Les Cahiers du GERAD ISSN: 0711–2440

An efficient scaled spectral preconditioner for sequences of symmetric positive definite linear systems

Y. Diouane, S. Gürol, O. Mouhtal, D. Orban

G–2024–66

Septembre 2024

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

An efficient scaled spectral preconditioner for sequences of symmetric positive definite linear systems

Youssef Diouane [a]
Selime Gürol [b]
Oussama Mouhtal [a, b]
Dominique Orban [a]

[a] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[b] *CERFACS / CECI CNRS UMR 5318, Toulouse, France*

youssef.diouane@polymtl.ca
gurol@cerfacs.fr
mouhtal@cerfacs.fr
dominique.orban@gerad.ca

Septembre 2024
Les Cahiers du GERAD
G–2024–66

Abstract : We explore a *scaled* spectral preconditioner for the efficient solution of sequences of symmetric and positive-definite linear systems. We design the scaled preconditioner not only as an approximation of the inverse of the linear system but also with consideration of its use within the conjugate gradient (CG) method. We propose three different strategies for selecting a scaling parameter, which aims to position the eigenvalues of the preconditioned matrix in a way that reduces the energy norm of the error, the quantity that CG monotonically decreases at each iteration. Our focus is on accelerating convergence especially in the early iterations, which is particularly important when CG is truncated due to computational cost constraints. Numerical experiments provide in data assimilation confirm that the *scaled* spectral preconditioner can significantly improve early CG convergence with negligible computational cost.

Keywords : Sequence of linear systems, conjugate gradient method, deflated CG, spectral preconditioner, convergence rate, data assimilation

Résumé : Nous explorons la mise à l'échelle d'un préconditionneur spectral pour résoudre efficacement une suite de systèmes linéaires symétriques et définis positifs. La mise à l'échelle proposée du préconditionneur agit non seulement comme une approximation de l'inverse du système linéaire, mais elle prend également en compte l'utilisation du préconditionneur dans la méthode du gradient conjugué (CG). Nous proposons trois stratégies différentes pour la sélection d'un paramètre de mise à l'échelle. L'objectif est de positionner les valeurs propres de la matrice préconditionnée de manière à réduire la norme d'énergie de l'erreur, qui est la quantité minimisée par CG à chaque itération. La méthodologie proposée permet d'accélérer la convergence, notamment lors des premières itérations de CG, ce qui est particulièrement important lorsque le CG est arrêté prématurément en raison des contraintes de coût de calcul. Des expériences numériques en assimilation de données confirment que la mise à l'échelle du préconditionneur spectral améliore de manière significative la convergence initiale du CG, avec un coût de calcul négligeable.

Acknowledgements: This work was funded by French National Programme LEFE/INSU.

1 Introduction

Efficiently solving sequences of symmetric positive-definite (SPD) linear systems

$$A^{(j)}x^{(j)} = b^{(j)}, \quad j = 1, 2, \ldots \tag{1}$$

is crucial in various inverse problems of computational science and engineering. For instance, in data assimilation [4, 15], where one aims to solve a large-scale weighted regularized nonlinear least-squares problem via the truncated Gauss-Newton algorithm (GN) [11, 20], each iteration involves solving a linear least-squares subproblem. The latter may be formulated as a large SPD linear system, typically solved using the preconditioned conjugate-gradient method (PCG). Since consecutive systems do not differ significantly, recycling Krylov subspace information has been explored and proven to be effective [6, 10, 17, 19].

One way of recycling Krylov subspace information involves leveraging search directions obtained from PCG on earlier systems to construct a limited-memory quasi-Newton preconditioner (LMP) [17, 19]. This preconditioner, built solely from PCG information, does not require explicit knowledge of any matrix in the sequence, making it particularly suitable for applications where only matrix-vector products are available, which is the case of data assimilation. [10] generalizes this limited-memory preconditioner, and introduces specific variants when used with eigen- or Ritz pairs.

They focused on a first-level preconditioner, capable of clustering most eigenvalues at 1 with few outliers, is already available for the first linear system in sequence. Then, they used LMP as a second-level preconditioner to improve the efficiency of the first. The goal of the LMP is to capture directions in a low-dimensional subspace that the first-level preconditioner may miss, and use them to improve convergence of PCG. When $A^{(j)} = A$ for all j, spectral analysis of the preconditioned matrix when used with k pairs has shown that LMP can cluster at least k eigenvalues at 1, and that the eigenvalues of the preconditioned matrix interlace with those of the original matrix [10]. The efficiency of this approach has been demonstrated in a real-life data assimilation applications [10, 24].

We focus on improving the performance of the *spectral LMP* [7, 10], which is built by using eigenpairs of $A^{(j)}$. The spectral LMP shares the same formulation as the abstract balancing domain decomposition method [18] and is equivalent to deflation-based preconditioning when used with a specific initial point [24].

