IMA Journal of Numerical Analysis (2025) **00**, 1–42
https://doi.org/DOI HERE

Global Iterative Methods for Sparse Approximate Inverses of Symmetric Positive-Definite Matrices

NICOLAS VENKOVIC AND HARTWIG ANZT
Computational Mathematics
School of Computation, Information and Technology (CIT)
Technical University of Munich
Heilbronn, 74076, Germany
*Corresponding author: nicolas.venkovic@tum.de,venkovic@gmail.com

[Received on 12 November 2025; revised on Date Month Year; accepted on Date Month Year]

The nonlinear (preconditioned) conjugate gradient N(P)CG method and the locally optimal (preconditioned) minimal residual LO(P)MR method, both of which are used for the iterative computation of sparse approximate inverses (SPAIs) of symmetric positive-definite (SPD) matrices, are introduced and analyzed. The (preconditioned) conjugate gradient (P)CG method is also employed and presented for comparison. The N(P)CG method is defined as a one-dimensional projection with residuals made orthogonal to the current search direction, itself made A-orthogonal to the last search direction. The residual orthogonality, expressed via Frobenius inner product, actually holds against all previous search directions, making each iterate globally optimal, that is, that minimizes the Frobenius A-norm of the error over the affine Krylov subspace of A^2 generated by the initial gradient. The LO(P)MR method is a two-dimensional projection method that enriches iterates produced by the (preconditioned) minimal residual (P)MR method. These approaches differ from existing descent methods and aim to accelerate convergence compared to other global iteration methods, including (P)MR and (preconditioned) steepest descent (P)SD, previously used for SPAI computation. The methods are implemented with practical dropping strategies to control the growth of nonzero components in the approximate inverse. Numerical experiments are performed in which approximate inverses of several sparse SPD matrices are computed. While N(P)CG does provide a slight improvement over (P)SD, it remains generally less effective than standard (P)MR. On the other hand, while (P)CG does sometimes improve upon (P)MR, its convergence is significantly affected by the dropping of nonzero components as well as ill-conditioning, and small eigenvalues of A. LO(P)MR, which is generally more robust than both (P)MR and (P)CG, consistently outperforms all other methods, converging faster to better approximations, often with smaller nonzero densities.

Keywords: Sparse approximate inverses; Global iterative methods; Nonlinear conjugate gradient; Local optimality.

1. Introduction

Sparse approximate inverses (SPAIs) have been used to precondition iterative solvers since the 1970s (see [3, 13]), in which case a sparse matrix $M \in \mathbb{R}^{n\times n}$ is formed explicitly to approximate the inverse A^{-1} of some typically sparse matrix $A \in \mathbb{R}^{n\times n}$, and used as a preconditioner to accelerate the iterative solve of linear systems of the form $Ax = b$ (resp., $AX = B$), see [5, 8] for overviews. The main advantage of resorting to SPAIs is that their application, which boils down to the execution of sparse matrix-vector (resp., sparse matrix-matrix) kernels, is both portable and parallelizable. The resort to SPAIs for preconditioning purposes relies on the assumption that sufficiently good but sparse approximations of A^{-1} exist and can be found. Strictly speaking, the inverse of a sparse matrix is generally dense.

© The Author(s) 2025. Published by Oxford University Press on behalf of the Institute of Mathematics and its Applications. All rights reserved.

More precisely, we may say that the inverse of an irreducible sparse matrix is structurally full, that is, for a given irreducible sparsity pattern, it is always possible to assign numerical values to the nonzeros in such a way that all entries of the inverse are nonzero [12]. That being said, it is rather common that many entries in the inverse of a sparse matrix have small magnitude, and dropping those values, that is, setting them to zero, minimally degrades the approximation of the inverse. A particularly telling instance of this phenomenon is that of banded symmetric positive definite (SPD) matrices whose inverses have entries bounded by an exponentially decaying envelope along rows and columns [11]. Similarly, strongly diagonally dominant matrices also exhibit a rapid decay in the magnitude of the nonzero entries of their inverses. This common feature of inverses of sparse matrices is leveraged for the design of SPAIs.

Over time, several methods have been proposed for the computation of SPAIs, see [1, 8, 9, 19] for overviews. Perhaps the most significant distinction between those methods is related to whether the SPAI is expressed as a product of structured matrices, that is, a factorization, commonly referred to as a factorized sparse approximate inverse (FSAI), see [6, 7, 15, 18], or simply as a sparse matrix, herein referred to as standard SPAI. The advantage of FSAIs is that they enable more explicit control of spectral characteristics such as non-singularity and positive definiteness. In this work, however, we focus on standard SPAIs and defer the treatment of FSAIs to another work. Most existing approaches to compute SPAIs are iterative methods aimed at minimizing the residual Frobenius norm, one exception being [14], where other norms are investigated. At each iteration of an iterative method for the computation of SPAIs, an iterate M_{i+1} of SPAI is formed from a previous iterate M_i with the goal of minimizing $\|I_n - AM_{i+1}\|$ for some given norm. The first documented attempts to develop such methods may be found in [2, 3, 4, 13]. In most of those early efforts, the sparsity pattern of the SPAI is assumed beforehand, eventually leading to decoupled and highly parallelizable least-squares problems. For those approaches, common choices of nonzero patterns are extracted from small powers of A which, in practice, can yield high nonzero densities, without guarantee of achieving a good approximation of A^{-1}. One of the most influential attempts to move away from the pre-definition of sparsity patterns is found in [10], where global iterative steepest descent and minimal residual methods are introduced, implemented with sparse data structures and computations, and deployed with dropping strategies, leading to nonzero patterns that evolve from one iteration to another with a contained nonzero density. As of today, iterative methods based on the minimal residual approach of [10] have become one of the most standard approaches to compute SPAIs.

As evidenced by experiments in Section 8.1, even when no dropping strategy is applied, global iterations based on the minimal residual and steepest descent methods often exhibit very slow convergence behaviors. More particularly, when applied to SPD matrices, these methods consistently fail to produce SPD approximate inverses, rendering the SPAI unfit to precondition iterative solves by conjugate gradient. In this work, we attempt to remedy these issues. To do so, we make the minimal residual method locally optimal, that is, we enrich the search space with the previous search direction, and apply orthogonality constraints accordingly, as a means to accelerate convergence. To improve the convergence behavior of the steepest descent method, we introduce a nonlinear conjugate gradient method in which the search directions, defined along gradients of Frobenius residual norms, are made A-orthogonal, thereby achieving a globally optimal Krylov subspace method. As a last alternative, we make use of the conjugate gradient method with matrix iterates and Frobenius inner products.

This work is organized as follows. First, the minimal residual and steepest descent methods are presented in Section 2. Then, the nonlinear conjugate gradient method is introduced in Section 3, the conjugate gradient method in Section 4, and the locally optimal minimal residual method in Section 5. Preconditioned iterations are described in Section 6, and dropping strategies in Section 7. In Section 8,

we present the results of numerical experiments which are meant to be reproducible using the Julia scripts available in the following GitHub repository:

https://github.com/venkovic/julia-global-spd-spai.

Our conclusions are presented in Section 9.

2. One-dimensional projection methods

Let $A \in \mathbb{R}^{n\times n}$ be an SPD matrix. We are interested in finding a right-approximate inverse of A, that is, $M \in \mathbb{R}^{n\times n}$ such that $I_n - AM$ is small in some sense. For that, let us consider the Frobenius inner product given by:

$$(X,Y)_F \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \mapsto \mathrm{tr}(X^T Y) \tag{2.1}$$

with the induced norm given by:

$$X \in \mathbb{R}^{n\times n} \mapsto (X,X)_F^{1/2} =: \|X\|_F \tag{2.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Formally, we are interested in finding some approximation M to the solution of

$$AX = I_n \tag{2.3}$$

or, alternatively, we seek to minimize the objective function $f : \mathbb{R}^{n\times n} \to [0,\infty)$ given by

$$f(M) := \|I_n - AM\|_F^2, \tag{2.4}$$

that is, the Frobenius norm of the residual of Eq. (2.3) for an approximation M of $X = A^{-1}$. Although Eq. (2.4) famously decomposes into n independent 2-norms of vector residuals (e.g., see Section 10.5 in [19]) which allows for embarrassingly parallelizable strategies, in this work, we are exclusively concerned with global iterations. That is, the matrix M is considered as a whole, instead of a set of n unrelated columns.

In this section, we present one-dimensional projection methods aimed at minimizing Eq. (2.4). One particular set of such methods is that of descent algorithms, which we define as follows. Given an iterate $M_i \in \mathbb{R}^{n\times n}$ of right-approximate inverse of A, a descent method defines a new iterate $M_{i+1} \in \mathbb{R}^{n\times n}$ as follows for a given search direction $P_i \in \mathbb{R}^{n\times n}$:

$$M_{i+1} \in M_i + \mathrm{span}\{P_i\} \;\text{ s.t. }\; R_{i+1} := I_n - AM_{i+1} \perp A\,\mathrm{span}\{P_i\} \;\text{ for }\; i = 0,1,\ldots \tag{2.5}$$

where the orthogonality is stated in terms of the Frobenius inner product. As long as $AP_i \neq 0_{n\times n}$, the application of the Petrov-Galerkin condition of Eq. (2.5) leads to the following update formula:

$$M_{i+1} = M_i + \alpha_i P_i \;\text{ where }\; \alpha_i = \frac{(R_i, AP_i)_F}{\|AP_i\|_F^2} \;\text{ for }\; i = 0,1,\ldots. \tag{2.6}$$

A standard result is that the iterate M_{i+1} is defined by Eqs. (2.5)-(2.6) if and only if the following holds:

$$M_{i+1} = \arg \min_{M \in M_i + \mathrm{span}\{P_i\}} \|I_n - AM\|_F. \tag{2.7}$$

Thus, in a sense, M_{i+1} is optimal over the affine search space $M_i + \text{span}\{P_i\}$. Moreover, the residual norm of an iterate from coordinate descent is non-increasing, i.e., we have $\|R_{i+1}\|_F \leq \|R_i\|_F$. Indeed,

$$\|R_{i+1}\|_F = \|I_n - AM_{i+1}\|_F = \|I_n - AM_i - \alpha_i AP_i\|_F \leq \|I_n - AM_i\|_F + |\alpha_i| \cdot \|AP_i\|_F \leq \|R_i\|_F.$$

Hence, the residual norm being non-increasing and bounded below by zero, coordinate descent algorithms converge to stationary points of the minimized objective function in Eq. (2.4). Moreover, the objective function in Eq. (2.4) being convex, convergence happens toward the unique global minimum $M = A^{-1}$. Given an initial right-approximate inverse $M_0 \in \mathbb{R}^{n\times n}$ of A, the update formula in Eq. (2.6) is at the heart of global descent iterations. Different variants of global descent methods are obtained depending on how the search direction P_i is defined. In what follows, we briefly review two existing variants, namely the minimal residual algorithm and the steepest descent method.

2.1. *Minimal residual method*

The simplest global descent method is obtained by letting the search direction P_i be given by the residual R_i:

$$P_i := R_i = I_n - AM_i. \tag{2.8}$$

The resulting method, referred to as the minimal residual (MR) algorithm, is described in Algo. 1.

Algorithm 1 MR(A, M_0)

$R_0 := I_n - AM_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := (R_i, AR_i)_F / \|AR_i\|_F^2$
  $M_{i+1} := M_i + \alpha_i R_i$
  $R_{i+1} := R_i - \alpha_i AR_i$
end for

2.2. *Steepest descent method*

As shown in Section 10.5.2 of [19], and given the symmetry of A, the gradient of the objective function Eq. (2.4) is given by

$$\nabla_M f(M) = -2AR \text{ where } R := I_n - AM. \tag{2.9}$$

From hereon, we find it useful to introduce the so-called gradient direction:

$$G_i := -AR_i. \tag{2.10}$$

Then, the steepest descent (SD) method consists of letting the search direction be opposite to the gradient direction:

$$P_i := -G_i \text{ for } i = 0, 1, \ldots. \tag{2.11}$$

The resulting procedure is given in Algo. 2.

Algorithm 2 SD(A, M_0)

$R_0 := I_n - AM_0$
$P_0 := AR_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_0 := (R_i, AP_i)_F / \|AP_i\|_F^2$
  $M_{i+1} := M_i + \alpha_i P_i$
  $R_{i+1} := R_i - \alpha_i AP_i$
  $P_{i+1} := AR_{i+1}$
end for

3. Nonlinear conjugate gradient method

Both the MR and SD methods suffer from slow convergence behaviors. The latter often more so than the former, see [10]. In an attempt to accelerate the convergence of the SD method, we define and derive a nonlinear conjugate gradient (NCG, see Section 5.2 in [21]) algorithm for the minimization of Eq. (2.4) by enforcing an orthogonality condition on the search directions, which, as for the SD method, are also defined along the gradient given by Eq. (2.9). As a result, we will show that, despite both methods having search directions defined along the gradient of Eq. (2.4), unlike the SD method, the NCG iterates are globally optimal. That is, the NCG iterates minimize Eq. (2.4) over an affine subspace spanned by all the previously generated search direction iterates. The NCG method is formally introduced in Definition 1, and the corresponding update formulae are given in Theorem 1.

Definition 1 (NCG method) *Given an SPD matrix $A \in \mathbb{R}^{n\times n}$ with a right-approximate inverse $M_0 \in \mathbb{R}^{n\times n}$, a sequence of NCG iterates of right-approximate inverses of A is defined by*

$$M_{i+1} \in M_i + span\{P_i\} \;\; s.t. \;\; R_{i+1} := I_n - AM_{i+1} \perp span\{P_i\} \;\; for \;\; i = 0, 1, \ldots, \tag{3.1}$$

where $P_i \in \mathbb{R}^{n\times n}$ is a search direction iterate defined as

$$P_i \in -G_i + span\{P_{i-1}\} \;\; s.t. \;\; P_i \perp A\, span\{P_{i-1}\} \;\; for \;\; i = 1, 2, \ldots \tag{3.2}$$

with $P_0 := -G_0$ and $G_i := -AR_i$, where G_i denotes the gradient direction of $f(M)$ at M_i.

Theorem 1 (NCG iterates) *The iterates of the NCG method (Definition 1) are given by*

$$M_{i+1} := M_i + \alpha_i P_i \;\; where \;\; \alpha_i := -\frac{(R_i, G_i)_F}{(P_i, AP_i)_F} \;\; and \;\; G_i := -AR_i \;\; for \;\; i = 0, 1, \ldots, \tag{3.3}$$

in which the search direction is updated as follows:

$$P_i := -G_i + \beta_i P_{i-1} \;\; where \;\; \beta_i := \frac{(R_i, G_i)_F}{(R_{i-1}, G_{i-1})_F} \;\; for \;\; i = 1, 2, \ldots. \tag{3.4}$$

Then, the residuals and gradient directions are orthogonal. That is:

$$(R_{i+1}, G_j)_F = 0 \;\; for \;\; j = 0, 1, \ldots, i. \tag{3.5}$$

The residuals are also orthogonal to the search directions:

$$(R_{i+1}, P_j)_F = 0 \text{ for } j = 0, 1, \ldots, i, \tag{3.6}$$

and the search directions are A-orthogonal:

$$(P_{i+1}, AP_j)_F = 0 \text{ for } j = 0, 1, \ldots, i. \tag{3.7}$$

Proof of Theorem 1 From Eq. (3.1), we have that the update formula for the right-approximate inverse is given by

$$M_{i+1} := M_i + \alpha_i P_i \text{ for } i = 0, 1, \ldots \tag{3.8}$$

where $\alpha_i \in \mathbb{R}$ is a step size which indicates how far along the search direction the new iterate is set away from the previous iterate. Another useful update formula is that of the residual:

$$R_{i+1} := R_i - \alpha_i A P_i \text{ for } i = 0, 1, \ldots. \tag{3.9}$$

From Eq. (3.2), we have that the update formula for the search direction is given by

$$P_i := -G_i + \beta_i P_{i-1} \text{ for } i = 1, 2, \ldots \text{ with } P_0 := -G_0, \tag{3.10}$$

where $\beta_i \in \mathbb{R}$ is a step size which indicates how far along the previous direction the new search direction is set away from $-G_i$.

In what follows, we find expressions for the step sizes, namely α_i in (1), and β_i in (2), thereby proving Eqs. (3.3) and (3.4). Then, the orthogonality properties stated in Eqs. (3.5), (3.6) and (3.7) are proved by induction. First, in (3), we show that these properties hold for the base case $i = 0$. Then, the induction hypothesis is made that

$$(R_i, G_j)_F = 0, \ (R_i, P_j)_F = 0 \text{ and } (P_i, AP_j)_F = 0 \text{ for } j = 0, 1, \ldots, i-1. \tag{3.11}$$

In (4), we then show that this hypothesis implies Eq. (3.5), while Eqs. (3.6) and (3.7) are shown to hold in (5) and (6), respectively.

