (2001), pp. 442–462.
- 697 [11] A. GAUL, M. H. GUTKNECHT, J. LIESEN, AND R. NABBEN, *A framework for deflated and*
698 *augmented Krylov subspace methods*, SIAM Journal on Matrix Analysis and Applications,
699 34 (2013), pp. 495–518.
- 700 [12] A. GITTENS AND M. W. MAHONEY, *Revisiting the Nyström method for improved large-scale*
701 *machine learning*, J. Mach. Learn. Res., 17 (2016), pp. 3977–4041.
- 702 [13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University
703 *Press*, third ed., 1996.
- 704 [14] L. GRIGORI, F. NATAF, AND S. YOUSEF, *Robust algebraic Schur complement preconditioners*
705 *based on low rank corrections*, Research Report RR-8557, INRIA, July 2014, [https://hal.](https://hal.inria.fr/hal-01017448)
706 [inria.fr/hal-01017448](https://hal.inria.fr/hal-01017448).
- 707 [15] M. H. GUTKNECHT, *Deflated and augmented Krylov subspace methods: A framework for*
708 *deflated BiCG and related solvers*, SIAM Journal on Matrix Analysis and Applications,
709 35 (2014), pp. 1444–1466.
- 710 [16] N. HALKO, P.-G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness:*
711 *Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM
712 *Review*, 53 (2011), pp. 217–288.
- 713 [17] F. HECHT, *New development in freefem++*, Journal of Numerical Mathematics, 20 (2012),
714 pp. 251–265.
- 715 [18] N. J. HIGHAM AND T. MARY, *A new preconditioner that exploits low-rank approximations to*

- factorization error, *SIAM Journal on Scientific Computing*, 41 (2019), pp. A59–A82.
- [19] J. HOOK, J. SCOTT, F. TISSEUR, AND J. HOGG, *A max-plus approach to incomplete Cholesky factorization preconditioners*, *SIAM Journal on Scientific Computing*, 40 (2018), pp. A1987–A2004.
- [20] *HSL. A collection of Fortran codes for large-scale scientific computation*, 2018. <http://www.hsl.rl.ac.uk>.
- [21] H. JI AND Y. LI, *A breakdown-free block conjugate gradient method*, *BIT Numerical Mathematics*, 57 (2017), pp. 379–403.
- [22] T. B. JÖNSTHÖVEL, M. B. VAN GIJZEN, C. VUIK, C. KASBERGEN, AND A. SCARPAS, *Preconditioned conjugate gradient method enhanced by deflation of rigid body modes applied to composite materials*, *Computer Modeling in Engineering & Sciences*, 47 (2009), pp. 97–118.
- [23] T. B. JÖNSTHÖVEL, M. B. VAN GIJZEN, C. VUIK, AND A. SCARPAS, *On the use of rigid body modes in the deflated preconditioned conjugate gradient method*, *SIAM Journal on Scientific Computing*, 35 (2013), pp. B207–B225.
- [24] E. F. KAASSCHIETER, *Preconditioned conjugate gradients for solving singular systems*, *Journal of Computational and Applied Mathematics*, 24 (1988), pp. 265–275.
- [25] K. KAHL AND H. RITTICH, *The deflated conjugate gradient method: Convergence, perturbation and accuracy*, *Linear Algebra and its Applications*, 515 (2017), pp. 111–129.
- [26] G. KARYPIS AND V. KUMAR, *METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, Technical Report 97-061, University of Minnesota, Department of Computer Science and Army HPC Research Center, 1997.
- [27] R. LI, Y. XI, AND Y. SAAD, *Schur complement-based domain decomposition preconditioners with low-rank corrections*, *Numerical Linear Algebra with Applications*, 23 (2016), pp. 706–729.
- [28] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: Foundations and algorithms*, *Acta Numerica*, 29 (2020), pp. 403–572.
- [29] *METIS - serial graph partitioning and fill-reducing matrix ordering*, 2020. <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>.
- [30] R. NABBEN AND C. VUIK, *A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners*, *Numerical Linear Algebra with Applications*, 15 (2008), pp. 355–372.
- [31] Y. NAKATSUKASA, *Fast and stable randomized low-rank matrix approximation*, 2020, <https://arxiv.org/abs/2009.11392>.