When designing preconditioners for PCG, the primary focus in the literature is mostly on A and the significance of the initial guess is overlooked. Although the importance of the initial guess is mentioned, its impact on the choice of a preconditioner is not well studied. Favorable eigenvalue distributions are also highlighted in terms of clustering, but there is little emphasis on the position of the clusters. The performance of the preconditioner is also measured in terms of the total number of iterations to converge, with little focus on the convergence in the early iterations. When PCG is truncated before convergence due to computational budget or when used as a solver within a optimization method like GN, the effect of the preconditioner on the early convergence of PCG is also crucial. In this paper, we aim to explore those overlooked aspects to design a good preconditioner. We not only aim to improve convergence by reducing the total number of iterations but also ensure that, from the very first iteration, the preconditioned iterates outperform those of the original system. In doing so, we specifically focus on strategically positioning the eigenvalues captured by the LMP, in that the energy norm of the error at each iteration of CG is reduced.

The paper is organized as follows. In Section 2 we start by introducing the necessary notation. In Section 3, we review PCG and its convergence properties. We then discuss the characteristics of an efficient preconditioner that can be applied to (1). Section 4 is our main contribution. We define the *scaled* spectral preconditioner and discuss its properties. Next, we outline three key approaches for selecting the scaling parameter, which influences the positioning of the eigenvalue cluster, to reduce total number of iterations and enhance convergence in the early iterations. In Section 5, we provide numerical experiments using the Lorenz 95 reference model from data assimilation to validate theoretical results. Finally, conclusions and perspectives are discussed in Section 6.

2 Notation

The matrix $A \in \mathbb{R}^{n\times n}$ is always SPD. Its spectral radius is $\rho(A)$. Its spectral decomposition is $A = S\Lambda S^\top$ with $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_1 \geq \ldots \geq \lambda_n > 0$, and $S = \begin{bmatrix} s_1 & \cdots & s_n \end{bmatrix}$ orthogonal. Its i-th eigenvalue is $\nu_i(A)$. Its range space is $\mathcal{R}(A)$. The A-norm, or *energy norm*, of vector x is $\|x\|_A = \sqrt{x^\top Ax}$. The spectral norm is $\|.\|_2$.

3 Background

3.1 CG algorithm

The Conjugate Gradient (CG) method [13] is the workhorse for $Ax = b$ with SPD $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. If $x_0 \in \mathbb{R}^n$ is an initial guess and $r_0 = b - Ax_0$ is the initial residual, then at every step $\ell = 1, 2, \ldots, n$, CG produces a unique approximation [22, p.176]

$$x_\ell \in x_0 + \mathcal{K}_\ell(A, r_0) \quad \text{such that} \quad r_\ell \perp \mathcal{K}_\ell(A, r_0), \tag{2}$$

which is equivalent [22, p.126] to

$$\|x^* - x_\ell\|_A = \min_{x\in x_0+\mathcal{K}_\ell(A,r_0)} \|x^* - x\|_A \,, \tag{3}$$

where x^* is the exact solution, $\mathcal{K}_\ell(A, r_0) := \operatorname{span}\{r_0, Ar_0, \ldots, A^{\ell-1}r_0\}$ is the ℓ-th Krylov subspace generated by A and r_0. In exact arithmetic, the method terminates in at most μ iterations, where μ is the grade of r_0 with respect to A, i.e., the maximum dimension of the Krylov subspace generated by A and r_0 [22]. The most popular and computationally efficient variant of (2) is the original formulation of [13], that recursively updates coupled 2-term recurrences for $x_{\ell+1}$, $r_{\ell+1}$, and the search direction $p_{\ell+1}$. Algorithm 1 states the complete algorithm. A common stopping criterion is based on sufficient decrease of the relative residual norm. However, in practical data assimilation implementations, a fixed number of iterations is used as stopping criterion due to computational budget constraints. CG is presented alongside its companion formulation, Algorithm 2, to be detailed in Section 3.3.

Algorithm 1 CG

```
$r_0 = b - Ax_0$

$\rho_0 = r_0^\top r_0$
$p_0 = r_0$
for $\ell = 0, 1, \ldots$ do
    $q_\ell = Ap_\ell$
    $\alpha_\ell = \rho_\ell/(q_\ell^\top p_\ell)$
    $x_{\ell+1} = x_\ell + \alpha_\ell p_\ell$
    $r_{\ell+1} = r_\ell - \alpha_\ell q_\ell$

    $\rho_{\ell+1} = r_{\ell+1}^\top r_{\ell+1}$
    $\beta_{\ell+1} = \rho_{\ell+1}/\rho_\ell$
    $p_{\ell+1} = r_{\ell+1} + \beta_{\ell+1} p_\ell$
end for
```

Algorithm 2 PCG

```
$\hat{r}_0 = b - A\hat{x}_0$
$z_0 = F\hat{r}_0$
$\hat{\rho}_0 = \hat{r}_0^\top z_0$
$\hat{p}_0 = z_0$
for $\ell = 0, 1, \ldots$ do
    $\hat{q}_\ell = A\hat{p}_\ell$
    $\hat{\alpha}_\ell = \hat{\rho}_\ell/(\hat{q}_\ell^\top \hat{p}_\ell)$
    $\hat{x}_{\ell+1} = \hat{x}_\ell + \hat{\alpha}_\ell \hat{p}_\ell$
    $\hat{r}_{\ell+1} = \hat{r}_\ell - \hat{\alpha}_\ell \hat{q}_\ell$
    $z_{\ell+1} = F\hat{r}_{\ell+1}$
    $\hat{\rho}_{\ell+1} = \hat{r}_{\ell+1}^\top z_{\ell+1}$
    $\hat{\beta}_{\ell+1} = \hat{\rho}_{\ell+1}/\hat{\rho}_\ell$
    $\hat{p}_{\ell+1} = z_{\ell+1} + \hat{\beta}_{\ell+1} \hat{p}_\ell$
end for
```