(1) **Proof of Eq.** (3.3).
From the orthogonality condition of Eq. (3.1), along with the update formulae in Eqs. (3.10) and (3.9) as well as the symmetry of A, we get:

$$\begin{aligned}
(R_{i+1}, P_i)_F &= 0 \\
(R_{i+1}, -G_i + \beta_i P_{i-1})_F &= 0 \\
-(R_{i+1}, G_i)_F + \beta_i (R_{i+1}, P_{i-1})_F &= 0 \\
-(R_{i+1}, G_i)_F + \beta_i (R_i - \alpha_i A P_i, P_{i-1})_F &= 0 \\
-(R_{i+1}, G_i)_F + \beta_i (R_i, P_{i-1})_F - \beta_i \alpha_i (AP_i, P_{i-1})_F &= 0 \\
-(R_{i+1}, G_i)_F + \beta_i \cancelto{0}{(R_i, P_{i-1})_F} - \beta_i \alpha_i \cancelto{0}{(P_i, AP_{i-1})_F} &= 0.
\end{aligned}$$

(0 by Eq. (3.1); 0 by Eq. (3.2))

Therefore, we have

$$(R_{i+1}, G_i)_F = 0 \text{ for } i = 0, 1, \ldots \tag{3.12}$$

which, once combined with the residual update formula in Eq. (3.9), leads to

$$\begin{aligned}(R_i - \alpha_i A P_i, G_i)_F &= 0\\ (R_i, G_i)_F - \alpha_i (A P_i, G_i)_F &= 0\end{aligned}$$

so that the step size α_i is given by:

$$\alpha_i = \frac{(R_i, G_i)_F}{(A P_i, G_i)_F} \text{ for } i = 0, 1, \ldots. \tag{3.13}$$

Using the update formula for the search direction in Eq. (3.10), the denominator of Eq. (3.13) can be recast into

$$\begin{aligned}(A P_i, G_i)_F &= (A P_i, -P_i + \beta_i P_{i-1})_F\\ &= -(A P_i, P_i)_F + \beta_i (A P_i, P_{i-1})_F\\ &= -(A P_i, P_i)_F + \beta_i \underbrace{(P_i, A P_{i-1})_F}_{0 \text{ by Eq. (3.2)}}\end{aligned}$$

so that Eq. (3.13) becomes

$$\alpha_i = -\frac{(R_i, G_i)_F}{(A P_i, P_i)_F} = -\frac{(R_i, G_i)_F}{(P_i, A P_i)_F} \text{ for } i = 0, 1, \ldots \tag{3.14}$$

where use is made of A's symmetry. □

(2) **Proof of Eq.** (3.4).
The search directions P_i and P_{i-1} are made A-orthogonal by construction, i.e., see the Petrov-Galerkin condition in Eq. (3.2). From that orthogonality condition, we obtain:

$$\begin{aligned}(P_i, A P_{i-1})_F &= 0\\ (-G_i + \beta_i P_{i-1}, A P_{i-1})_F &= 0\\ -(G_i, A P_{i-1})_F + \beta_i (P_{i-1}, A P_{i-1})_F &= 0\end{aligned}$$

so that a first formula for the step size β_i is given by:

$$\beta_i = \frac{(G_i, A P_{i-1})_F}{(P_{i-1}, A P_{i-1})_F} \text{ for } i = 1, 2, \ldots. \tag{3.15}$$

From the residual update formula in Eq. (3.9), we have

$$A P_{i-1} = \frac{R_{i-1} - R_i}{\alpha_{i-1}},$$

which is used as follows in the numerator of Eq. (3.15):

$$\begin{aligned}(G_i, AP_{i-1})_F &= \frac{(G_i, R_{i-1} - R_i)_F}{\alpha_{i-1}} \\ &= \frac{(G_i, R_{i-1})_F}{\alpha_{i-1}} - \frac{(G_i, R_i)_F}{\alpha_{i-1}} \\ &= \frac{\cancel{(R_i, G_{i-1})_F}}{\cancel{\alpha_{i-1}}} - \frac{(G_i, R_i)_F}{\alpha_{i-1}} \\ &\qquad \text{0 by Eq. (3.12)} \\ &= -\frac{(G_i, R_i)_F}{\alpha_{i-1}}.\end{aligned}$$

Combining this last result with the expression for the step sizes β_i and α_i in Eqs. (3.15) and (3.14), respectively, we get:

$$\begin{aligned}\beta_i &= \frac{(G_i, AP_{i-1})_F}{(P_{i-1}, AP_{i-1})_F} \\ &= -\frac{1}{\alpha_{i-1}} \frac{(G_i, R_i)_F}{(P_{i-1}, AP_{i-1})_F} \\ &= \frac{(G_i, R_i)_F}{(G_{i-1}, R_{i-1})_F} \\ &= \frac{(R_i, G_i)_F}{(R_{i-1}, G_{i-1})_F}. \quad \square\end{aligned}$$

(3) **Proof of base cases of Eqs. (3.5), (3.6) and (3.7).**
First, we prove the base case of Eq. (3.5) using the residual update formula in Eq. (3.9) along with the expression for α_i in Eq. (3.13):

$$\begin{aligned}(R_1, G_0)_F &= (R_0 - \alpha_0 AP_0, G_0)_F \\ &= (R_0, G_0)_F - \alpha_0 (AP_0, G_0)_F \\ &= (R_0, G_0)_F - \frac{(R_0, G_0)_F}{(AP_0, G_0)_F} (AP_0, G_0)_F \\ &= 0. \quad \square\end{aligned}$$

We need nothing more than the definition of the initial search direction, i.e., $P_0 := -G_0$, along with the orthogonality property we just showed between R_1 and G_0, to prove the base case of Eq. (3.6):

$$(R_1, P_0)_F = -(R_1, G_0)_F = 0. \quad \square$$

Finally, the base case of Eq. (3.7) is proved using the update formula for search directions from Eq. (3.4) along with the expression for the step size β_i in Eq. (3.15):

$$\begin{aligned}(P_1, AP_0)_F &= (-G_1 + \beta_1 P_0, AP_0)_F \\ &= -(G_1, AP_0)_F + \beta_1 (P_0, AP_0)_F \\ &= -(G_1, AP_0)_F + \frac{(G_1, AP_0)_F}{(P_0, AP_0)_F} (P_0, AP_0)_F \\ &= 0. \quad \square\end{aligned}$$

(4) **Proof of Eq.** (3.5).
Using the residual update formula in Eq. (3.9), we get

$$\begin{aligned}(R_{i+1}, G_j)_F &= (R_i - \alpha_i AP_i, G_j)_F \\ &= (R_i, G_j)_F - \alpha_i (AP_i, G_j)_F,\end{aligned}$$

in which we then use the update formula for search directions given by Eq. (3.10), so that

$$\begin{aligned}(R_{i+1}, G_j)_F &= (R_i, G_j)_F - \alpha_i (AP_i, \beta_j P_{j-1} - P_j)_F \\ &= (R_i, G_j)_F - \alpha_i \beta_j (AP_i, P_{j-1})_F + \alpha_i (AP_i, P_j)_F.\end{aligned}$$

If we make use of A's symmetry, this yields

$$(R_{i+1}, G_j)_F = (R_i, G_j)_F - \alpha_i \beta_j \underbrace{(P_i, AP_{j-1})_F}_{0 \text{ by Eq. (3.11)}} + \alpha_i (P_i, AP_j)_F \quad \text{for} \quad j = 0, 1, \ldots, i, \tag{3.16}$$

where the middle term of the right-hand side cancels out due to the induction hypothesis, leaving us with

$$(R_{i+1}, G_j)_F = (R_i, G_j)_F + \alpha_i (P_i, AP_j)_F \quad \text{for} \quad j = 0, 1, \ldots, i. \tag{3.17}$$

Here, we distinguish between two cases. First, for $j = i$, orthogonality was already proven, as stated by Eq. (3.12). Carrying Eq. (3.12) within Eq. (3.17) yields the expression for the step size α_i given by Eq. (3.14). Then, for $j < i$, the induction hypothesis implies both $(R_i, G_j)_F = 0$ and $(P_i, AP_j)_F = 0$ for $j = 0, 1, \ldots, i-1$. Thus, in combination with Eq. (3.12), we can now state

$$(R_{i+1}, G_j)_F = 0 \quad \text{for} \quad j = 0, 1, \ldots, i. \quad \square$$

(5) **Proof of Eq.** (3.6).
Making use of the residual update formula in Eq. (3.9), we get

$$\begin{aligned}(R_{i+1}, P_j)_F &= (R_i - \alpha_i AP_i, P_j)_F \\ &= (R_i, P_j)_F - \alpha_i (AP_i, P_j)_F\end{aligned}$$

so that, due to the symmetry of A, we have

$$(R_{i+1}, P_j)_F = (R_i, P_j)_F - \alpha_i (P_i, AP_j)_F \quad \text{for} \quad j = 0, 1, \ldots, i. \tag{3.18}$$

Once again, we distinguish between two cases. First, for $j = i$, due to the orthogonality condition in Eq. (3.1), Eq. (3.18) yields

$$\alpha_i = \frac{(R_i, P_i)_F}{(P_i, AP_i)_F}, \tag{3.19}$$

which is yet another valid expression for the step size α_i. Then, for $j < i$, $(R_i, P_j)_F$ and $(P_i, AP_j)_F$ both cancels out due the induction hypothesis, leading to

$$(R_{i+1}, P_j)_F = 0 \;\text{ for }\; j = 0, 1, \ldots, i. \quad \square$$

(6) **Proof of Eq.** (3.7).
Using the update formulae for search directions and residuals given by Eqs. (3.10) and (3.9), respectively, we obtain

$$\begin{aligned}
(P_{i+1}, AP_j)_F &= (-G_{i+1} + \beta_{i+1} P_i, AP_j)_F \\
&= -(G_{i+1}, AP_j)_F + \beta_{i+1}(P_i, AP_j)_F \\
&= -\frac{(G_{i+1}, R_j - R_{j+1})_F}{\alpha_j} + \beta_{i+1}(P_i, AP_j)_F \\
&= -\frac{(G_{i+1}, R_j)_F}{\alpha_j} + \frac{(G_{i+1}, R_{j+1})_F}{\alpha_j} + \beta_{i+1}(P_i, AP_j)_F.
\end{aligned}$$

From the definition of $G_{i+1} := -AR_{i+1}$ and A's symmetry, we get

$$(P_{i+1}, AP_j)_F = -\frac{\cancelto{0 \text{ by Eq. (3.5)}}{(R_{i+1}, G_j)_F}}{\alpha_j} + \frac{(R_{i+1}, G_{j+1})_F}{\alpha_j} + \beta_{i+1}(P_i, AP_j)_F \;\text{ for }\; j = 0, 1, \ldots, i.$$

Note that the A-orthogonality holds by construction for $j = i$. Then, for $j < i$, (R_{i+1}, G_{j+1}) cancels out by Eq. (3.5), and $(P_i, AP_j)_F$ does so as well, due to the induction hypothesis. Therefore, we do have

$$(P_{i+1}, AP_j)_F = 0 \;\text{ for }\; j = 0, 1, \ldots, i. \quad \square$$

Proposition 2 (NCG as a Krylov subspace method) *The iterates of the NCG method (Definition 1) are equivalently defined as*

$$M_i \in M_0 + \mathscr{K}_i(A^2, G_0) \;\; s.t. \;\; R_i := I_n - AM_i \perp \mathscr{K}_i(A^2, G_0) \;\; for \;\; i = 1, 2, \ldots \tag{3.20}$$

where $G_0 := -AR_0$, and $\mathscr{K}_i(A^2, G_0)$ denotes the Krylov subspace of A^2 generated by the initial gradient G_0:

$$\mathscr{K}_i(A^2, G_0) := span\{G_0, A^2 G_0, \ldots, A^{2(i-1)} G_0\}. \tag{3.21}$$

Moreover, the search directions $P_0, \ldots, P_{i-1}$ constitute an A-orthogonal basis of this Krylov subspace, which is equally spanned by the gradient directions $G_0, \ldots, G_{i-1}$:

$$span\{P_0, P_1, \ldots, P_{i-1}\} = span\{G_0, G_1, \ldots, G_{i-1}\} = \mathscr{K}_i(A^2, G_0). \tag{3.22}$$

Proof of Proposition 2 In (1), we prove Eq. (3.22). In (2), we show that the NCG iterate of Definition 1 is an orthogonal projection in the affine span of search directions.

(1) **Proof of Eq.** (3.22).
In (1.1), we prove that

$$G_i, P_i \in \mathscr{K}_{i+1}(A^2, G_0), \text{ for } i = 0, 1, \ldots \tag{3.23}$$

which, when combined with the linear independence of gradient and search direction iterates, implies

$$\text{span}\{G_0, \ldots, G_{i-1}\} \subseteq \mathscr{K}_i(A^2, G_0) \tag{3.24}$$

$$\text{span}\{P_0, \ldots, P_{i-1}\} \subseteq \mathscr{K}_i(A^2, G_0). \tag{3.25}$$

Then, in (1.2), we show that

$$\text{span}\{G_0, \ldots, G_{i-1}\} \supseteq \mathscr{K}_i(A^2, G_0) \tag{3.26}$$

and

$$\text{span}\{P_0, \ldots, P_{i-1}\} \supseteq \mathscr{K}_i(A^2, G_0). \tag{3.27}$$

We recall that the A-orthogonality of search directions was already proven in Theorem 1. Thus, this completes the Proof of Eq. (3.22). □

(1.1) **Proof of Eq.** (3.23).
The proof is made by induction. The base case of Eq. (3.23) is trivially satisfied:

$$\mathscr{K}_1(A^2, G_0) = \text{span}\{G_0\} \ni G_0, P_0 := -G_0. \tag{3.28}$$

Eq. (3.23) is assumed to hold for i. Then, from the residual update formula in Eq. (3.9), we obtain the following gradient update formula:

$$\begin{aligned} G_{i+1} &= -AR_{i+1} \\ &= -A(R_i - \alpha_i A P_i) \\ &= G_i + \alpha_i A^2 P_i. \end{aligned} \tag{3.29}$$

The induction hypothesis in eq. (3.23) suggests

$$G_i \in \mathscr{K}_{i+1}(A^2, G_0) \subseteq K_{i+2}(A^2, G_0) \tag{3.30}$$

and

$$P_i \in \mathscr{K}_{i+1}(A^2, G_0) \implies A^2 P_i \in A^2 \mathscr{K}_{i+1}(A^2, G_0) \subseteq \mathscr{K}_{i+2}(A^2, G_0). \tag{3.31}$$

Combining this last statement with Eqs. (3.29) and (3.10), we get $G_{i+1}, P_{i+1} \in \mathscr{K}_{i+2}(A^2, G_0)$. □

(1.2) **Proof of Eqs.** (3.26) **and** (3.27).

This proof is also made by induction. The base cases of Eqs. (3.26) and (3.27) are covered by Eq. (3.28). Then, let us assume that Eq. (3.27) holds, so that

$$A^{2i}G_0 = A^2(A^{2(i-1)}G_0) \in \text{span}\{A^2P_0, \ldots, A^2P_{i-1}\}. \tag{3.32}$$

From the gradient update formula in Eq. (3.29), we have

$$A^2P_j = \frac{G_{j+1} - G_j}{\alpha_j} \text{ for } j = 0, 1, \ldots, i-1. \tag{3.33}$$

Combining Eqs. (3.32) and (3.33), we obtain

$$A^{2i}G_0 \in \text{span}\{G_0, \ldots, G_i\}. \tag{3.34}$$

Assuming that Eq. (3.26) holds, its combination with Eq. (3.34) yields

$$\text{span}\{G_0, \ldots, G_i\} \supseteq \text{span}\{G_0, A^2G_0, \ldots, A^{2i}G_0\} = \mathscr{K}_{i+1}(A^2, G_0). \tag{3.35}$$

This completes the proof of Eq. (3.26).
The second part of the proof goes as follows:

$$\begin{aligned}
\text{span}\{P_0, P_1, \ldots, P_{i-1}, P_i\} &\underset{\text{by Eq. (3.10)}}{\supseteq} \text{span}\{P_0, P_1, \ldots, P_{i-1}, G_i\} \\
&\underset{\text{by Eq. (3.27)}}{\supseteq} \text{span}\{G_0, A^2G_0, \ldots, A^{2(i-1)}G_0, G_i\} \\
&\underset{\text{by Eq. (3.24)}}{\supseteq} \text{span}\{G_0, G_1, \ldots, G_{i-1}, G_i\} \\
&\underset{\text{by Eq. (3.35)}}{\supseteq} \text{span}\{G_0, A^2G_0, \ldots, A^{2(i-1)}G_0, A^{2i}G_0\}
\end{aligned}$$

so that

$$\text{span}\{P_0, \ldots, P_i\} \supseteq \mathscr{K}_{i+1}(A^2, G_0). \quad \square \tag{3.36}$$

(2) **Proof that the NCG iterate of Definition 1 is an orthogonal projection in the affine span of search directions**.
From Eq. (3.3), it is clear that

$$M_i = M_0 + \sum_{j=0}^{i-1} \alpha_j P_j \in M_0 + \text{span}\{P_0, \ldots, P_{i-1}\}. \tag{3.37}$$

Then, from Eq. (3.6), we have

$$(R_i, P_j)_F = 0 \text{ for } j = 0, 1, \ldots, i-1 \tag{3.38}$$

so that

$$R_i := I_n - AM_i \perp \text{span}\{P_0, \ldots, P_{i-1}\}. \quad \square \tag{3.39}$$

Theorem 3 (Optimality of NCG iterates) *The NCG iterates are given by*

$$M_i \in M_0 + \mathscr{K}_i(A^2, G_0) \;\; s.t. \;\; R_i := I_n - AM_i \perp \mathscr{K}(A^2, G_0) \;\; for \;\; i = 1, 2, \ldots \tag{3.40}$$

if and only if the right-approximate inverse M_i is optimal in the sense that

$$\|A^{-1} - M_i\|_{F,A} = \min_{M \in M_0 + \mathscr{K}_i(A^2, G_0)} \|A^{-1} - M\|_{F,A} \;\; for \;\; i = 1, 2, \ldots. \tag{3.41}$$

That is, the NCG iterate minimizes the Frobenius A-norm of the error $E := A^{-1} - M$ over the affine Krylov subspace of A^2 generated by the initial gradient $G_0 := -AR_0$.