- [32] F. NATAF, H. XIANG, V. DOLEAN, AND N. SPILLANE, *A coarse space construction based on local Dirichlet-to-Neumann maps*, *SIAM Journal on Scientific Computing*, 33 (2011), pp. 1623–1642.
- [33] R. A. NICOLAIDES, *Deflation of conjugate gradients with applications to boundary value problems*, *SIAM J. on Numerical Analysis*, 24 (1987), pp. 355–365.
- [34] E. J. NYSTRÖM, *Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben*, *Acta Mathematica*, 54 (1930), pp. 185–204.
- [35] D. P. O’LEARY, *The block conjugate gradient algorithm and related methods*, *Linear Algebra and its Applications*, 29 (1980), pp. 293–322.
- [36] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd ed., 2003.
- [37] A. K. SAIBABA, *Randomized subspace iteration: Analysis of canonical angles and unitarily invariant norms*, *SIAM Journal on Matrix Analysis and Applications*, 40 (2019), pp. 23–48.
- [38] J. A. SCOTT, *A parallel frontal solver for finite element applications*, *International J. of Numerical Methods in Engineering*, 50 (2001), pp. 1131–1144.
- [39] J. A. SCOTT AND M. TŪMA, *HSL_MI28: An efficient and robust limited-memory incomplete Cholesky factorization code*, *ACM Transactions on Mathematical Software*, 40 (2014), pp. 24:1–19.
- [40] N. SPILLANE, V. DOLEAN, P. HAURET, F. NATAF, C. PECHSTEIN, AND R. SCHEICHL, *Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps*, *Numerische Mathematik*, 126 (2014), pp. 741–770.
- [41] N. SPILLANE AND D. RIXEN, *Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms*, *International Journal for Numerical Methods in Engineering*, 95 (2013), pp. 953–990.
- [42] G. W. STEWART, *A Krylov–Schur algorithm for large eigenproblems*, *SIAM Journal on Matrix Analysis and Applications*, 23 (2002), pp. 601–614.

- 778 [43] J. M. TANG, S. P. MACLACHLAN, R. NABBEN, AND C. VUIK, *A comparison of two-level*
779 *preconditioners based on multigrid and deflation*, SIAM Journal on Matrix Analysis and
780 Applications, 31 (2010), pp. 1715–1739.
- 781 [44] J. M. TANG, R. NABBEN, C. VUIK, AND Y. A. ERLANGGA, *Comparison of two-level*
782 *preconditioners derived from deflation, domain decomposition and multigrid methods*,
783 Journal of Scientific Computing, 39 (2009), pp. 340–370.
- 784 [45] C. VUIK, A. SEGAL, AND J. A. MELJERINK, *An efficient preconditioned CG method for the*
785 *solution of a class of layered problems with extreme contrasts in the coefficients*, Journal
786 of Computational Physics, 152 (1999), pp. 385–403.
- 787 [46] C. VUIK, A. SEGAL, J. A. MELJERINK, AND G. T. WIJMA, *The construction of projection vectors*
788 *for a deflated ICCG method applied to problems with extreme contrasts in the coefficients*,
789 Journal of Computational Physics, 172 (2001), pp. 426–450.
- 790 [47] C. K. I. WILLIAMS AND M. SEEGER, *Using the Nyström method to speed up kernel machines*,
791 in Advances in Neural Information Processing Systems 13, T. K. Leen, T. G. Dietterich,
792 and V. Tresp, eds., MIT Press, 2001, pp. 682–688.
- 793 [48] D. WOODRUFF, *Sketching as a Tool for Numerical Linear Algebra*, Foundations and Trends(r)
794 in Theoretical Computer Science Series, Now Publishers, 2014.
- 795 [49] Y. XI, R. LI, AND Y. SAAD, *An algebraic multilevel preconditioner with low-rank corrections*
796 *for sparse symmetric matrices*, SIAM Journal on Matrix Analysis and Applications, 37
797 (2016), pp. 235–259.
- 798 [50] Q. ZHENG, Y. XI, AND Y. SAAD, *A power Schur complement low-rank correction preconditioner*
799 *for general sparse linear systems*, SIAM Journal on Matrix Analysis and Applications, 42
800 (2021), pp. 659–682.