3.2 Convergence properties of CG

The approximation x_ℓ uniquely determined by (2) minimizes the error in the energy norm:

$$\|x^* - x_\ell\|_A^2 = \min_{p\in\mathbb{P}_\ell(0)} \|p(A)(x^* - x_0)\|_A^2 = \min_{p\in\mathbb{P}_\ell(0)} \sum_{i=1}^n p(\lambda_i)^2 \frac{\eta_i^2}{\lambda_i}, \tag{4}$$

where $\eta_i = s_i^\top r_0$ and $\mathbb{P}_\ell(0)$ is the set of polynomials of degree at most ℓ with value 1 at zero [22, p.193]. Thus, at each iteration, CG solves a certain weighted polynomial approximation problem over the discrete set $\{\lambda_1, \dots, \lambda_n\}$. Moreover, if $z_1^{(\ell)}, \dots, z_\ell^{(\ell)}$ are the ℓ roots of the solution p_ℓ^* to (4),

$$\|x^* - x_\ell\|_A^2 = \sum_{i=1}^n p_\ell^*(\lambda_i)^2 \frac{\eta_i^2}{\lambda_i} = \sum_{i=1}^n \prod_{j=1}^\ell \left(1 - \frac{\lambda_i}{z_j^{(\ell)}}\right)^2 \frac{\eta_i^2}{\lambda_i}. \tag{5}$$

The $z_j^{(\ell)}$ are the *Ritz values* [5]. From (5), if $z_j^{(\ell)}$ is close to a λ_i, we expect a significant reduction in the error in energy norm. Based on the above, [5] explains the rate of convergence of CG in terms of the convergence of the Ritz values to eigenvalues of A. Assuming that $\lambda_1, \dots, \lambda_n$ take on the r distinct values $\rho_1, \dots, \rho_r$, CG converges in at most r iterations [20, Theorem 5.4].

Using (4) and maximizing over the values $p(\lambda_i)$ [22, p.194] leads to

$$\frac{\|x^* - x_\ell\|_A}{\|x^* - x_0\|_A} \le \min_{p\in\mathbb{P}_\ell(0)} \max_{1\le i\le n} |p(\lambda_i)|. \tag{6}$$

By replacing $\{\lambda_1, \dots, \lambda_n\}$ with the interval $[\lambda_1, \lambda_n]$ and using Chebyshev polynomials, we obtain an upper bound [22, p.194]:

$$\frac{\|x^* - x_\ell\|_A}{\|x^* - x_0\|_A} \le 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^\ell, \tag{7}$$

where $\kappa(A) := \lambda_1/\lambda_n$ is the condition number of A. While (6) and (7) provide the worst-case behavior of CG [12], the convergence properties may vary significantly from the worst case for a specific initial approximation. Note also that upper bounds (6) and (7) only depend on A, and not on r_0. Though (7) relates the convergence behavior of CG to $\kappa(A)$, one should be careful as convergence is also influenced by the clustering of the eigenvalues and their positioning [2, 3].

3.3 Properties of a good preconditioner

In many practical applications, a preconditioner is essential for accelerating the convergence of CG [1, 25]. Assume that a preconditioner $F = UU^\top \in \mathbb{R}^{n\times n}$ is available in a factored form, where U is SPD, and consider the system with split preconditioner

$$U^\top AUy = U^\top b, \tag{8}$$

whose matrix is also SPD. System (8) can then be solved with CG. The latter updates estimate y_ℓ that can be used to recover $\hat{x}_\ell := Uy_\ell$. Algorithm 2, the preconditioned conjugate gradients method, is equivalent to the procedure just described, but only involves solves with F and does not assume knowledge of U [8, p.532]. PCG updates $\hat{x}_\ell$ directly.

PCG looks for an approximate solution in the Krylov subspace

$$x_0 + U\mathcal{K}_\ell(U^\top AU, U^\top r_0),$$

and as in (4), it minimizes the energy norm,

$$\|x^* - \hat{x}_\ell\|_A = \min_{q\in\mathbb{P}_\ell(0)} \|Uq(U^\top AU)U^{-1}(x^* - x_0)\|_A. \tag{9}$$

Although there is no general method for building a good preconditioner [1, 25], leveraging the convergence properties of CG on (9) often leads to the following criteria: (i) F should approximate the inverse of A, (ii) F should be cheap to apply, (iii) $\kappa(U^\top AU)$ should be smaller than $\kappa(A)$, and (iv) $U^\top AU$ should have a more favorable distribution of eigenvalues than A. Note that, all four criteria only focus on A and overlook the significance of the initial guess.