Proof of Theorem 3 First, in (1), we prove that NCG iterates given by Eq. (3.40) are globally optimal as stated by Eq. (3.41). Then we show the converse in (2).

(1) **Proof that NCG iterates given by Eq. (3.40) are globally optimal as per Eq. (3.41).**
Let us define

$$\Delta M_i := \sum_{j=0}^{i-1} \alpha_j P_j \in \mathscr{K}_j(A^2, G_0) \tag{3.42}$$

so that $M_i = M_0 + \Delta M_i$. Then, the orthogonality condition of Eq. (3.40) is recast as follows:

$$\begin{aligned} R_i &\perp \mathscr{K}_i(A^2, G_0) \\ I_n - AM_i &\perp \mathscr{K}_i(A^2, G_0) \\ A(A^{-1} - M_i) &\perp \mathscr{K}_i(A^2, G_0) \\ A(A^{-1} - (M_0 + \Delta M_i)) &\perp \mathscr{K}_i(A^2, G_0). \end{aligned}$$

From hereon, we denote the forward error of the i-th right-approximate inverse iterate by

$$E_i := A^{-1} - M_i \tag{3.43}$$

so that we have

$$AE_i \perp \mathscr{K}_i(A^2, G_0) \tag{3.44}$$

$$A(E_0 - \Delta M_i) \perp \mathscr{K}_i(A^2, G_0). \tag{3.45}$$

Let us then introduce $\perp_A$ to denote orthogonality with respect to the Frobenius inner product weighted by A, that is,

$$X \perp_A Y \iff (X, Y)_{F,A} = 0 \tag{3.46}$$

where

$$(X, Y) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \mapsto (X, Y)_{F,A} := (AX, Y)_F \tag{3.47}$$

is what we call the Frobenius inner product weighted by A, with the induced norm

$$X \in \mathbb{R}^{n\times n} \mapsto \|X\|_{F,A} := (X,X)_{F,A}^{1/2}. \tag{3.48}$$

Then, the orthogonality condition stated in Eqs. (3.44)-(3.45) reads

$$E_i \perp_A \mathscr{K}_i(A^2, G_0) \tag{3.49}$$

$$E_0 - \Delta M_i \perp_A \mathscr{K}_i(A^2, G_0). \tag{3.50}$$

Consequently, by the orthogonal projection theorem, we have

$$\|E_i\|_{F,A} = \|E_0 - \Delta M_i^{-1}\|_{F,A} = \min_{\Delta M \in \mathscr{K}_i(A^2,G_0)} \|E_0 - \Delta M\|_{F,A} \tag{3.51}$$

which we recast as follows:

$$\begin{aligned}\|E_i\|_{F,A} &= \min_{\Delta M \in \mathscr{K}_i(A^2,G_0)} \|A^{-1} - M_0 - \Delta M\|_{F,A} \\ &= \min_{M \in M_0 + \mathscr{K}_i(A^2,G_0)} \|A^{-1} - M\|_{F,A}. \quad \square\end{aligned}$$

(2) **Proof that globally optimal iterates produce orthogonal residuals**.
The optimality of M_i stated in Eq. (3.41) may be recast as follows:

$$\begin{aligned}\|A^{-1} - M_i\|_{F,A} &= \min_{M \in M_0 + \mathscr{K}_i(A^2,G_0)} \|A^{-1} - M\|_{F,A} \\ &= \min_{\Delta M \in \mathscr{K}_i(A^2,G_0)} \|A^{-1} - (M_0 + \Delta M)\|_{F,A} \\ &= \min_{\Delta M \in \mathscr{K}_i(A^2,G_0)} \|E_0 - \Delta M\|_{F,A}.\end{aligned}$$

Then, from the orthogonal projection theorem, we have

$$\begin{aligned} E_0 - \Delta M_i &\perp_A \mathscr{K}_i(A^2, G_0) \\ A^{-1} - (M_0 + \Delta M_i) &\perp_A \mathscr{K}_i(A^2, G_0) \\ A^{-1} - M_i &\perp_A \mathscr{K}_i(A^2, G_0) \\ I_n - AM_i &\perp \mathscr{K}_i(A^2, G_0) \\ R_i &\perp \mathscr{K}_i(A^2, G_0). \quad \square \end{aligned}$$

Theorem 4 (Convergence of NCG iterates) *Let A have $k \leq n$ distinct eigenvalues. Then, the NCG iterates (Definition 1) converge to A^{-1} in no more than k iterations. Moreover, the Frobenius A-norm of the error is bounded as follows:*

$$\frac{\|A^{-1} - M_i\|_{F,A}}{\|A^{-1} - M_0\|_{F,A}} \leq 2\left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^i \quad \text{for } i = 0,1,\ldots \tag{3.52}$$

where $\kappa(A) = \lambda_{max}(A)/\lambda_{min}(A)$ is the spectral condition number of A.

Proof of Theorem 4 Now that the NCG method (Definition 1) has been been recast into an orthogonal projection onto an affine Krylov subspace (see Proposition 2), the proof of Theorem (4) becomes highly derivative of state-of-the-art results on the convergence of standard NCG iterates. Therefore, for the sake brevity, we only sketch a few lines of this proof, highlighting the points which differ from already available results. In (1), we prove the bound in Eq. (3.52), and in (2), we show that NCG iterates converge in k iterations or less, where $k \leq n$ is the number of distinct eigenvalues of A.

(1) **Proof of Eq.** (3.52).
Let $M \in M_0 + \mathscr{K}_i(A^2, G_0)$, then there exists a polynomial $p \in \mathscr{P}_{i-1}$ in the set $\mathscr{P}_{i-1}$ of all real polynomials of degree no greater than $i-1$, such that

$$M = M_0 + p(A^2)G_0. \tag{3.53}$$

The forward error associated to M can then be recast into

$$E := A^{-1} - M = (I_n + A^2 p(A^2))E_0 = q(A^2)E_0 \tag{3.54}$$

where we introduced a new polynomial $q : t \mapsto q(t) := 1 + t \cdot p(t)$ such that $q \in \mathscr{P}_i$ and $q(0) = 1$. Then, we have

$$\|E\|_{F,A}^2 = \|q(A^2)E_0\|_{F,A}^2. \tag{3.55}$$

Since A is SPD, it is diagonalizable, and it admits an eigendecomposition $A = Q\Lambda Q^T$ where Q is orthogonal and Λ is diagonal, with the eigenvalues of A as diagonal components. Clearly, we have $A^2 = Q\Lambda^2 Q^T$ and $q(A^2) = Qq(\Lambda^2)Q^T$, so that we obtain

$$\|E\|_{F,A}^2 = \|q(\Lambda^2)Q^T E_0\|_{F,\Lambda}. \tag{3.56}$$

In particular, as per Theorem 2, using Eq. (3.56), we can show that the CG iterate is such that

$$\begin{aligned}\|E_i\|_{F,A}^2 := \|A^{-1} - M_i^{-1}\|_{F,A}^2 &= \min_{\substack{q \in \mathscr{P}_i \\ q(0)=1}} \|q(\Lambda^2)Q^T E_0\|_{F,\Lambda}^2 \\ &\leq \min_{\substack{q \in \mathscr{P}_i \\ q(0)=1}} \max_{1 \leq k \leq n} \left\{ q\left(\lambda_k(A)^2\right)^2 \right\} \|E_0\|_{F,A}^2\end{aligned}$$

where we made use of the optimality property over matrix polynomials of orthogonal projections on affine Krylov subspaces. We then have the following sharp bound:

$$\frac{\|A^{-1} - M_i\|_{F,A}^2}{\|A^{-1} - M_0\|_{F,A}^2} \leq \min_{\substack{q \in \mathscr{P}_i \\ q(0)=1}} \max_{1 \leq k \leq n} \left\{ q\left(\lambda_k(A)^2\right)^2 \right\}. \tag{3.57}$$

Building on standard results such as the proofs of Theorems 6.27 and 6.28 in [1], we obtain

$$\min_{\substack{q \in \mathscr{P}_i \\ q(0)=1}} \max_{1 \leq k \leq n} \left\{ q\left(\lambda_k(A)^2\right)^2 \right\} \leq 2 \left(\frac{\lambda_{max}(A)/\lambda_{min}(A) - 1}{\lambda_{max}(A)/\lambda_{min}(A) + 1} \right)^i. \quad \square \tag{3.58}$$

(2) **Proof of convergence in no more that $k \leq n$ iterations**.
From Eq. (3.55) along with the optimality property over matrix polynomials of orthogonal projections on affine Krylov subspaces, for every CG iterate M_i^{-1}, we have

$$\|A^{-1} - M_i\|_{F,A} =: \|E_i\|_{F,A}^2 = \min_{\substack{q \in \mathscr{P}_i \\ q(0)=1}} \|q(A^2)E_0\|_{F,A}^2. \tag{3.59}$$

But since A has k distinct eigenvalues, so does A^2, and the minimum polynomial q_{min} of A^2 has degree k. Thus, we have

$$\|E_k\|_{F,A} = \min_{\substack{q \in \mathscr{P}_k \\ q(0)=1}} \|q(A^2)E_0\|_{F,A}^2 \leq \|q_{min}(A^2)E_0\|_{F,A}^2 = 0. \quad \square \tag{3.60}$$

Algorithm 3 NCG(A, M_0)

$R_0 := I_n - AM_0$
$G_0 := -AR_0$
$P_0 = -G_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := -(R_i, G_i)_F / (P_i, AP_i)_F$
  $M_{i+1} := M_i + \alpha_i P_i$
  $R_{i+1} := R_i - \alpha_i A P_i$
  $G_{i+1} := -AR_{i+1}$
  $\beta_{i+1} := (R_{i+1}, G_{i+1})_F / (R_i, G_i)_F$
  $P_{i+1} := -G_{i+1} + \beta_{i+1} P_i$
end for

4. Conjugate gradient method

In this Section, we present and summarize properties of the conjugate gradient (CG) method for the computation of SPAIs. Similarly as NCG renders the SD method globally optimal over a Krylov subspace of A^2 upon A-orthogonalizing the search directions, the CG algorithm renders the MR method globally optimal over a Krylov subspace of A. The CG method is defined in Definition 2, the corresponding iterates are given in Proposition 5, and the associated procedure is described in Algo. 3.

Definition 2 (CG method) *Given an SPD matrix $A \in \mathbb{R}^{n\times n}$ with a right-approximate inverse $M_0 \in \mathbb{R}^{n\times n}$, a sequence of CG iterates of right-approximate inverses of A is defined by*

$$M_{i+1} \in M_i + span\{P_i\} \;\; s.t. \;\; R_{i+1} := I_n - AM_{i+1} \perp span\{P_i\} \;\; for \;\; i = 0, 1, \ldots, \tag{4.1}$$

where $P_i \in \mathbb{R}^{n\times n}$ is a search direction iterate defined as

$$P_i \in R_i + span\{P_{i-1}\} \;\; s.t. \;\; P_i \perp A\, span\{P_{i-1}\} \;\; for \;\; i = 1, 2, \ldots \tag{4.2}$$

with $P_0 := R_0$.

Proposition 5 (CG iterates) *The iterates of the CG method (Definition 2) are given by*

$$M_{i+1} := M_i + \alpha_i P_i \;\; where \;\; \alpha_i := \frac{(R_i, R_i)_F}{(P_i, AP_i)_F} \;\; for \;\; i = 0, 1, \ldots, \tag{4.3}$$

in which the search direction is updated as follows:

$$P_i := R_i + \beta_i P_{i-1} \;\; where \;\; \beta_i := \frac{(R_i, R_i)_F}{(R_{i-1}, R_{i-1})_F} \;\; for \;\; i = 1, 2, \ldots. \tag{4.4}$$

Then, the residuals are orthogonal. That is:

$$(R_{i+1}, R_j)_F = 0 \;\; for \;\; j = 0, 1, \ldots, i. \tag{4.5}$$

The residuals are also orthogonal to the search directions:

$$(R_{i+1}, P_j)_F = 0 \ \textit{for} \ \ j = 0, 1, \ldots, i, \tag{4.6}$$

and the search directions are A-orthogonal:

$$(P_{i+1}, AP_j)_F = 0 \ \textit{for} \ \ j = 0, 1, \ldots, i. \tag{4.7}$$

Proof of Proposition 5 Beside the replacement of vector iterates by matrix iterates, and vector dot products by Frobenius inner products, the proof is state-of-the-art. □

Algorithm 4 CG(A, M_0)

$R_0 := I_n - AM_0$
$P_0 = R_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := \|R_i\|_F^2 / (P_i, AP_i)_F$
  $M_{i+1} := M_i + \alpha_i P_i$
  $R_{i+1} := R_i - \alpha_i AP_i$
  $\beta_{i+1} := \|R_{i+1}\|_F^2 / \|R_i\|_F^2$
  $P_{i+1} := R_{i+1} + \beta_{i+1} P_i$
end for

Similarly as with NCG in Section 3, the CG method, although it is formulated in Definition 2 as a one-dimensional projection, is globally optimal over a Krylov subspace of A (see Proposition 6) over which it minimizes the Frobenius A-norm of the error, as stated in Theorem 7. Additionally, the Frobenius A-norm of the error is bounded above as expressed in Eq. (4.13).

Proposition 6 (CG as a Krylov subspace method) *The iterates of the CG method (Definition 2) are equivalently defined as*

$$M_i \in M_0 + \mathscr{K}_i(A, R_0) \;\; s.t. \;\; R_i := I_n - AM_i \perp \mathscr{K}_i(A, R_0) \;\; for \;\; i = 1, 2, \ldots \tag{4.8}$$

where $\mathscr{K}_i(A, R_0)$ denotes the Krylov subspace of A generated by the initial residual:

$$\mathscr{K}_i(A, R_0) := span\{R_0, AR_0, \ldots, A^{i-1}R_0\}. \tag{4.9}$$

Moreover, the search directions $P_0, \ldots, P_{i-1}$ constitute an A-orthogonal basis of this Krylov subspace, which is equally spanned by the residuals:

$$span\{P_0, P_1, \ldots, P_{i-1}\} = span\{R_0, R_1, \ldots, R_{i-1}\} = \mathscr{K}_i(A, R_0). \tag{4.10}$$

Theorem 7 (Optimality of CG iterates) *The CG iterates are given by*

$$M_i \in M_0 + \mathscr{K}_i(A, R_0) \;\; s.t. \;\; R_i \perp \mathscr{K}(A, R_0) \;\; for \;\; i = 1, 2, \ldots \tag{4.11}$$

if and only if the right-approximate inverse M_i is optimal in the sense that

$$\|A^{-1} - M_i\|_{F,A} = \min_{M_i \in M_0 + \mathscr{K}_i(A, R_0)} \|A^{-1} - M_i\|_{F,A} \;\; for \;\; i = 1, 2, \ldots. \tag{4.12}$$

That is, the CG iterate minimizes the Frobenius A-norm of the error $E := A^{-1} - M$ over the affine Krylov subspace of A generated by the initial residual.

Theorem 8 (Convergence of CG iterates) *Let A have $k \leq n$ distinct eigenvalues. Then, the CG iterates (Definition 2) converge to A^{-1} in no more than k iterations. Moreover, the Frobenius A-norm of the error is bounded as follows:*

$$\frac{\|A^{-1} - M_i\|_{F,A}}{\|A^{-1} - M_0\|_{F,A}} \leq \left(\frac{\kappa(A)^{1/2} - 1}{\kappa(A)^{1/2} + 1} \right)^i \;\; for \;\; i = 0, 1, \ldots. \tag{4.13}$$

5. Locally optimal variants

The concept of local optimality was introduced by Andrew Knyazev in [17] as a means to enrich the approximation of eigenpairs produced by CG [16]. The idea stems from the observation that, when applying the CG method to Rayleigh quotient minimization, every new iterate minimizes the Rayleigh quotient over the affine one-dimensional subspace spanned by the current search direction. While the current search direction is a linear combination of the previous residual and search direction, this linear combination remains unchanged when forming the new optimal iterate. Local optimality consists of extending the search space from the one-dimensional span of the current search direction, to the two-dimensional span of the previous residual and search direction.

Here, we introduce local optimality to the case where we minimize the objective function in Eq. (2.4). Standard descent algorithms such as MR and SD form a new iterate M_{i+1} which achieves the minimum given by:

$$\min_{M \in M_i + \text{span}\{P_i\}} \|I_n - AM\|_F. \tag{5.1}$$

We can think of the MR method as a trivial case of

$$P_i \in \text{span}\{R_i, P_{i-1}\}. \tag{5.2}$$

Indeed, MR assumes $P_i = R_i$ for $i = 0, 1, \ldots$. Now, upon letting $P_{-1} := 0_{n\times n}$ and $P_0 := R_0$, we can show, by induction, that

$$\min_{M \in M_i + \text{span}\{R_i, P_{i-1}\}} \|I_n - AM\|_F \leq \min_{M \in M_i + \text{span}\{P_i\}} \|I_n - AM\|_F \quad \text{for } i = 0, 1, \ldots. \tag{5.3}$$

This observation is at the heart of the locally optimal minimal residual (LOMR) method we introduce in Definition 3 with update formulae given in Theorem 9, and which yields Algo. 5. We can then say that LOMR converges faster than MR to $M = A^{-1}$.

Definition 3 (LOMR method) *Given an SPD matrix $A \in \mathbb{R}^{n\times n}$ with a right-approximate inverse $M_0 \in \mathbb{R}^{n\times n}$, a sequence of LOMR iterates of right-approximate inverses of A is defined by*

$$M_{i+1} := \underset{M \in M_i + span\{R_i, P_{i-1}\}}{\arg\min} \|I_n - AM\|_F \ \textit{for } i = 0, 1, \ldots, \tag{5.4}$$

where $P_{-1} := 0_{n\times n}$, $P_0 := R_0$ and $R_i := I_n - AM_i$ for $i = 0, 1, \ldots$.

Theorem 9 *The iterates of the LOMR method (Definition 3) are given by*

$$M_{i+1} := M_i + \delta_i R_i + \gamma_i P_{i-1} \ \textit{for } i = 0, 1, \ldots \tag{5.5}$$

where

$$\delta_i := \begin{cases} (R_i, AR_i)_F / (AR_i, AR_i)_F & \textit{if } i = 0 \\ [(AP_{i-1}, AP_{i-1})_F \cdot (R_i, AR_i)_F - (AR_i, AP_{i-1})_F \cdot (R_i, AP_{i-1})_F]/c_i & \textit{otherwise} \end{cases} \tag{5.6}$$

and

$$\gamma_i := \begin{cases} 1 & \text{if } i = 0 \\ [(AR_i, AR_i)_F \cdot (R_i, AP_{i-1})_F - (AR_i, AP_{i-1})_F \cdot (R_i, AR_i)_F]/c_i & \text{otherwise} \end{cases} \tag{5.7}$$

in which

$$c_i := (AR_i, AR_i)_F \cdot (AP_{i-1}, AP_{i-1})_F - (AR_i, AP_{i-1})_F^2 \text{ for } i = 1, 2, \ldots. \tag{5.8}$$

The residual R_{i+1} and search direction P_i are updated as follows:

$$R_{i+1} := R_i - \delta_i AR_i - \gamma_i AP_{i-1}, \tag{5.9}$$

$$P_i := R_i + (\gamma_i/\delta_i)P_{i-1}. \tag{5.10}$$

Proof of Theorem 9 The optimality condition of Eq. (5.3) may be recast as follows:

$$\begin{aligned} \|R_{i+1}\|_F &= \min_{M \in M_i + \text{span}\{R_i, P_{i-1}\}} \|I_n - AM\|_F \\ &= \min_{\Delta M \in \text{span}\{R_i, P_{i-1}\}} \|R_i - A\Delta M\|_F \\ &= \min_{\Delta R \in A\,\text{span}\{R_i, P_{i-1}\}} \|R_i - \Delta R\|_F. \end{aligned}$$

By the theorem of orthogonal projections, this last orthogonality statement holds if and only if

$$\Delta R \in A\,\text{span}\{R_i, P_{i-1}\} \text{ and } R_i - \Delta R \perp A\,\text{span}\{R_i, P_{i-1}\}, \tag{5.11}$$

so that there exist $\delta_i, \gamma_i \in \mathbb{R}$ such that

$$\begin{cases} (AR_i, R_i - A(\delta_i R_i + \gamma_i P_{i-1}))_F = 0 \\ (AP_{i-1}, R_i - A(\delta_i R_i + \gamma_i P_{i-1}))_F = 0 \end{cases} \tag{5.12}$$

which yields

$$\begin{cases} (AR_i, R_i)_F = \delta_i (AR_i, AR_i)_F + \gamma_i (AR_i, AP_{i-1})_F \\ (AP_{i-1}, R_i)_F = \delta_i (AP_{i-1}, AR_i)_F + \gamma_i (AP_{i-1}, AP_{i-1})_F \end{cases} \tag{5.13}$$

and whose solution is given by Eqs. (5.6), (5.7) and (5.8) to form the update formula given by:

$$M_{i+1} := M_i + \delta_i R_i + \gamma_i P_{i-1} \text{ for } i = 1, 2, \ldots.$$

For $i = 0$, the update formula is the same as that of the MR method, that is:

$$M_1 := M_0 + \frac{(R_0, AR_0)_F}{(AR_0, AR_0)_F} R_0. \tag{5.14}$$

As for the search direction, the combination of Eqs. (5.2) and (5.5) allows for different, but equally valid update formulae of the search direction. In particular, we may assume

$$M_{i+1} = M_i + \delta_i P_i, \tag{5.15}$$

which implies the update formula given by Eq. (5.10). □

Algorithm 5 LOMR(A, M_0)

$R_0 := I_n - AM_0$
$P_{-1} := 0_{n\times n}$
for $i = 0, 1, \ldots$ **do**
  if $i = 0$ **then**
    $\delta_i := (R_i, AR_i,)_F / \|AR_i\|_F^2$
    $\gamma_i := 1$
  else
    $c_i := \|AR_i\|_F^2 \cdot \|AP_{i-1}\|_F^2 - (AR_i, AP_{i-1})_F^2$
    $\delta_i := \left[\|AP_{i-1}\|_F^2 \cdot (R_i, AR_i,)_F - (AR_i, AP_{i-1})_F \cdot (R_i, AP_{i-1})_F\right] / c_i$
    $\gamma_i := \left[\|AR_i\|_F^2 \cdot (R_i, AP_{i-1},)_F - (AR_i, AP_{i-1})_F \cdot (R_i, AR_i)_F\right] / c_i$
  end if
  $M_{i+1} := M_i + \delta_i R_i + \gamma_i P_{i-1}$
  $R_{i+1} := R_i - \delta_i AR_i - \gamma_i AP_{i-1}$
  $P_i := R_i + (\gamma_i / \delta_i) P_{i-1}$
end for

Note that, similarly to how we defined an LOMR method by enriching MR iterations, we could derive a locally optimal version of the SD method. However, as we shall confirm in Section 8.1, the SD method offers poor convergence properties compared to MR, so that it is not worth defining an LOSD method. As for the NCG and CG methods, their global optimality over the affine span of search directions renders these methods locally optimal in the sense stated in Eq. (5.3). Thus, if one were to define locally optimal versions of the NCG or CG method, one would essentially derive expensive procedures equivalent to the standard NCG and CG methods. If you wish to convince yourself of this, we invite the reader to experiment with the lopcg routine made available in the GitHub repository of this paper.

6. Preconditioned iterations

The convergence behavior of the iterative methods presented thus far is known to depend on the spectrum of A. In practice, the number of iterations needed to achieve convergence may be so inconveniently large, that the use of a preconditioner becomes necessary. For that matter, the simplest approach is perhaps that of left-preconditioning, that is, say we are equipped with a non-singular matrix $\Pi \in \mathbb{R}^{n\times n}$ which somehow approximates the inverse of A, and such that $X \in \mathbb{R}^{n\times n} \mapsto \Pi X$ can be computed efficiently. Note that for the approach to remain consistent with our objective to construct a sparse approximate inverse, the preconditioner Π needs also be sparse. Then, instead of solving Eq. (2.3), we search for approximate solutions of

$$\Pi A X = \Pi \tag{6.1}$$

where the coefficient matrix ΠA is tentatively more amenable to efficient iterative solves. Importantly, we note that, although A is SPD, even if Π is SPD, or simply symmetric, the coefficient matrix of Eq. (6.1) is generally not symmetric. While this lack of symmetry or positive-definiteness is not an issue for the MR and SD methods, it does require adjustment for a proper application to the NCG and CG algorithms.

First, we consider the one-dimensional projection methods, namely the MR and SD algorithms. The simplest way to account for preconditioning in those cases is to assemble the coefficient matrix ΠA and, trivially, the right-hand-side Π of Eq. (6.1). Then, we introduce the following changes of variables:

$$A \mapsto \Pi A \quad \text{and} \quad I_n \mapsto \Pi \tag{6.2}$$

within Algos. 1 and 2, where the variable R_i is replaced by Z_i for $i = 0, 1, \ldots$, where Z_i is the preconditioned residual. The benefit of this approach is that, at first glance, the preconditioner needs be applied only once, to A, at the start of the algorithm. The downside is that the norm of the iterate Z_i is that of the preconditioned residual $\Pi(I_n - AM)_i$. Monitoring convergence with respect to $\|Z_i\|_F$ is not always advisable. The most straightforward workaround is simply to evaluate the non-preconditioned residual for the purpose of monitoring convergence. The downside of this is that one needs to store both ΠA and A. The resulting procedure is Algo. 6 for the MR method, and Algo. 7 for the SD algorithm.

Algorithm 6 PMR(A, Π, M_0)

$Z_0 := \Pi - \Pi A M_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := (Z_i, \Pi A Z_i)_F / \|\Pi A Z_i\|_F^2$
  $M_{i+1} := M_i + \alpha_i Z_i$
  $Z_{i+1} := Z_i - \alpha_i \Pi A Z_i$
end for

Algorithm 7 PSD(A, Π, M_0)

$Z_0 := \Pi - \Pi A M_0$
$P_0 := \Pi A Z_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_0 := (Z_i, \Pi A P_i)_F / \|\Pi A P_i\|_F^2$
  $M_{i+1}^{-1} := M_i^{-1} + \alpha_i P_i$
  $Z_{i+1} := Z_i - \alpha_i \Pi A P_i$
  $P_{i+1} := \Pi A Z_{i+1}$
end for

Second, as previously mentioned, the NCG and CG methods may not be applied directly to the left-preconditioned system in Eq. (6.1), or at least, not without loosing their optimality properties. Preconditioning can nevertheless be applied to NCG (and CG) iterations, and we explain how in Proposition 10, with the resulting procedure given in Algo. 8.

Proposition 10 (NPCG method) *Let $\Pi \in \mathbb{R}^{n \times n}$ be an SPD preconditioner for the problem in Eq. (2.3). The NPCG iterates defined by*

$$M_i \in M_0 + \mathscr{K}_i((\Pi A)^2, G_0) \;\; s.t. \;\; R_i := I_n - AM_i \perp \mathscr{K}_i((\Pi A)^2, G_0) \;\; for \;\; i = 1, 2, \ldots \tag{6.3}$$

where $G_0 := -\Pi A Z_0$ and in which $Z_i := \Pi R_i$ denotes the preconditioned residual, have the following update formula:

$$M_{i+1} := M_i + \alpha_i P_i \ \text{ where } \ \alpha_i := -\frac{(R_i, G_i)_F}{(P_i, AP_i)_F} \ \text{ and } \ G_i := -\Pi A Z_i \ \text{ for } \ i = 0, 1, \ldots. \tag{6.4}$$

The update formulae for the search directions (P_i's) and non-preconditioned residuals (R_i's) are still given by Eqs. (3.4) *and* (3.9)*, respectively. Furthermore, the search directions $P_0, \ldots, P_{i-1}$ constitute an A-orthogonal basis of the Krylov subspace, which is equally spanned by the gradient directions $G_0, \ldots, G_{i-1}$:*

$$span\{P_0, P_1, \ldots, P_{i-1}\} = span\{G_0, G_1, \ldots, G_{i-1}\} = \mathscr{K}_i((\Pi A)^2, G_0). \tag{6.5}$$

Proof of Proposition 10 The proof being derivative of known derivations of standard PCG iterations, we only provide a brief description. A being SPD, several equivalent approaches can be considered to derive NPCG iterates. To us, the simplest approach consists of applying the NCG method to the left-preconditioned system given by

$$\Pi A M = \Pi, \tag{6.6}$$

using a Frobenius inner product weighted by Π for the computation of step sizes. This choice of inner product renders ΠA self-adjoint and, through a bit of work, it can be shown that the update formulae for the step sizes remain unaltered by the preconditioning. However, a preconditioned residual $Z_i := \Pi R_i$ needs be introduced, which is used for the gradient direction $G_i := -\Pi A Z_i$. □

Algorithm 8 NPCG(A, M_0, Π)

$R_0 := I_n - AM_0$
$Z_0 := \Pi R_0$
$G_0 := -\Pi A Z_0$
$P_0 = -G_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := -(R_i, G_i)_F / (P_i, AP_i)_F$
  $M_{i+1} := M_i + \alpha_i P_i$
  $R_{i+1} := R_i - \alpha_i A P_i$
  $Z_{i+1} := \Pi R_{i+1}$
  $G_{i+1} := -\Pi A Z_{i+1}$
  $\beta_{i+1} := (R_{i+1}, G_{i+1})_F / (R_i, G_i)_F$
  $P_{i+1} := -G_{i+1} + \beta_{i+1} P_i$
end for

Algorithm 9 PCG(A, M_0, Π)

$R_0 := I_n - AM_0$
$Z_0 := \Pi R_0$
$P_0 = Z_0$
for $i = 0, 1, \ldots$ **do**
  $\alpha_i := (R_i, Z_i)_F / (P_i, AP_i)_F$
  $M_{i+1} := M_i + \alpha_i P_i$
  $R_{i+1} := R_i - \alpha_i A P_i$
  $Z_{i+1} := \Pi R_{i+1}$
  $\beta_{i+1} := (R_{i+1}, Z_{i+1})_F / (R_i, Z_i)_F$
  $P_{i+1} := Z_{i+1} + \beta_{i+1} P_i$
end for

Preconditioning is applied to the CG and LOMR methods very similarly to what is done in the proof of Proposition 10. That is, the method is applied to the left-preconditioned system in Eq. (6.6), whereas the standard Frobenius inner product is replaced by an inner product weighted by Π^{-1}, thus rendering ΠA self-adjoint with respect to the inner product, eventually leading to the procedures given in Algos. 9 and 10.

7. Dropping strategies

As the iterative methods of Sections 2 to 6 are deployed, the number of nonzero components in the main iterate M and the search direction P, where we drop the iteration index for clarity, can grow uncontrollably, slowing down iterations, likely exceeding memory capacity. To circumvent this issue, dropping strategies need be deployed. That is, methods to set nonzero values in M and P to zero, thus relieving some of the burden of excessive storage requirements, all the while minimizing convergence hindering. In this Section, we draw inspiration from the work of [10], with the difference that we adapt these methods to a global treatment of the nonzero patterns rather than per-column approaches.

Algorithm 10 LOPMR(A, M_0, Π)

$R_0 := I_n - AM_0$
$P_{-1} := 0_{n\times n}$
$Z_0 := \Pi R_0$
for $i = 0, 1, \ldots$ **do**
  if $i = 0$ **then**
    $\delta_i := (Z_i, AZ_i,)_F / (AZ_i, \Pi AZ_i)_F$
    $\gamma_i := 1$
  else
    $c_i := (AZ_i, \Pi AZ_i)_F \cdot (AP_{i-1}, \Pi AP_{i-1})_F - (AZ_i, \Pi AP_{i-1})_F^2$
    $\delta_i := \left[(AP_{i-1}, \Pi AP_{i-1})_F \cdot (Z_i, AZ_i,)_F - (AZ_i, \Pi AP_{i-1})_F \cdot (Z_i, AP_{i-1})_F\right] / c_i$
    $\gamma_i := \left[(AZ_i, \Pi AZ_i)_F \cdot (Z_i, AP_{i-1},)_F - (AZ_i, \Pi AP_{i-1})_F \cdot (Z_i, AZ_i)_F\right] / c_i$
  end if
  $M_{i+1} := M_i + \delta_i Z_i + \gamma_i P_{i-1}$
  $R_{i+1} := R_i - \delta_i AZ_i - \gamma_i AP_{i-1}$
  $P_i := Z_i + (\gamma_i/\delta_i) P_{i-1}$
  $Z_{i+1} := \Pi R_{i+1}$
end for

7.1. *Dropping in the main iterate*

Let the nonzero components of M be denoted by $m_{k\ell}$, and the corresponding residual be $R := I_n - AM$. We wish to set multiple nonzero components of M to zero, simultaneously. In particular, let us introduce the set

$$\mathscr{D}_M \subset \mathscr{NNZ}(M) := \{(k,\ell) \text{ such that } m_{k\ell} \neq 0\} \tag{7.1}$$

of indices corresponding to the entries we wish to drop, where $\mathscr{NNZ}(M)$ is the set of indices of nonzero components in M. The perturbed right-approximate inverse iterate is then given by:

$$\widehat{M} := M - \sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell} e_k e_\ell^T \tag{7.2}$$

and the corresponding perturbed residual is given by:

$$\widehat{R} = R + \sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell} A e_k e_\ell^T. \tag{7.3}$$

Then, the Frobenius norm of the perturbed residual becomes

$$\|\widehat{R}\|_F^2 = \|R\|_F^2 + \sum_{(k,\ell)\in\mathscr{D}_M} \sum_{(r,s)\in\mathscr{D}_M} m_{k\ell} m_{rs} \left(Ae_k e_\ell^T, Ae_r e_s^T\right)_F + 2 \sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell} \cdot (AR)_{k\ell} \tag{7.4}$$

where $(AR)_{k\ell}$ denotes the (k,ℓ)-entry of AR. Note that the coupling term $(Ae_k e_\ell^T, Ae_r e_s^T)_F$ cancels for all $\ell \neq s$. This means that the effect of nonzero dropping interactions with other droppings only occurs

within columns:

$$\sum_{(k,\ell)\in\mathscr{D}_M}\sum_{(r,s)\in\mathscr{D}_M} m_{k\ell}m_{rs}\left(Ae_k e_\ell^T, Ae_r e_s^T\right)_F = \sum_{t=1}^{n}\sum_{\substack{(k,\ell)\in\mathscr{D}_M\\ \ell=t}}\sum_{\substack{(r,s)\in\mathscr{D}_M\\ s=t}} m_{kt}m_{rt}(Ae_k)^T(Ae_r). \tag{7.5}$$

The overall dropping effect on the residual Frobenius norm can then be recast into

$$\begin{aligned}\|\widehat{R}\|_F^2 - \|R\|_F^2 = &\sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell}^2\|Ae_k\|_2^2 + 2\sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell}\cdot(AR)_{k\ell}\\ &+\sum_{t=1}^{n}\sum_{\substack{(k,\ell)\in\mathscr{D}_M\\ \ell=t}}\sum_{\substack{(r,s)\in\mathscr{D}_M\\ s=t\\ r\neq k}} m_{kt}m_{rt}(Ae_k)^T(Ae_r).\end{aligned} \tag{7.6}$$

Note that, for the case in which there is only one nonzero component dropped per column, the last term vanishes, in which case we have:

$$\|\widehat{R}\|_F^2 - \|R\|_F^2 = \sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell}^2\|Ae_k\|_2^2 + 2\sum_{(k,\ell)\in\mathscr{D}_M} m_{k\ell}\cdot(AR)_{k\ell}. \tag{7.7}$$

In practice, we do want to drop more than one nonzero component per column at once, but for practical reasons, we ignore the last term on the right-hand-side of Eq. (7.6), even when we do drop more than one nonzero element per column. That is, Eq. (7.7) provides a blueprint to pick nonzero entries of M^{-1} whose dropping leads to the most significant decrease $\|\widehat{R}\|_F^2 - \|R\|_F^2$ of residual Frobenius norm. Although we expect the magnitude of the coupling term ignored in Eq. (7.6) to grow with $|\mathscr{D}_M|$, we find no approach to account for this term in the design of a sufficiently efficient dropping strategy. Our dropping strategy is as follows:

1. Symmetrize M: $M := \left(M + M^T\right)/2$.
2. Drop insignificant non-diagonal components: $m_{k\ell} := 0$ for all $|m_{k\ell}| < u$ with $k \neq \ell$, where u denotes the unit round-off.
3. If a threshold density is achieved, i.e., if $|\mathscr{NNZ}(M)| > m$ for some fixed $m \ll n^2$, apply the strategy of Eq. (7.8). That is, whenever $|\mathscr{NNZ}(M)| > m$, we deploy the following strategy:

$$\text{For } (k,\ell)\in\mathscr{D}_M = \underset{\substack{\mathscr{S}\subset\mathscr{NNZ}(M)\setminus\{(k,k),k=1,\ldots,n\}\\ |\mathscr{NNZ}(M)|-|\mathscr{S}|=m}}{\arg}\ \min\left\{\sum_{(k,\ell)\in\mathscr{S}} m_{k\ell}\|Ae_k\|_2^2 + 2\sum_{(k,\ell)\in\mathscr{S}} m_{k\ell}\cdot(AR)_{k\ell}\right\}, \tag{7.8}$$

$$\text{set } m_{k\ell} := 0.$$

Conveniently, the matrix AR is already formed at each iteration in all three non-preconditioned methods presented here. For the preconditioned methods, extra storage may be needed to store AR along AZ. The dot products $\|Ae_1\|_2^2, \ldots, \|Ae_n\|_2^2$ need be computed only once, at the start of the algorithm.