3.4 Preconditioning for a sequence of linear systems

In the context of (1), it is common to use a first level preconditioner, $F^{(1)}$, for the initial linear system, $A^{(1)}x^{(1)} = b^{(1)}$. The selection of the first-level preconditioner depends on the problem and may take into account both the physics of the problem and the algebraic structure of $A^{(1)}$ [1, 21, 25]. To further accelerate convergence of an iterative method such as PCG on subsequent linear systems $A^{(j+1)}x^{(j+1)} = b^{(j+1)}$, one can perform a low-rank update of the most-recent preconditioner, $F^{(j)}$, leveraging information obtained from solving $A^{(j)}x^{(j)} = b^{(j)}$ [10, 17].

One common choice of low-rank update is to use the (approximate) spectrum of $A^{(j)}$ [6, 7, 10]. The main idea is to capture the eigenvalues not captured by the first-level preconditioner, and cluster them to a positive quantity, typically around 1.

In this paper, we will consider the case where only the right-hand side is changing over the sequence of the linear systems, i.e., $A^{(j)} = A$ for all j. Perturbation analysis with respect to A will be presented in a forthcoming paper.

4 A scaled spectral preconditioner

We focus on the scaled spectral preconditioner, known in the literature as the deflating preconditioner [7] or spectral Limited Memory Preconditioner (LMP) [10], which is defined using a scaling parameter that determines the positioning of the cluster. We will provide several strategies for the choice of the scaling parameter, which has a significant impact on the convergence of PCG.

Let us assume that k largest eigenvalues of A, i.e. $\{\lambda_i\}_{i=1}^k$, are available. We define the spectral preconditioner

$$F_\theta := I_n + \sum_{i=1}^{k} \left(\frac{\theta}{\lambda_i} - 1\right) s_i s_i^\top = I_n + S_k(\theta\Lambda_k^{-1} - I_k)S_k^\top = S \begin{bmatrix} \theta\Lambda_k^{-1} & \\ & I_{n-k} \end{bmatrix} S^\top, \tag{10}$$

where $S_k := \begin{bmatrix} s_1 & \cdots & s_k \end{bmatrix}$ and $\Lambda_k := \operatorname{diag}(\lambda_1, \dots, \lambda_k)$. The factor of $F_\theta = U_\theta^2$ is

$$U_\theta = U_\theta^\top := I_n + \sum_{i=1}^{k} \left(\sqrt{\frac{\theta}{\lambda_i}} - 1\right) s_i s_i^\top = S \begin{bmatrix} \sqrt{\theta}\Lambda_k^{-\frac{1}{2}} & \\ & I_{n-k} \end{bmatrix} S^\top. \tag{11}$$

Preconditioner F_θ clusters $\lambda_1, \dots, \lambda_k$ around θ, and leaves the rest of the spectrum untouched, i.e.,

$$U_\theta A U_\theta = S \begin{bmatrix} \theta I_k & \\ & \bar{\Lambda}_k \end{bmatrix} S^\top = \theta S_k S_k^\top + \bar{S}_k \bar{\Lambda}_k \bar{S}_k^\top, \tag{12}$$

where $\bar{S}_k := \begin{bmatrix} s_{k+1} & \cdots & s_n \end{bmatrix}$ and $\bar{\Lambda}_k := \operatorname{diag}(\lambda_{k+1}, \dots, \lambda_n)$. As in (9), PCG minimizes

$$\begin{aligned} \|x^* - \hat{x}_\ell(\theta)\|_A &= \min_{q \in \mathbb{P}_\ell(0)} \|U_\theta q\,(U_\theta A U_\theta)\, U_\theta^{-1}\,(x^* - x_0)\|_A \\ &= \min_{q \in \mathbb{P}_\ell(0)} \|q\,(U_\theta A U_\theta)\,(x^* - x_0)\|_A, \end{aligned} \tag{13}$$

where we used $U_\theta q\,(U_\theta A U_\theta)\, U_\theta^{-1} = U_\theta U_\theta^{-1} q\,(U_\theta A U_\theta) = q\,(U_\theta A U_\theta)$. Using (4) in the context of Equation (13), we obtain the following result.

Theorem 1. *Let $\hat{x}_\ell(\theta)$ be generated at iteration ℓ of Algorithm 2 applied to $Ax = b$ with preconditioner (10). Then,*

$$\|x^* - \hat{x}_\ell(\theta)\|_A^2 = \min_{q \in \mathbb{P}_\ell(0)} \sum_{i=1}^{k} \frac{\eta_i^2}{\lambda_i} q(\theta)^2 + \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q(\lambda_i)^2, \tag{14}$$

where $\eta_i = s_i^\top r_0$ is the i-th component of the initial residual in the basis S.