TABLE 1 *Operation count per iteration without dropping*

Method	Operation count per iteration
MR	1 SpGEMM† + 1 Frobenius inner product + 1 Frobenius norm + 2 SpGEAMs*
SD	2 SpGEMMs + 1 Frobenius inner product + 1 Frobenius norm + 2 SpGEAMs
NCG	2 SpGEMMs + 2 Frobenius inner products + 3 SpGEAMs
CG	1 SpGEMM + 1 Frobenius inner product + 1 Frobenius norm + 3 SpGEAMs
LOMR	2 SpGEMMs + 3 Frobenius inner products + 2 Frobenius norms + 5 SpGEAMs

†: General product of two sparse matrices.

*: General addition of two sparse matrices.

7.2. *Dropping in the search direction*

Let the nonzero components of P be denoted by $p_{k\ell}$. We wish to set multiple nonzero components of P to zero simultaneously. In particular, let us introduce the set

$$\mathscr{D}_P \subset \mathscr{NNZ}(P) := \{(k,\ell) \text{ such that } p_{k\ell} \neq 0\} \tag{7.9}$$

of indices corresponding to the entries we wish to drop, where $\mathscr{NNZ}(P)$ is the set of indices of nonzero components in P. The dropping strategy for the search direction is not given as sophisticated a consideration as that of the main iterate. That is, whenever $|\mathscr{NNZ}(P)| > m$ for some $m \ll n^2$, we deploy the following strategy:

$$\text{For } (k,\ell) \in \mathscr{D}_P = \underset{\substack{\mathscr{S}\subset\mathscr{NNZ}(P)\\ |\mathscr{NNZ}(P)|-|\mathscr{S}|=m}}{\arg\min} \left\{ \sum_{(k,\ell)\in\mathscr{S}} |p_{k\ell}| \right\}, \text{ set } p_{k\ell} := 0. \tag{7.10}$$

This strategy is simpler to deploy than Eq. (7.8) and requires no additional storage. Note that, unlike those of the main iterate, the diagonal components the search direction can be dropped.

7.3. *Effects of dropping on implementations*

As dropping strategies are applied to an iterate M_{i+1}, the update formulae for the residual iterate R_{i+1} do not hold anymore. Consequently, for each of the algorithms presented in this paper, once dropping strategies are deployed, the residual iterates need be explicitly computed, that is, we use the relation $R_{i+1} := I_n - AM_{i+1}$ instead of the previously derived update formulae. In Table 1, we detail the operation count per iteration when no dropping strategy is deployed. As shown in Table 2, once dropping strategies are deployed, each residual update entails an additional product between sparse matrices (SpGEMM). In general, the methods that make use of the gradient, that is, SD and NCG iterations, require one additional SpGEMM compared to the one-dimensional projections with search directions defined along the residual, that is, MR and CG iterations. Enforcing local optimality, that is, LOMR iterations, also require one additional SpGEMM per iteration.

8. Numerical experiments

In this section, we present the results of a number of experiments that were conducted to showcase the behavior of the algorithms presented in this paper, particularly in comparison to the state-of-the-art (P)MR and (P)SD methods that are used for global iterations of approximate inverses. These

TABLE 2 *Operation count per iteration with dropping*

Method	Operation count per iteration			
MR	2 SpGEMM†	+ 1 Frobenius inner product	+ 1 Frobenius norm	+ 2 SpGEAMs
SD	3 SpGEMMs	+ 1 Frobenius inner product	+ 1 Frobenius norm	+ 2 SpGEAMs
NCG	3 SpGEMMs	+ 2 Frobenius inner products		+ 3 SpGEAMs
CG	2 SpGEMM	+ 1 Frobenius inner product	+ 1 Frobenius norm	+ 3 SpGEAMs
LOMR	3 SpGEMMs	+ 3 Frobenius inner products	+ 2 Frobenius norms	+ 4 SpGEAMs

†: General product of two sparse matrices.

*: General addition of two sparse matrices.

experiments are labeled from Experiment01 to Experiment07 with corresponding Julia scripts and Python post-processing scripts available at https://github.com/venkovic/julia-global-spd-spai. For these experiments, we present 10 test sparse SPD matrices and their main characteristics in Table 3. The nonzero patterns of these matrices are plotted in Fig. 1. We distinguish between two groups of matrices. First, we have relatively small matrices, i.e., $n < 5{,}000$, for which we are able to compute the entire spectrum, as well as run global iteration algorithms without the need to drop nonzero values. These matrices are

- bcsstk21: Ill-conditioned stiffness matrix from a structural mechanics problem of a clamped square plate. All eigenvalues are greater than 1. Bandwidth of 125 with only a handful of nonzero diagonals, typical of finite element discretizations.
- tri100eigs4k: Custom-made ill-conditioned tridiagonal matrix $A = LL^T$, where the lower bidiagonal matrix L has 100 distinct random eigenvalues (diagonal components) independently and identically distributed (iid) as uniform random variables (RVs) in $(0.05, 1.05)$. All other eigenvalues (diagonal components) of L are 1. The sub-diagonal components of L are iid uniform RVs in $(0,1)$. A is nearly singular with a smallest eigenvalue of 10^{-8}.
- msc04515: Moderately conditioned structural mechanics matrix from Boeing obtained using the NASTRAN finite element software. Bandwidth of 162 with only a handful of nonzero diagonals. Relatively large eigenvalues with a spectrum spanning from 10^4 to 10^{10}.

The seven other matrices, for which the growth of nonzero density needs be contained, have dimensions spanning from more than 10,000 to more than 30,000. These matrices are

- bundle1: Moderately conditioned generalized arrowhead matrix from 3D vision bundle adjustment. Very large eigenvalues spanning from 10^9 to 10^{12}.
- 4bw100eigs20k: Ill-conditioned custom-made banded matrix $A = LL^T$, where the lower triangular matrix L has a dense bandwidth of 4. The factor L has 100 iid random distinct eigenvalues (diagonal components) uniformly distributed in $(0, 0.05)$. All the other eigenvalues of L are set to 1. Each sub-diagonal of L has uniform iid random components in $(0, 0.01)$. A is nearly singular with a smallest eigenvalue of 10^{-9}.
- 4bw100eigs20k2: Ill-conditioned custom-made matrix $A = LL^T$ similar to 4bw100eigs20k, but for which the random distinct eigenvalues of L are in $(0, 10^5)$. Unlike 4bw100eigs20k, 4bw100eigs20k2 is not nearly singular.

TABLE 3 *Metadata of test sparse matrices*

Matrix	n	nnz	nnz/n^2	λ_{min}	λ_{max}	κ
bcsstk21†	3,600	26,600	2.05×10^{-3}	7.21×10^{0}	1.27×10^{8}	1.76×10^{7}
tri100eigs4k*	4,000	11,998	7.50×10^{-4}	9.26×10^{-9}	3.56×10^{0}	3.85×10^{8}
msc04515*	4,515	97,707	4.79×10^{-3}	1.39×10^{4}	3.15×10^{10}	2.26×10^{6}
bundle1†	10,581	770,811	6.89×10^{-3}	6.40×10^{9}	6.43×10^{12}	1.00×10^{3}
4bw100eigs20k*	20,000	179,980	4.50×10^{-4}	2.47×10^{-9}	1.05×10^{0}	4.25×10^{8}
4bw100eigs20k2*	20,000	179,980	4.50×10^{-4}	9.74×10^{-1}	9.79×10^{9}	1.01×10^{10}
rand20k*	20,000	99,772	2.49×10^{-4}	8.70×10^{-2}	1.00×10^{8}	1.15×10^{9}
rand20k2*	20,000	99,772	2.49×10^{-4}	8.67×10^{-6}	1.00×10^{4}	1.15×10^{9}
wathen100†	30,401	471,601	5.10×10^{-4}	6.36×10^{-2}	3.70×10^{2}	5.82×10^{3}
Poisson32k*	31,839	221,375	2.18×10^{-4}	6.21×10^{-4}	9.66×10^{2}	1.55×10^{5}

†: https://sparse.tamu.edu/

*: https://github.com/venkovic/matrix-market

- rand20k: Ill-conditioned custom-made matrix $A = LL^T$ with random off-diagonal nonzero pattern. The lower triangular factor L has iid random eigenvalues in $(0, 10^4)$, and all nonzero off-diagonal components are in $(0, 1)$.
- rand20k2: Ill-conditioned custom-made matrix $A = LL^T$ with the same nonzero pattern and off-diagonal values as rand20k. For rand20k2, the eigenvalues of L are in $(0, 10^2)$, rendering A nearly singular, with a smallest eigenvalue of 10^{-6}.
- wathen100: Moderately conditioned random banded matrix with a bandwidth of 304, coming from Andy Wathen at Oxford University.
- Poisson32k: Moderately conditioned stiffness matrix of a P1 finite element unstructured discretization of the 2D Poisson PDE with a random variable coefficient sampled as a Gaussian process with unit-variance squared exponential covariance. Originally from [20].

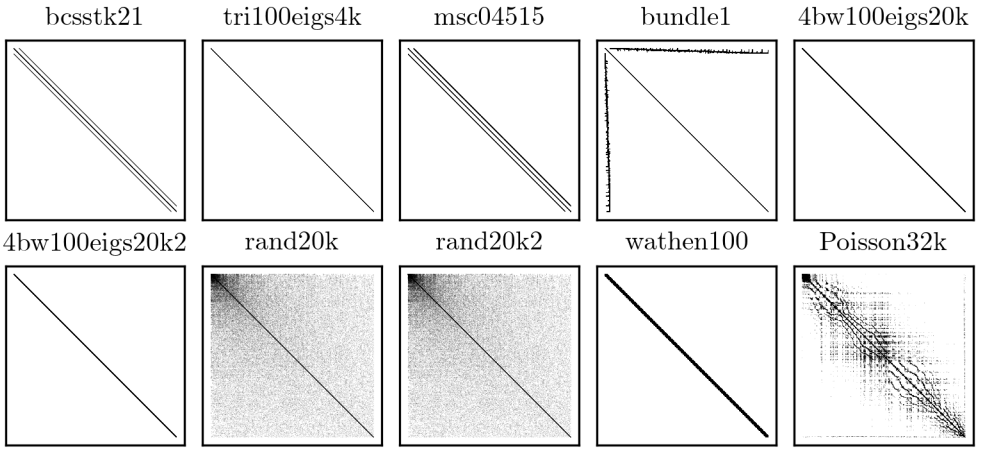

FIG. 1. Nonzero structures of test sparse matrices (Experiment00).

8.1. *Dropping-free experiments*

In this section, we present the results of dropping-free experiments, first on small matrices, for which entire spectra can be computed, and then, on medium size matrices.

8.1.1. Small matrices

We benchmark five methods (PSD, NPCG, PMR, PCG, and LOPMR) using Jacobi preconditioners to compute approximate inverses of three small matrices: bcsstk21, tri100eigs4k, and msc04515. This experiment (Experiment01 in the corresponding GitHub repository) is performed without dropping, allowing the nonzero densities of the main iterate and search direction to grow uncontrollably. Despite this unconstrained expansion of the nonzero pattern, the approximate inverse of each matrix exhibits density bounds that are inherent to both the original sparse matrix and the nature of the global iteration method. For bcsstk21, all five methods yield approximate inverses with densities of exactly 50%. For msc04515, all approximate inverses achieve 52% density. The tri100eigs4k matrix shows method-dependent density variations: PMR achieves the sparsest result at 1.24% density, followed by PSD at 2.31%, while both PCG and LOPMR yield identical densities of 14.5%, and NPCG produces the densest approximate inverse at 27.7%.

We plot the Frobenius norm of the residual as a function of the number of iterations in Figure 2. The results show that both PCG and LOPMR achieve convergence for all three matrices. LOPMR exhibits monotone decrease of the residual norm, whereas PCG, which does not minimize the residual norm, can show significant oscillations which are particularly evident for the nearly singular and ill-conditioned matrix tri100eigs4k. Note, however, that after several hundred iterations on the tri100eigs4k matrix, PCG does converge and nearly matches LOPMR's behavior. This convergence pattern is consistent with Krylov subspace approximations of matrices having few distinct eigenvalues, which possess low-degree minimal polynomials that enable rapid convergence once the iteration count approaches the number of distinct eigenvalues.

Figure 2 shows that the Frobenius residual norm barely converges for PSD, PCG, and PMR. Among these three methods, PSD performs worst, while NPCG offers the expected slight improvement over

PSD. PMR outperforms NPCG for bcsstk21 and msc04515, but the reverse is true for tri100eigs4k. In all cases, these three methods fail to achieve the symbolic threshold of $\|R\|_F < 1$, below which the approximate inverse is guaranteed to be non-singular (though not necessarily SPD).

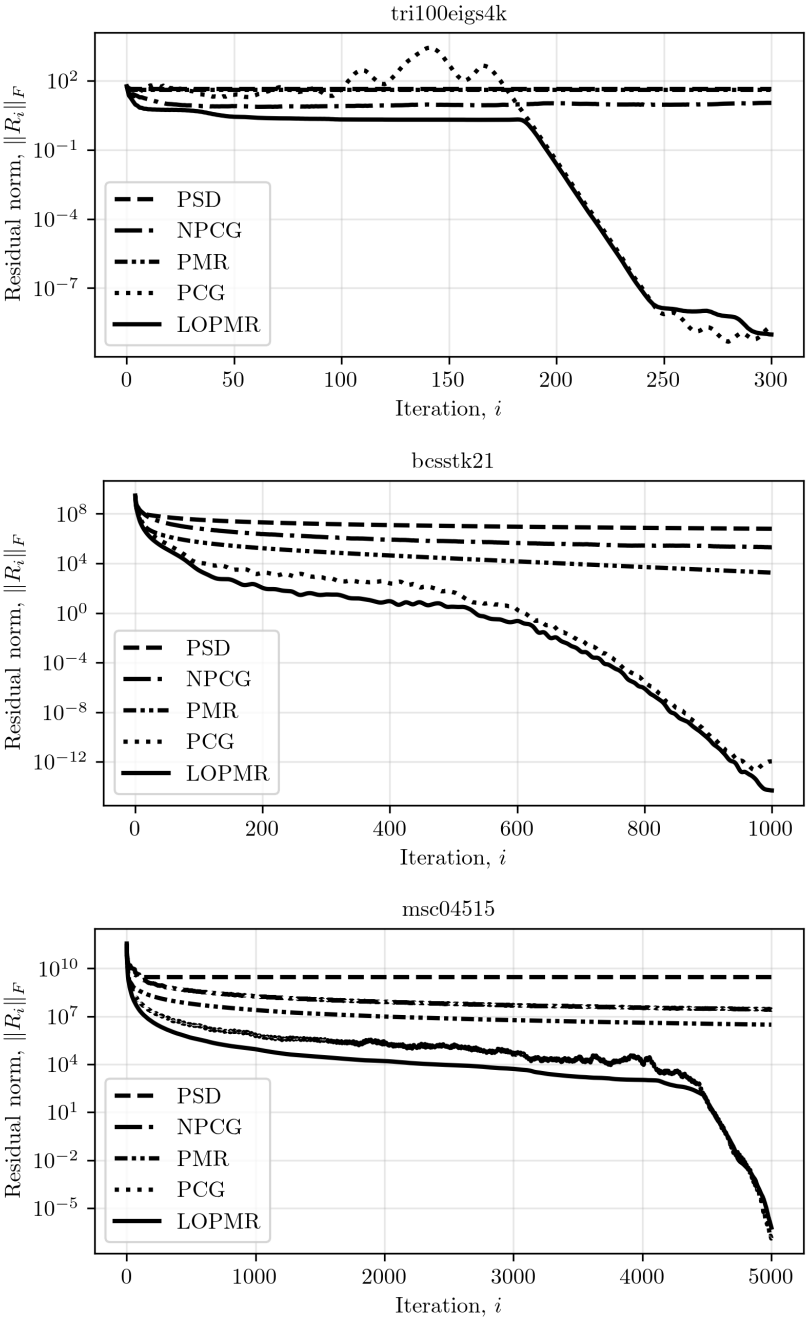

FIG. 2. Convergence plots of dropping-free experiments (Experiment01).

Figure 3 presents the spectra of the approximate inverses obtained by each method for all three small matrices. Most notably, the approximate inverses generated by PCG and LOPMR remarkably reproduce the spectrum of A^{-1}. Importantly, these approximate inverses are SPD, which has significant implications for deploying an approximate inverse as a preconditioner in standard CG iterations. In contrast, the three other methods (PSD, NPCG, and PMR) all fail to produce positive definite approximate inverses, which represents a major shortcoming for practical applications. Note, however, that we purposely selected matrices that expose the limitations of these methods. For some matrices, all five methods can produce SPD approximate inverses that reasonably reproduce the spectrum of A^{-1}. We invite readers to test this with the Poisson4k[1], triclust4k[1] and triunif4k[1] matrices. Nevertheless, both PCG and LOPMR consistently outperform the other three methods.

[1] https://github.com/venkovic/matrix-market

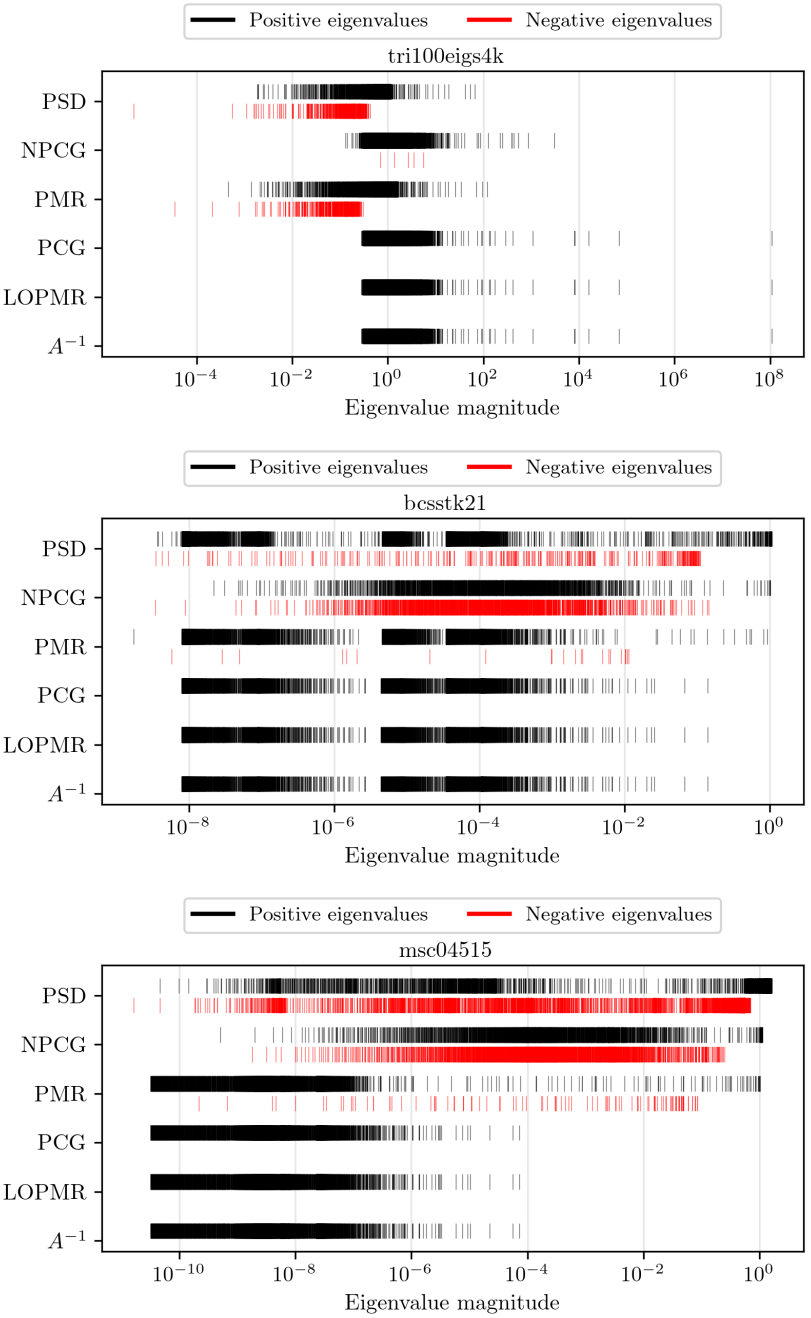

FIG. 3. Converged spectra of dropping-free experiments (Experiment01).

8.1.2. Medium size matrices

We now focus on the seven other matrices for which we cannot afford to let the approximate inverse and search direction grow uncontrollably dense, nor can we compute complete spectra for these matrices. Nevertheless, we first examine how global iteration methods perform without dropping. Given the behaviors observed in the small matrix experiments, we hereafter focus on the two best-performing methods (LOPMR and PCG) as well as the current state-of-the-art method for global iteration (PMR). We deploy these three methods with a stopping criterion based on the density of the main iterate M_i: iteration stops once a density of 3% or higher is achieved. Given the different nonzero patterns of the matrices under consideration, the number of iterations required to reach this stopping criterion varies significantly, as reported in Table 4.

Table 4 shows that, besides the custom-made matrices 4bw100eigs20k and 4bw100eigs20k2, all methods produce iterates with exactly the same density. Another notable observation is how far beyond the 3% threshold some iterates can jump in a single iteration. For instance, the approximate inverses for bundle1 all achieve 21.5% density. Less dramatically but still significantly, the approximate inverses of rand20k and rand20k2 reach 7.86% density. We attribute bundle1's sudden density increase to the matrix's arrowhead structure, while for the two random matrices, we assume the randomness of the nonzero pattern causes the abrupt rise in main iterate density. These observations are important because they indicate which matrices will likely require more aggressive dropping in the next section. We will see that large amounts of dropping not only carry computational cost but can also detrimentally impact the convergence of global iteration methods.

Except for 4bw100eigs20k2, all tested matrices reach the stopping criterion based on the density upper bound. Despite having the worst condition number, the Frobenius residual norm of the 4bw100eigs20k2 approximate inverse decreases very quickly when using PCG and LOPMR, allowing iteration to stop based on a residual norm criterion such as $\|R_i\| < 1$. In contrast, PMR stops at 200 iterations because the Frobenius residual norm of its approximate inverses completely stagnates with no sign of convergence, while producing only very slow density growth. All three methods stop well before reaching the 3% density threshold. Given this behavior, 4bw100eigs20k2 is not a suitable candidate for dropping experiments, since PCG and LOPMR have already converged while PMR makes no progress despite being far from requiring dropping. Therefore, we can directly examine the quality of the approximate inverses generated for 4bw100eigs20k2. Figure 4 shows both the convergence of the Frobenius residual norm and the matrix-vector linear PCG convergence behavior to a 10^{-6} backward error when using these approximate inverses as preconditioners. The approximate inverses generated by PCG and LOPMR both constitute excellent preconditioners, reducing convergence from 94 to 4 iterations, with the same density of 0.25%. Surprisingly, despite producing an indefinite approximate inverse with high residual norm, PMR's approximate inverse, which has a density of 0.84%, still achieves convergence when applied to PCG, though it slows the linear solve from 94 to 177 iterations.

For the six other medium-sized matrices, most approximate inverses are indefinite, and those that are positive definite do not yet constitute good preconditioners. Therefore, all six matrices (bundle1, 4bw100eigs20k, rand20k, rand20k2, wathen100, and Poisson32k) are suitable candidates for dropping experiments. By conducting these experiments, we aim to continue global iterations to further decrease the residual norm while dropping sufficient nonzero values to control memory usage. These experiments are documented in Section 8.2.

As previously mentioned, our sparse matrix selection is designed to demonstrate situations where our proposed methods (PCG and, more particularly, LOPMR) outperform PMR. However, some matrices allow all three methods (PMR, PCG, and LOPMR) to construct SPD approximate inverses

TABLE 4 *Summary results of dropping-free SPAIs stopped after density reaches 3% (Experiment03).*

Method	λ_{min}	λ_{max}	$\|I_n - AM\|_F$	# *iters*	nnz/n^2
		bundle1			
PMR	-6.81×10^{0}	8.06×10^{0}	4.59×10^{12}	2	2.15×10^{-1}
PCG	-7.39×10^{0}	8.29×10^{0}	3.96×10^{12}	2	2.15×10^{-1}
LOPMR	-7.79×10^{0}	8.82×10^{0}	2.79×10^{12}	2	2.15×10^{-1}
		4bw100eigs20k			
PMR	1.92×10^{0}	2.98×10^{5}	9.87×10^{0}	76	1.04×10^{-2}
PCG	-9.90×10^{4}	7.94×10^{8}	2.39×10^{2}	76	3.02×10^{-2}
LOPMR	1.91×10^{0}	7.10×10^{7}	1.44×10^{0}	76	3.02×10^{-2}
		4bw100eigs20k2			
PMR	-4.78×10^{2}	4.81×10^{2}	2.16×10^{10}	200	8.41×10^{-3}
PCG	4.45×10^{-9}	2.05×10^{0}	1.64×10^{-2}	6	2.45×10^{-3}
LOPMR	4.80×10^{-9}	2.05×10^{0}	1.67×10^{-2}	6	2.45×10^{-3}
		rand20k			
PMR	-1.41×10^{3}	1.42×10^{3}	3.18×10^{9}	5	7.86×10^{-2}
PCG	-3.10×10^{1}	3.59×10^{1}	8.07×10^{4}	5	7.86×10^{-2}
LOPMR	-4.97×10^{1}	5.46×10^{1}	1.19×10^{5}	5	7.86×10^{-2}
		rand20k2			
PMR	-9.03×10^{1}	8.82×10^{3}	1.66×10^{3}	5	7.86×10^{-2}
PCG	-8.37×10^{1}	8.81×10^{3}	1.94×10^{3}	5	7.86×10^{-2}
LOPMR	-8.68×10^{1}	8.81×10^{3}	1.23×10^{3}	5	7.86×10^{-2}
		wathen100			
PMR	-5.01×10^{-2}	3.14×10^{1}	5.31×10^{2}	9	3.02×10^{-2}
PCG	-4.26×10^{-2}	3.11×10^{1}	1.26×10^{2}	9	3.02×10^{-2}
LOPMR	-1.28×10^{-2}	3.10×10^{1}	1.04×10^{2}	9	3.02×10^{-2}
		Poisson32k			
PMR	-9.90×10^{-2}	1.05×10^{2}	1.73×10^{1}	17	3.22×10^{-2}
PCG	-8.20×10^{-1}	5.06×10^{2}	3.35×10^{1}	17	3.22×10^{-2}
LOPMR	1.96×10^{-2}	3.44×10^{2}	7.99×10^{0}	17	3.22×10^{-2}

that constitute good preconditioners even when limiting the main iterate density to a maximum of 3%. This is the case for the following matrices:

- crystm02[2]: Well-conditioned banded mass matrix for the finite element modeling of free crystal vibration. Submitted by Boeing. Nearly singular, with a smallest eigenvalue of 10^{-15}.
- Dubcova1[2]: Well-conditioned matrix obtained with a PDE solver. Submitted by Dubcova et al. from the University of Texas at El Paso.
- jnlbrng1[2]: Well-conditioned banded matrix from a quadratic journal bearing optimization problem.
- minsurfo[2]: Well-conditioned banded matrix from minimum surface optimization problem.

- ted_B[2]: Ill-conditioned banded matrix obtained by finite element method applied to coupled thermoelasticity equations on a bar. Nearly singular matrix, with a smallest eigenvalue of 10^{-8}. Submitted by Bindel from UC Berkeley.

Note that for all these matrices, the approximate inverse obtained by LOPMR always performed better as a preconditioner to linear matrix-vector PCG iterations than the approximate inverses obtained by PMR and PCR.

8.2. *Experiments with dropping*

Experiment05 through Experiment07 are performed with nonzero dropping strategies. In Experiment05, we apply the PMR, PCG, and LOPMR methods to compute SPAIs using the dropping strategy described in Section 7. The results are summarized in Table 5 for stages of the iterative procedures at which satisfactory SPAIs are obtained, when possible, and for the last computed SPAI otherwise. In Experiment06, we assess the performance of the SPAIs obtained from Experiment05 when used as preconditioners for linear solves by CG. Experiment07 is an extension of Experiment06, in which the number of iterations necessary for the convergence of linear solves to backward errors of 10^{-6} is assessed for every single iterate of the SPAI. From the results of Experiment05, we note that dropping takes effect from the first iteration through symmetrization of the main iterate, followed by dropping nonzero values with magnitudes smaller than unit round-off. Consequently, even though the 4bw100eigs20k and rand20k matrices quickly reached the maximum 3% density in the previous experiment, they do not reach this maximum density in Experiment05. For the 4bw100eigs20k matrix, LOPMR converges to an SPD SPAI with 0.12% density, whereas PMR makes no progress (as shown in Figure 6) but nevertheless yields an SPD SPAI with 0.16% density. Both SPAIs constitute good preconditioners, with LOPMR outperforming PMR. For the rand20k matrix, neither PMR nor PCG makes substantial progress toward a good inverse approximation, and as seen in Figure 7, both SPAIs cannot be used as effective preconditioners. In contrast, LOPMR converges in just a few global iterations to a very sparse (0.03%) SPAI that closely approximates the inverse, providing a preconditioner that enables PCG convergence in 3 iterations compared to 1,341 iterations for unpreconditioned CG. The rand20k matrix truly demonstrates LOPMR's capacity to build high-quality SPAIs.

The bundle1 matrix is of particular interest because, as shown in the previous section, it grows very dense through global iterations. Consequently, dropping nonzero values becomes the computationally dominant part of global iterations, making the additional computations of LOPMR and PCG over PMR negligible. As shown in Table 5, neither PMR nor PCG achieves SPD SPAIs, even though PMR achieves a smaller Frobenius residual norm than LOPMR–a very rare occurrence. The SPAI obtained by LOPMR constitutes a good preconditioner, enabling PCG convergence in 44 iterations compared to 133 iterations for unpreconditioned CG.

Another matrix for which dropping nonzeros has a significant effect is rand20k2. Despite having the same nonzero structure and condition number as rand20k, the rand20k2 matrix proves more difficult for achieving SPAIs. As shown in Figure 8, neither PMR nor PCG makes substantial progress toward a low-residual SPAI. In contrast, the LOPMR method exhibits periodic spikes in the residual norm, which are artifacts of the dropping strategy. Throughout the global iteration, the density of the main iterate oscillates in phase with the convergence pattern shown in Figure 8. Despite experiencing this

[2] https://sparse.tamu.edu/

TABLE 5 *Summary results of SPAIs obtained with dropping of nonzero values for maximum densities of 3% (Experiment05).*

Method	λ_{min}	λ_{max}	$\|I_n - AM\|_F$	# *iters*	nnz/n^2
		bundle1			
PMR	-1.37×10^{-13}	1.56×10^{-10}	2.49×10^{1}	200	3.00×10^{-2}
PCG	-6.25×10^{-4}	4.23×10^{-3}	6.53×10^{8}	200	3.00×10^{-2}
LOPMR	2.17×10^{-13}	1.56×10^{-10}	7.31×10^{1}	200	3.00×10^{-2}
		4bw100eigs20k			
PMR	9.55×10^{-1}	1.49×10^{5}	9.86×10^{0}	5	1.63×10^{-3}
PCG	6.70×10^{-1}	3.26×10^{8}	2.94×10^{1}	200	6.33×10^{-3}
LOPMR	9.42×10^{-1}	2.46×10^{5}	5.85×10^{0}	3	1.25×10^{-3}
		rand20k			
PMR	1.03×10^{-1}	1.47×10^{6}	2.35×10^{10}	200	1.47×10^{-3}
PCG	-3.16×10^{3}	1.86×10^{7}	1.33×10^{13}	200	1.07×10^{-3}
LOPMR	1.25×10^{-6}	1.15×10^{1}	5.49×10^{0}	17	3.39×10^{-4}
		rand20k2			
PMR	-1.68×10^{0}	4.93×10^{4}	9.16×10^{5}	200	1.42×10^{-2}
PCG	-7.97×10^{1}	1.28×10^{6}	6.30×10^{7}	200	2.21×10^{-2}
LOPMR	4.01×10^{-3}	2.02×10^{4}	6.40×10^{1}	200	1.63×10^{-2}
		wathen100			
PMR	2.71×10^{-3}	1.57×10^{1}	3.74×10^{1}	100	3.00×10^{-2}
PCG	2.57×10^{-3}	1.57×10^{1}	3.18×10^{1}	100	3.00×10^{-2}
LOPMR	2.71×10^{-3}	1.57×10^{1}	2.94×10^{1}	46	3.00×10^{-2}
		Poisson32k			
PMR	1.04×10^{-2}	1.52×10^{2}	1.67×10^{1}	100	3.00×10^{-2}
PCG	-1.62×10^{1}	4.46×10^{2}	1.56×10^{2}	100	3.00×10^{-2}
LOPMR	6.67×10^{-3}	1.32×10^{2}	9.92×10^{0}	14	2.22×10^{-2}

convergence artifact, LOPMR's SPAI constitutes an excellent preconditioner, enabling PCG to converge in 6 iterations compared to 1,329 iterations for unpreconditioned CG.