Proof. Given (12), we have for any polynomial q,

$$q\left(U_\theta A U_\theta\right) = Sq\left(\begin{bmatrix} \theta I_k & \\ & \bar{\Lambda}_k \end{bmatrix}\right) S^\top.$$

Since $x^* - x_0 = A^{-1} r_0 = S \Lambda^{-1} S^\top r_0$,

$$q\left(U_\theta A U_\theta\right)(x^* - x_0) = Sq\left(\begin{bmatrix} \theta I_k & \\ & \bar{\Lambda}_k \end{bmatrix}\right) \Lambda^{-1} S^\top r_0. \tag{15}$$

Substituting Equation (15) into Equation (13), we obtain the result. □

The scaled LMP Equation (11) is typically used with $\theta = 1$. This choice is operational in numerical weather forecast [6, 24]. In the next subsections, we explore various choices for θ aiming to improve convergence properties of PCG.

4.1 On the choice of the scaling parameter

The scaling parameter θ, which defines the position of the cluster, is often set to 1 [6, 7, 10]. This choice is motivated by several factors, such as the eigenvalue distribution of A, the behavior of the first-level preconditioner, and the convergence behavior of PCG.

We investigate clustering the eigenvalues at a general $\theta > 0$, which, compared with the conventional choice of 1, results in enhanced convergence of PCG. It is important to note that the notion of "better convergence" may vary across different applications. For instance, in some applications, one may require high accuracy, in which case, a better convergence may be defined as a lower number of iterations. In other applications, we may want to get an approximate solution quickly, which requires to improve the convergence especially in the early iterations. In this case, there is no guarantee that the early preconditioned iterates will provide a better reduction in the energy norm compared to the unpreconditioned iterates (Section 4.2). For certain applications, such as numerical weather forecast, where PCG is stopped before reaching convergence due to computational budget, early convergence properties could be of critical importance. As a first direction, we will focus on the following question:

> *Is there $\theta > 0$ such that for any x_0,*
>
> $$\|x^* - \hat{x}_\ell(\theta)\|_A \leq \|x^* - x_\ell\|_A, \quad \ell = 1, \ldots, n? \tag{16}$$

To accelerate early convergence, we will investigate optimal choices for θ with respect to the error in the energy norm at the first iteration of PCG, i.e.,

$$\min_\theta \Phi(\theta) := \|x^* - \hat{x}_1(\theta)\|_A^2.$$

We focus solely on the first iteration as it allows us to derive the optimal value of θ in closed form.

On the other hand, for PCG, it is well known that removing eigenvalues causing convergence delay can improve the convergence rate significantly [6, 10]. This can be done by using deflation techniques, in which the aim is to "hide" (problematic) parts of the spectrum of A from PCG, so that the convergence rate of PCG is improved [14, 23]. Finally, our focus will be also on answering the question

> *Can we choose $\theta > 0$ such that for any x_0, PCG generates iterates close to those of deflation techniques?*

4.2 θ providing lower error in energy norm

In general, although scaled spectral preconditioning is expected to help reduce the number of iterations required to achieve convergence, Equation (16) may not hold for any choice of $\theta > 0$ and all iterations ℓ as given by the following proposition.

Proposition 1. *Let x_1 be the first CG iterate when solving $Ax = b$. Let $\hat{x}_1(\theta)$ be generated at the first iteration of Algorithm 2 applied to $Ax = b$ with preconditioner (10). Let x_0 be such that $\eta_i^2 = \lambda_i$ for $i = k, k+1$ and $\eta_i = 0$ otherwise. Then,*

$$\|x^* - \hat{x}_1(\theta)\|_A^2 \leq \|x^* - x_1\|_A^2 \iff \frac{\lambda_{k+1}^2}{\lambda_k} \leq \theta \leq \lambda_k.$$

Proof. For $\ell = 1$, (5) yields $\|x^* - x_1\|_A^2 = p_1^*(\lambda_k)^2 + p_1^*(\lambda_{k+1})^2$, where

$$p_1^*(\lambda) = 1 - \frac{r_0^\top r_0}{r_0^\top A r_0}\lambda = 1 - \frac{\lambda_k + \lambda_{k+1}}{\lambda_k^2 + \lambda_{k+1}^2}\lambda.$$

Similarly, (14) gives $\|x^* - \hat{x}_1(\theta)\|_A^2 = q_{1,\theta}^*(\theta)^2 + q_{1,\theta}^*(\lambda_{k+1})^2$, where

$$q_{1,\theta}^*(\lambda) = 1 - \frac{r_0^\top F_\theta r_0}{r_0^\top F_\theta A F_\theta r_0}\lambda = 1 - \frac{\theta + \lambda_{k+1}}{\theta^2 + \lambda_{k+1}^2}\lambda$$

is the polynomial that realizes the minimum. Using these relations, we obtain

$$\|x^* - x_1\|_A^2 = \left(1 - \frac{\lambda_k + \lambda_{k+1}}{\lambda_k^2 + \lambda_{k+1}^2}\lambda_k\right)^2 + \left(1 - \frac{\lambda_k + \lambda_{k+1}}{\lambda_k^2 + \lambda_{k+1}^2}\lambda_{k+1}\right)^2 = \frac{(\lambda_k - \lambda_{k+1})^2}{\lambda_k^2 + \lambda_{k+1}^2}$$

and

$$\|x^* - \hat{x}_1(\theta)\|_A^2 = \left(1 - \frac{\theta + \lambda_{k+1}}{\theta^2 + \lambda_{k+1}^2}\theta\right)^2 + \left(1 - \frac{\theta + \lambda_{k+1}}{\theta^2 + \lambda_{k+1}^2}\lambda_{k+1}\right)^2 = \frac{(\theta - \lambda_{k+1})^2}{\theta^2 + \lambda_{k+1}^2}.$$