The wathen100 matrix is perhaps the most amenable to SPAI computation of the 10 matrices considered in this work. As shown in Figure 9, all three methods (PMR, PCG and LOPMR) converge to SPD SPAIs, all of which with 3% density, constitute good preconditioners enabling PCG to converge in less than 20 iterations compared to the 168 iterations needed by CG. Once again, LOPMR's SPAI proves to be a better preconditioner than PMR's.

The last tested matrix is Poisson32k. Figure 10 shows the extent to which PCG is impacted by dropping nonzero values, with a clearly noticeable bump in the residual norm when the maximum density is first achieved at iteration 16. Neither PMR nor PCG achieves SPAIs that can constitute good preconditioners, as evidenced by the stagnation of PCG iterations. In contrast, LOPMR quickly converges to an SPD SPAI with 2.22% density, reducing the number of solver iterations from 1,214 to 47. However, an important caveat of the LOPMR method applied to the Poisson32k matrix is that, as

documented by the results of Experiment07, if the iterative procedure is continued beyond 17 iterations, which is when the maximum density is achieved and the dropping strategy effectively becomes active, the quality of the SPAI deteriorates, and using this SPAI as a preconditioner fails to reach the desired convergence of linear solves.

While the sparse matrix selection in Table 3 demonstrates LOPMR's superiority over other global iteration methods for computing SPAIs of SPD matrices, we encountered some matrices for which no good SPD SPAI could be obtained with a maximum density of 3%, regardless of the number of iterations or method used. These matrices are:

- bodyy6[2]: Moderately conditioned matrix from a structural mechanics problem. Submitted by Alex Pothen from NASA.
- gyro_k[2]: Ill-conditioned stiffness matrix from a model order reduction problem.
- gyro_m[2]: Moderately conditioned and nearly singular mass matrix from a model order reduction problem.
- Pres_Poisson[2]: Moderately conditioned matrix from computational fluid dynamics.

9. Conclusion

Global iterative methods were introduced in this work with the goal of improving the convergence behavior of the SD and MR methods for the computation of SPAIs, particularly for SPD matrices. First, the NCG method is introduced using gradient directions for searches, just like for SD, but with the additional constraint that each search direction is made A-orthogonal to the previous search direction, thus making NCG a globally optimal method that minimizes the Frobenius A-norm of the error over the Krylov subspace of A^2 generated by the initial gradient. Secondly, the CG method has a similar effect to NCG on SD, but with respect to the MR method. That is, like MR, the CG method is a one-dimensional projection with search directions defined along the residual, with the additional constraint that each search direction is made A-orthogonal to the previous direction, thus making CG globally optimal, yielding minimizers of the Frobenius A-norm of the error over the Krylov subspace of A generated by the initial residual. Lastly, the LOMR method is introduced by enriching the search space of the MR method with the previous search direction and adapting the orthogonality constraint accordingly.

As evidenced by dropping-free experiments, the NCG and CG methods do accelerate the convergence behaviors of the SD and MR methods, respectively. However, just like SD, the NCG method also exhibits a significantly slower convergence behavior than the MR method, which can be explained by the fact that the NCG method is a Krylov subspace method of A^2, rendering it much more prone to convergence hindering due to the poor conditioning of A than CG, which is a Krylov subspace method of A. Those dropping-free experiments reveal that the LOMR method performs best among all five tested methods, though the CG method exhibits closely matching convergence behavior, albeit with non-monotonic decreases of residual norms and significant oscillations at times. The superiority of the LOMR method for computing good SPAIs is further confirmed when dropping strategies are deployed. The success of the LOMR method comes at a cost, namely one SpGEMM, two Frobenius inner products, one Frobenius norm, and one SpGEAM more than CG, the second-best performing method in this work. Therefore, if CG is capable of achieving a good SPAI, we recommend it over LOMR. However, as shown through experiments, there are a number of SPD matrices for which only LOMR is capable of achieving a sufficiently good SPAI to serve as a preconditioner.

---

[2] https://sparse.tamu.edu/

Some questions remain to be answered. First, while LOMR and CG prove better than the state-of-the-art MR at building good SPAIs of SPD matrices, their iterates are still not guaranteed to be SPD. A logical next step would be to develop locally optimal methods analogous to LOMR for the computation of FSAIs for which positive definiteness is consistently enforced for each iterate. Second, although not documented in this work, we did deploy efforts to try to design effective stopping criteria that guarantee the quality of an SPAI as a preconditioner. Unfortunately, those efforts were unsuccessful, and as of now, we see no better approach than to test each iterate of an SPAI as a preconditioner, as in Experiment07. Third, the methods introduced in this work need to be adapted to parallel computing. Lastly, a similar study has been initiated for the case of general matrices.

REFERENCES

1. Zhong-Zhi Bai and Jianyu Pan. *Matrix Analysis and Computations*. SIAM, 2021.
2. M Benson, Jürgen Krettmann, and M Wright. Parallel algorithms for the solution of certain large sparse linear systems. *International journal of computer mathematics*, 16(4):245–260, 1984.
3. M. W. Benson. Iterative solution of large scale linear systems. Master's thesis, Lakehead University, 1973.
4. Maurice W Benson. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. *Util. Math.*, 22:127–140, 1982.
5. Michele Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, 182(2):418–477, 2002.
6. Michele Benzi, Carl D Meyer, and Miroslav Tuma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM Journal on Scientific Computing*, 17(5):1135–1149, 1996.
7. Michele Benzi and Miroslav Tuma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 19(3):968–994, 1998.
8. Michele Benzi and Miroslav Tuma. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2-3):305–340, 1999.
9. Edmond Chow and Yousef Saad. Parallel approximate inverse preconditioners. In *PPSC*. Citeseer, 1997.
10. Edmond Chow and Yousef Saad. Approximate inverse preconditioners via sparse-sparse iterations. *SIAM J. Sci. Comput.*, 19:995–1023, 1998.
11. Stephen Demko, William F Moss, and Philip W Smith. Decay rates for inverses of band matrices. *Mathematics of computation*, 43(168):491–499, 1984.
12. Iain S Duff, Albert M Erisman, C William Gear, and John K Reid. Sparsity structure and gaussian elimination. *ACM Signum newsletter*, 23(2):2–8, 1988.
13. Paul O Frederickson. *Fast approximate inversion of large sparse linear systems*. Lakehead University, Department of Mathematical Sciences, 1975.
14. Ruth M Holland, Andy J Wathen, and Gareth J Shaw. Sparse approximate inverses and target matrices. *SIAM Journal on Scientific Computing*, 26(3):1000–1011, 2005.
15. Igor E Kaporin. New convergence results and preconditioning strategies for the conjugate gradient method. *Numerical linear algebra with applications*, 1(2):179–210, 1994.
16. Andrew Knyazev. A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace. In *Numerical Treatment of Eigenvalue Problems Vol. 5/Numerische Behandlung von Eigenwertaufgaben Band 5: Workshop in Oberwolfach, February 25–March 3, 1990/Tagung in Oberwolfach, 25. Februar–3. März 1990*, pages 143–154. Springer, 1991.
17. Andrew Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.
18. L Yu Kolotilina and A Yu Yeremin. Factorized sparse approximate inverse preconditionings i. theory. *SIAM Journal on Matrix Analysis and Applications*, 14(1):45–58, 1993.
19. Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2000.

20. Nicolas Venkovic. *Preconditioning strategies for stochastic elliptic partial differential equations*. PhD thesis, Université de Bordeaux, 2023.
21. Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

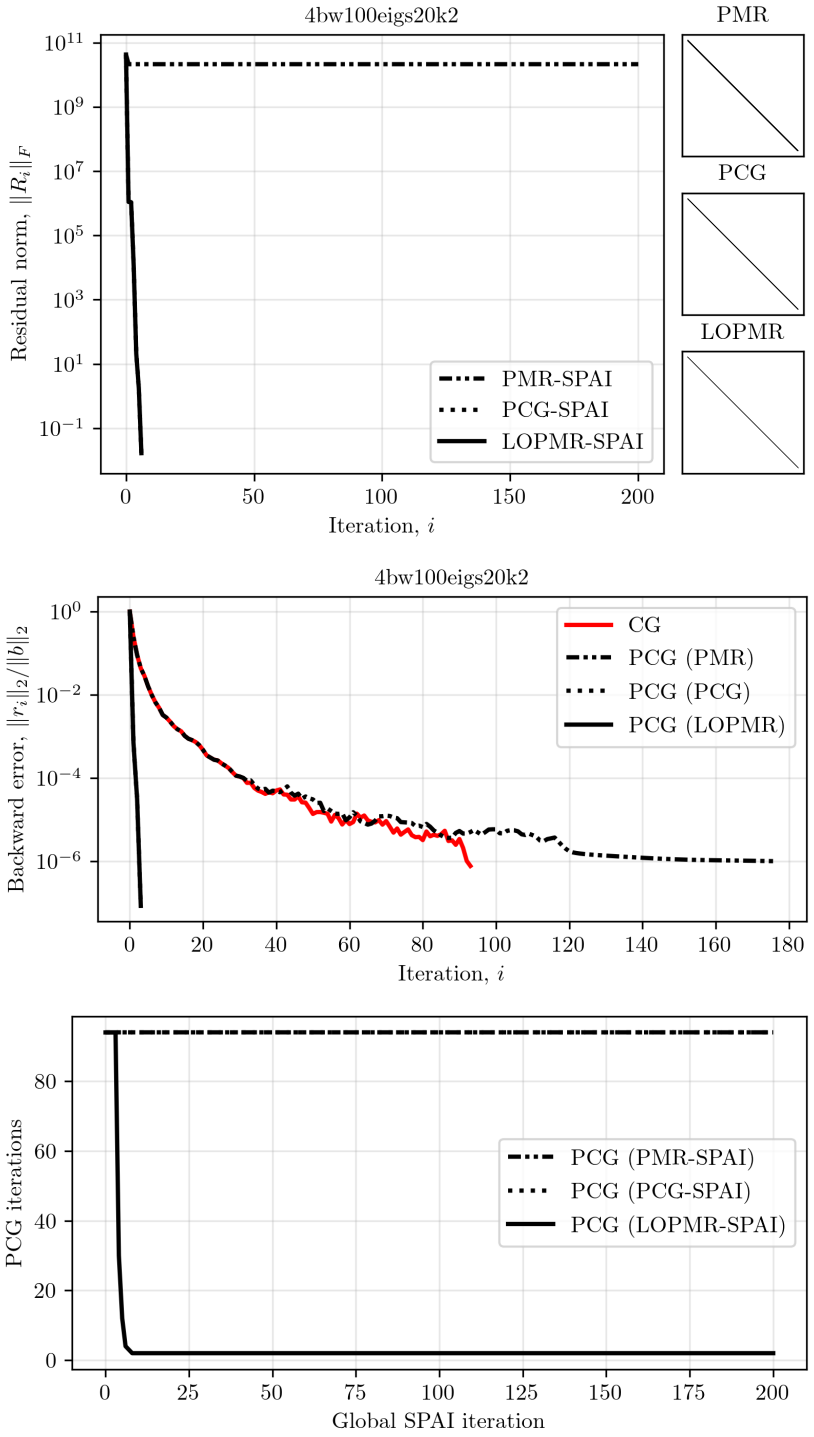

FIG. 4. Convergence plot of dropping-free SPAIs and PCG results for the matrix 4bw100eigs20k2 (Experiment03-04 and Experiment07).

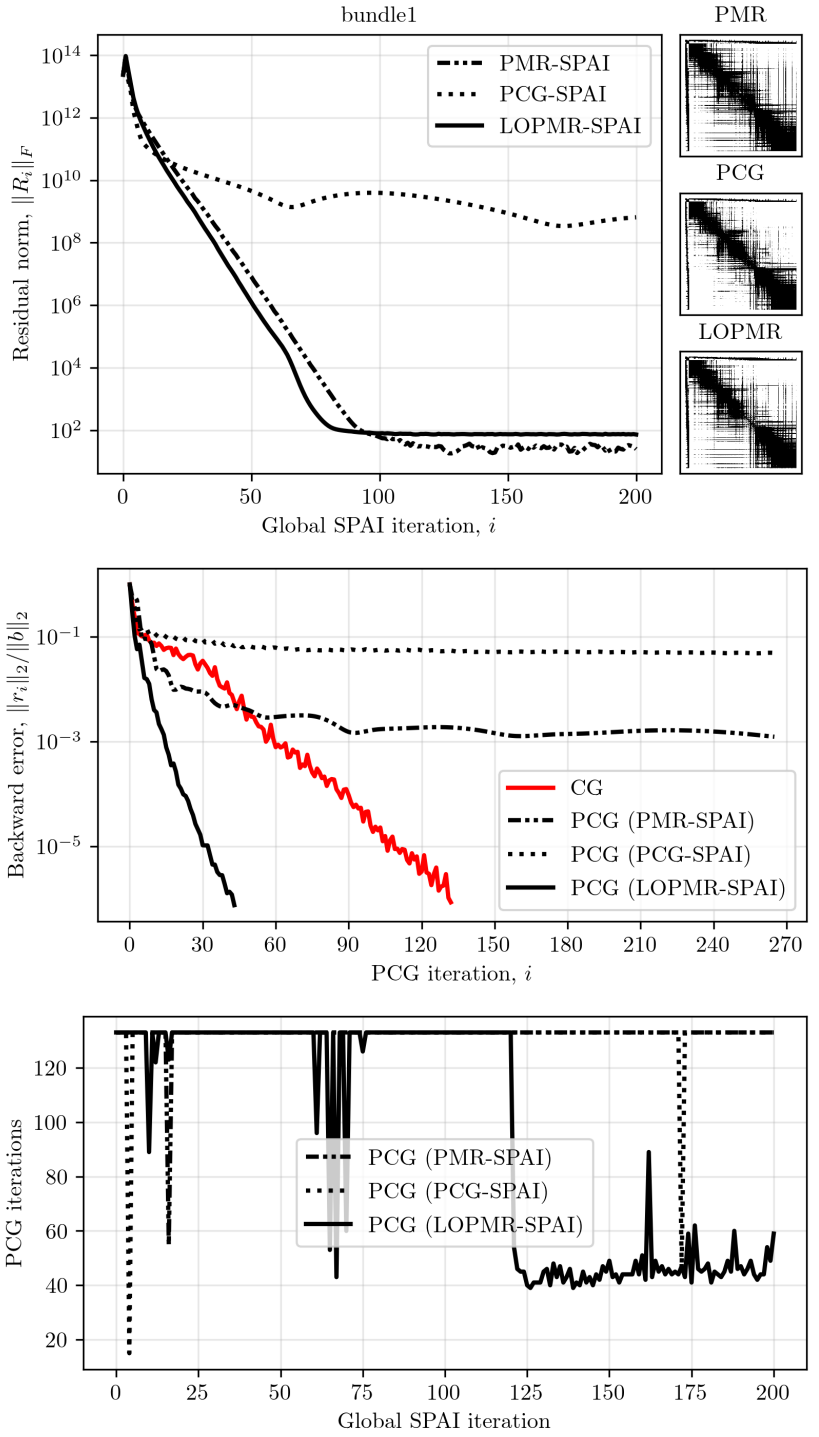

FIG. 5. Convergence plots and PCG results for SPAIs of the matrix bundle1 obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).

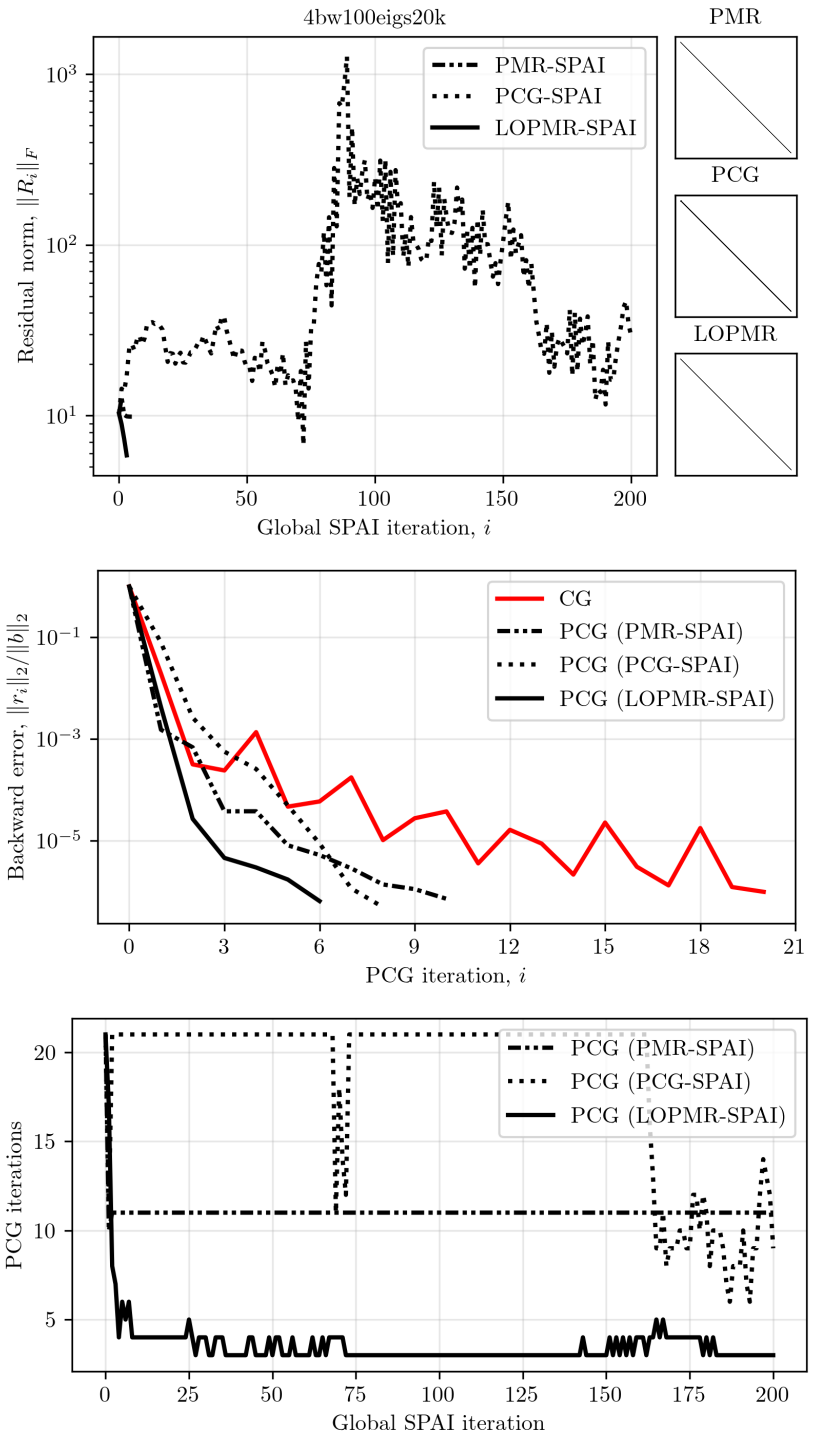

FIG. 6. Convergence plots and PCG results for SPAIs of the matrix 4bw100eigs20k obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).

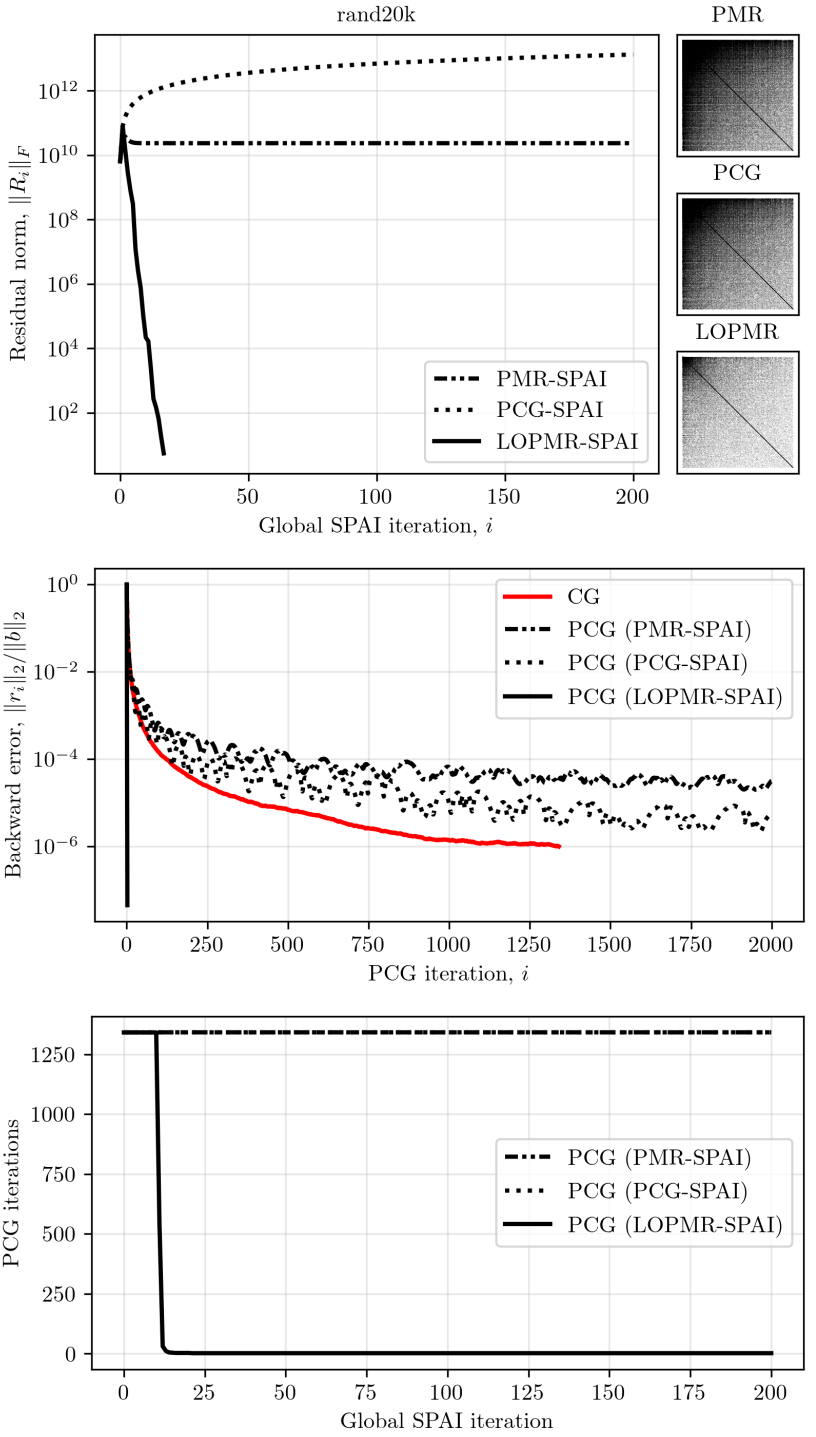

FIG. 7. Convergence plots and PCG results for SPAIs of the matrix rand20k obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).

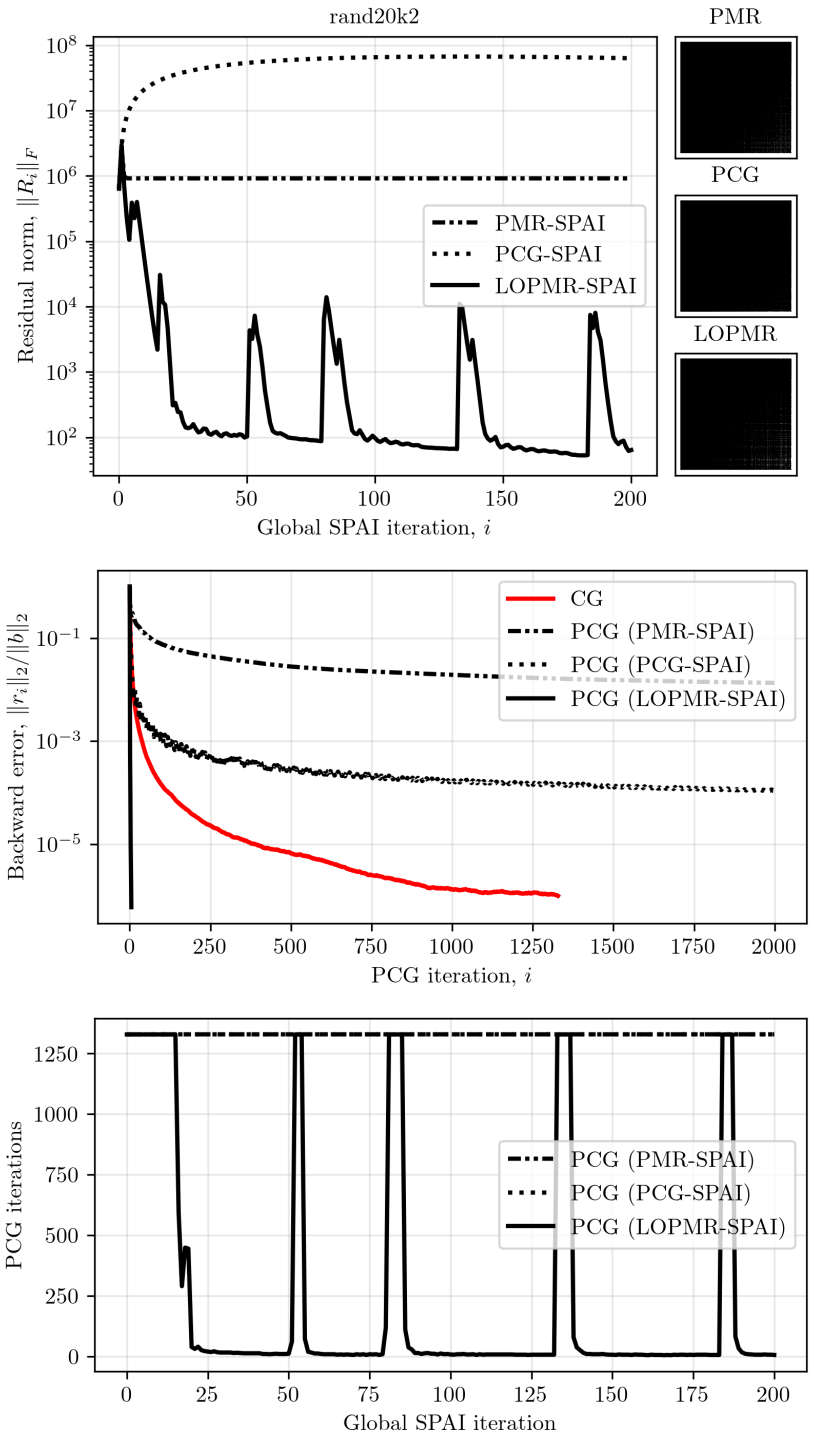

FIG. 8. Convergence plots and PCG results for SPAIs of the matrix rand20k2 obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).

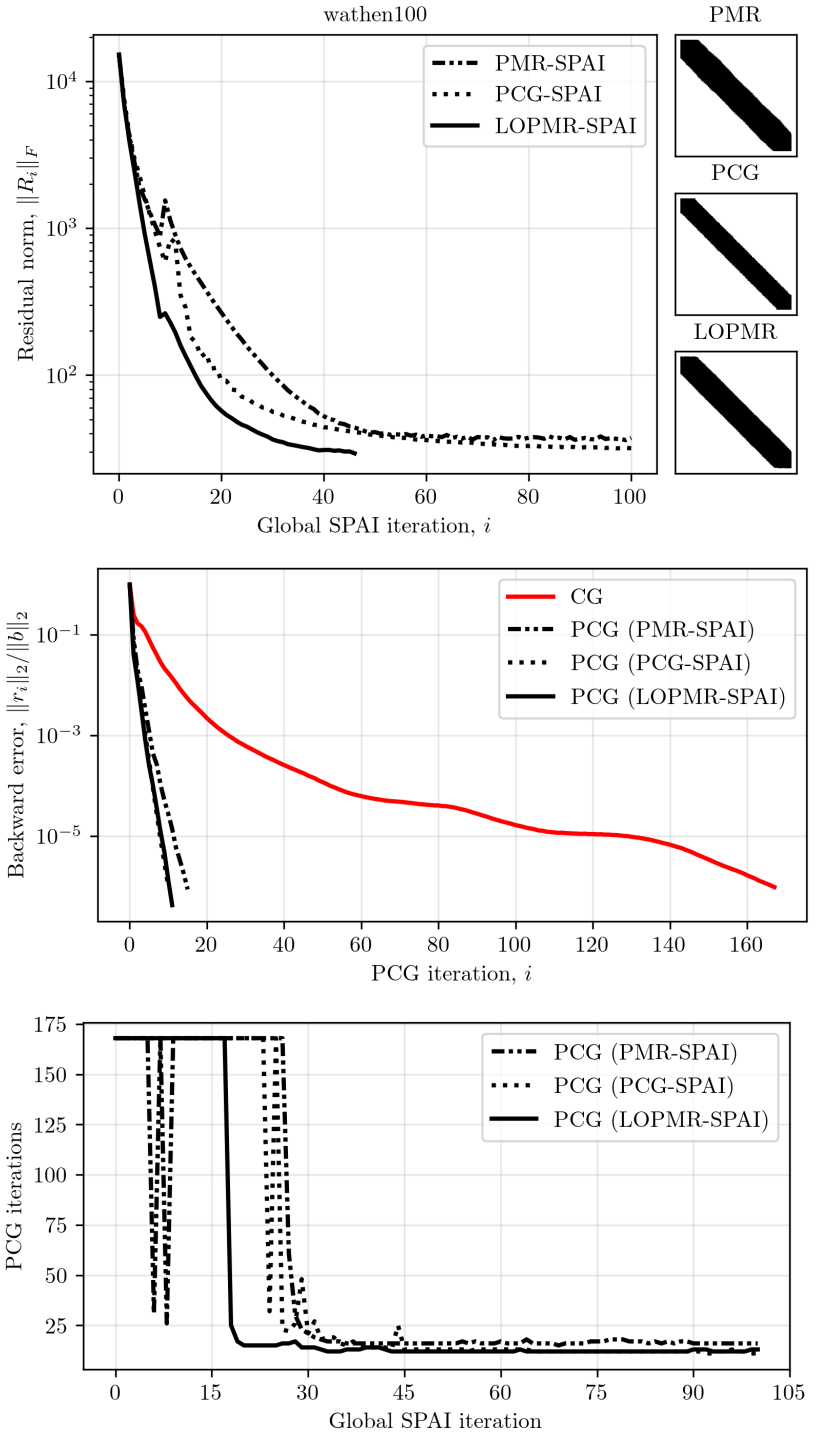

FIG. 9. Convergence plots and PCG results for SPAIs of the matrix wathen100 obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).

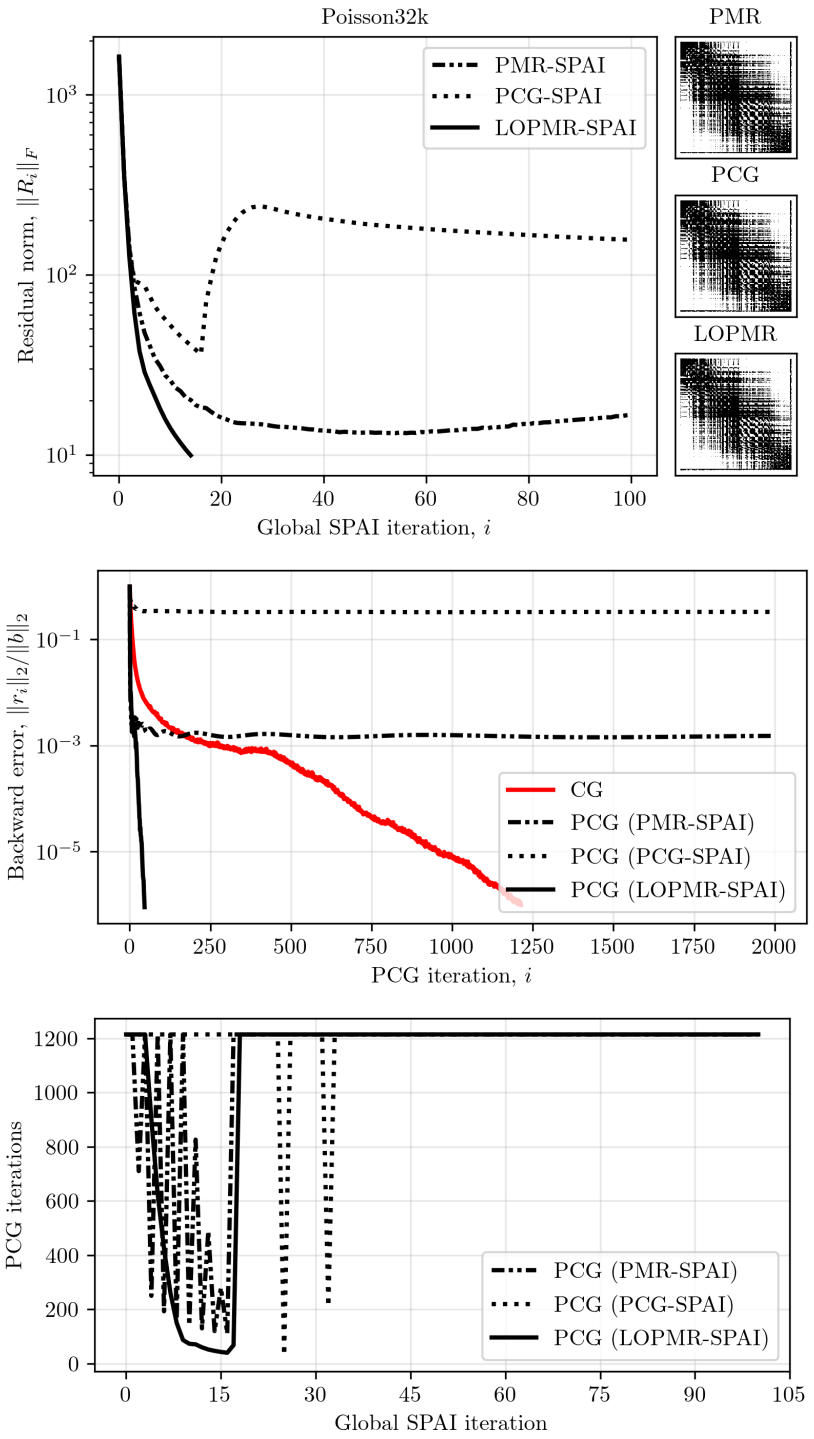

FIG. 10. Convergence plots and PCG results for SPAIs of the matrix Poisson32k obtained with dropping of nonzero values for maximum densities of 3% (Experiment05-07).