Hence,

$$\frac{(\theta - \lambda_{k+1})^2}{\theta^2 + \lambda_{k+1}^2} \leq \frac{(\lambda_k - \lambda_{k+1})^2}{\lambda_k^2 + \lambda_{k+1}^2} \iff \frac{\lambda_{k+1}^2}{\lambda_k} \leq \theta \leq \lambda_k. \qquad \square$$

Proposition 1 shows that Equation (16) is not satisfied for all $\theta > 0$. If $\theta > 0$ lies outside of $[\lambda_{k+1}^2/\lambda_k, \lambda_k]$, then $\|x^* - \hat{x}_1(\theta)\|_A > \|x^* - x_1\|_A$ for x_0 as defined in Proposition 1.

In what comes next, we focus on the properties of θ such that Equation (16) is guaranteed for all iterations ℓ, and for any given x_0. An intuitive approach is to identify a range of θ values where the eigenvalue ratios of the preconditioned matrix are less than or equal to those of the unpreconditioned matrix, as noted in [12, Lemma 1]. The next lemma shows that this property holds for $\theta \in [\lambda_{k+1}, \lambda_k]$, and for such choice, there exists a polynomial that promotes favorable PCG convergence.

Lemma 1. *Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$, $\ell \in \{1, \ldots, n\}$, and $k \in \{1, \ldots, \ell\}$. For any $\theta \in [\lambda_{k+1}, \lambda_k]$, and any polynomial p of degree ℓ such that $p(0) = 1$ and whose roots all lie in $[\lambda_n, \lambda_1]$, there exists a polynomial q of degree ℓ such that $q(0) = 1$ and*

$$\begin{aligned} |q(\theta)| &\leq |p(\lambda_i)|, \quad i = 1, \ldots, k \\ |q(\lambda_i)| &\leq |p(\lambda_i)|, \quad i = k+1, \ldots, n. \end{aligned}$$

Proof. Let us denote $(\mu_j)_{1 \leq j \leq \ell}$ the roots of the polynomial p given in decreasing order, so $p(\lambda) = \prod_{i=1}^{\ell}\left(1 - \frac{\lambda}{\mu_i}\right)$ for any $\lambda \geq 0$. Then, three cases may occur:

Case 1: For all $j \in \{1,\dots,\ell\}$, $\mu_j < \theta$, we choose $q(\lambda) = p(\lambda)$, then simply we have for $i \in \{k+1,\dots,n\}$, $|q(\lambda_i)| = |p(\lambda_i)|$. For $i \in \{1,\dots,k\}$, using the property that $\mu_j < \theta \le \lambda_i$, we obtain

$$1 - \frac{\lambda_i}{\mu_j} \le 1 - \frac{\theta}{\mu_j} \le 0.$$

Thus, we have $|1 - \frac{\theta}{\mu_j}| \le |1 - \frac{\lambda_i}{\mu_j}|$, and consequently $|q(\theta)| \le |p(\lambda_i)|$.

Case 2: If for all $j \in \{1,\dots,\ell\}$, $\theta \le \mu_j$, we choose $q(\lambda) = \prod_{j=1}^{\ell}\left(1 - \frac{\lambda}{\theta}\right) = \left(1 - \frac{\lambda}{\theta}\right)^l$. Then simply for $i \in \{1,\dots,k\}$, $|q(\theta)| = 0 \le |p(\lambda_i)|$. For $i \in \{k+1,\dots,n\}$, using the property $\lambda_{k+1} \le \theta \le \mu_j$, we obtain

$$0 \le 1 - \frac{\lambda_i}{\lambda_{k+1}} \le 1 - \frac{\lambda_i}{\theta} \le 1 - \frac{\lambda_i}{\mu_j}.$$

Therefore, for $i = k+1,\dots,n$, $|q(\lambda_i)| \le |p(\lambda_i)|$.

Case 3: let $s \in \{1,\dots,\ell-1\}$ such that for $j = 1,\dots,s$, $\theta \le \mu_j \le \lambda_1$, and for $j = s+1,\dots,\ell$, $\lambda_n \le \mu_j < \theta$. Let's denote

$$q(\lambda) = \prod_{j=1}^{s}\left(1 - \frac{\lambda}{\theta}\right)\prod_{j=s+1}^{\ell}\left(1 - \frac{\lambda}{\mu_j}\right) = \left(1 - \frac{\lambda}{\theta}\right)^s \prod_{j=s+1}^{\ell}\left(1 - \frac{\lambda}{\mu_j}\right).$$

We have $q(\theta) = 0$, so $|q(\theta)| \le |p(\lambda_i)|$ for $i \in \{1,\dots,k\}$. For $i \in \{k+1,\dots,n\}$ and $j \in \{1,\dots,s\}$, we have

$$0 \le 1 - \frac{\lambda_i}{\lambda_{k+1}} \le 1 - \frac{\lambda_i}{\theta} \le 1 - \frac{\lambda_i}{\mu_j},$$

because $\lambda_{k+1} \le \theta \le \mu_j$. Therefore, for $i = k+1,\dots,n$, $|q(\lambda_i)| \le |p(\lambda_i)|$. □

Now, we can present a result that enables comparing the error in energy norm between the preconditioned system given by (8) and the unpreconditioned system, $Ax = b$.

Theorem 2. *Let $(x_\ell)_{\ell\in\{1,\dots,n\}}$ and $\hat{x}_\ell(\theta)_{\ell\in\{1,\dots,n\}}$ be the sequences generated by CG and PCG with F_θ with $\theta \in [\lambda_{k+1}, \lambda_k]$, respectively, when solving $Ax = b$. Assume that $\hat{x}_0(\theta) = x_0$. Then, for all $\ell = 1,\dots,n$, $\|x^* - \hat{x}_\ell(\theta)\|_A \le \|x^* - x_\ell\|_A$.*

Proof. Let $\ell \in \{1,\dots,n\}$. From (5),

$$\|x^* - x_\ell\|_A^2 = \min_{p\in\mathbb{P}_\ell(0)} \|p_\ell(A)(x^* - x_0)\|_A^2 = \sum_{i=1}^{n}\frac{\eta_i^2}{\lambda_i}p_\ell^*(\lambda_i)^2, \tag{17}$$

where η_i represents the components of the initial residual $r_0 = b - Ax_0$ in the eigenspace of A. Applying Lemma 1 to p_ℓ^*, there exists a polynomial q of degree ℓ with $q(0) = 1$ such that

$$\begin{aligned} |q(\theta)| &\le |p_\ell^*(\lambda_i)|, \quad i \in \{1,\dots,k\} \\ |q(\lambda_i)| &\le |p_\ell^*(\lambda_i)|, \quad i \in \{k+1,\dots,n\}. \end{aligned}$$

Applying these inequalities to (17) yields

$$\begin{aligned} \|x^* - x_\ell\|_A^2 = \sum_{i=1}^{n}\frac{\eta_i^2}{\lambda_i}p_\ell^*(\lambda_i)^2 &\ge \sum_{i=1}^{k}\frac{\eta_i^2}{\lambda_i}q(\theta)^2 + \sum_{i=k+1}^{n}\frac{\eta_i^2}{\lambda_i}q(\lambda_i)^2 \\ &\ge \min_{q\in\mathbb{P}_\ell(0)}\left(\sum_{i=1}^{k}\frac{\eta_i^2}{\lambda_i}q(\theta)^2 + \sum_{i=k+1}^{n}\frac{\eta_i^2}{\lambda_i}q(\lambda_i)^2\right) = \|x^* - \hat{x}_\ell(\theta)\|_A^2. \end{aligned}$$

□

Theorem 2 offers a range of choices for θ. Next, we discuss the practical and theoretical choices from this range. Let us remind that to construct the spectral LMP (11), we are given k eigenpairs. As a result, one practical choice is $\theta = \lambda_k$. This idea is summarized in the following corollary.

Corollary 1. *Let $\theta = \lambda_k$. Then, $\|x^* - \hat{x}_\ell(\lambda_k)\|_A \leq \|x^* - x_\ell\|_A$ for any x_0 and for all $\ell \in \{1, \dots, n\}$.*

The next theorem shows that increasing k results in improved convergence.

Theorem 3. *Let $1 < k_1 \leq k_2 < n$ and $\theta_{k_1} \in [\lambda_{k_1+1}, \lambda_{k_1}]$, $\theta_{k_2} \in [\lambda_{k_2+1}, \lambda_{k_2}]$ with, $\theta_{k_2} \leq \theta_{k_1}$. Let $(\hat{x}_\ell(\theta_{k_1}))_{\ell \in \{1,\dots,n\}}$, $(\hat{x}_\ell(\theta_{k_2}))_{\ell \in \{1,\dots,n\}}$ be the sequences obtained from PCG iterates when solving $Ax = b$ using $F_{\theta_{k_1}}$ and $F_{\theta_{k_2}}$ respectively with an arbitrary initial guess x_0. Then, for all $\ell \in \{1, \dots, n\}$, one has:*

$$\|x^* - \hat{x}_\ell(\theta_{k_2})\|_A \leq \|x^* - \hat{x}_\ell(\theta_{k_1})\|_A .$$

Proof. The eigenvalues of the preconditioned matrix using $F_{\theta_{k_1}}$ and $F_{\theta_{k_2}}$ are given in decreasing order respectively as

$$\rho_i = \begin{cases} \theta_{k_1} & i \in \{1, \dots, k_1\} \\ \lambda_i & \text{otherwise,} \end{cases} \quad \text{and} \quad \widetilde{\rho}_i = \begin{cases} \theta_{k_2} & i \in \{1, \dots, k_2\} \\ \lambda_i & \text{otherwise.} \end{cases}$$

As $k_1 < k_2$, it follows that $\widetilde{\rho}_{k_2} \leq \rho_{k_1} = \theta_{k_1}$. Therefore, $\widetilde{\rho}_i$ can be expressed as a function of ρ_i as

$$\widetilde{\rho}_i = \begin{cases} \theta_{k_2} \in [\rho_{k_2+1}, \rho_{k_2}] & i \in \{1, \dots, k_2\} \\ \rho_i & \text{otherwise.} \end{cases}$$

Using Lemma 1, for the polynomial $q^*_{\ell,\theta_{k_1}}$, there exists a polynomial q of degree ℓ with $q(0) = 1$, such that for $i \in \{1, \dots, n\}$,

$$\begin{aligned} |q(\theta_{k_2})| &\leq |q^*_{\ell,\theta_{k_1}}(\rho_i)|, \quad i \in \{1, \dots, k_2\} \\ |q(\rho_i)| &\leq |q^*_{\ell,\theta_{k_1}}(\rho_i)|, \quad i \in \{k_2 + 1, \dots, n\} \end{aligned}$$

Applying this result to (14) yields that

$$\begin{aligned} \|x^* - \hat{x}_\ell(\theta_{k_1})\|_A^2 &= \sum_{i=1}^{n} \frac{\eta_i^2}{\lambda_i} q^*_{\ell,\theta_{k_1}}(\rho_i)^2 \\ &\geq \sum_{i=1}^{k_2} \frac{\eta_i^2}{\lambda_i} q(\theta_{k_2})^2 + \sum_{i=k_2+1}^{n} \frac{\eta_i^2}{\lambda_i} q(\rho_i)^2 \\ &\geq \min_{q \in \mathbb{P}_\ell(0)} \left(\sum_{i=1}^{k_2} \frac{\eta_i^2}{\lambda_i} q(\theta_{k_2})^2 + \sum_{i=k_2+1}^{n} \frac{\eta_i^2}{\lambda_i} q(\lambda_i)^2 \right) = \|x^* - \hat{x}_\ell(\theta_{k_2})\|_A^2 . \end{aligned}$$ □

One can see that $k_1 < k_2 \implies \theta_{k_2} \leq \theta_{k_1}$, since λ_i are in decreasing order. In addition, when $k_1 = k_2$, Theorem 3 shows that λ_{k_1+1} is the best choice in $[\lambda_{k_1+1}, \lambda_{k_1}]$ in terms of reducing the error with respect to the unpreconditioned system.

4.3 Optimal choice for θ with respect to the initial residual

Our objective is to determine the value of θ that minimizes the energy norm of the error at the initial iterate. This will provide us with the optimal reduction at the first iterate,

$$\theta_r \in \arg\min_{\theta > 0} \Phi(\theta) := \|x^* - \hat{x}_1(\theta)\|_A^2 . \tag{18}$$

The expression for θ_r is stated in the following theorem.

Theorem 4. *Let $r_0 = b - Ax_0$. The unique $\lambda_n \leq \theta_r \leq \lambda_{k+1}$ satisfying (18) is*

$$\theta_r := \frac{\sum_{i=k+1}^n \lambda_i \eta_i^2}{\sum_{i=k+1}^n \eta_i^2} = \frac{r_0^\top A r_0 - r_0 S_k \Lambda_k S_k^\top r_0}{r_0^\top r_0 - r_0^\top S_k S_k^\top r_0}. \tag{19}$$

Proof. First, Theorem 1 implies

$$\|x^* - \hat{x}_1(\theta)\|_A^2 = \sum_{i=1}^k \frac{\eta_i^2}{\lambda_i} q_{1,\theta}^*(\theta)^2 + \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i} q_{1,\theta}^*(\lambda_i)^2 \tag{20}$$

where $\eta_i = s_i^\top r_0$ and $q_{1,\theta}^*(\lambda) = 1 - \dfrac{r_0^\top F_\theta r_0}{r_0^\top F_\theta A F_\theta r_0}\lambda$. Using (10), we obtain

$$r_0^\top F_\theta r_0 = \theta \sum_{i=1}^k \frac{\eta_i^2}{\lambda_i} + \sum_{i=k+1}^n \eta_i^2 \quad \text{and} \quad r_0^\top F_\theta A F_\theta r_0 = \theta^2 \sum_{i=1}^k \frac{\eta_i^2}{\lambda_i} + \sum_{i=k+1}^n \lambda_i \eta_i^2. \tag{21}$$

Then, for all $\theta > 0$, $\Phi(\theta)$ simplifies to

$$\Phi(\theta) = a_1 \left(\frac{a_2\theta - a_3}{a_1\theta^2 + a_3} \right)^2 + \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i} \left(1 - \frac{a_1\theta + a_2}{a_1\theta^2 + a_3}\lambda_i \right)^2,$$

where $a_1 = \sum_{i=1}^k \dfrac{\eta_i^2}{\lambda_i}$, $a_2 = \sum_{i=k+1}^n \eta_i^2$ and $a_3 = \sum_{i=k+1}^n \lambda_i \eta_i^2$. The derivative of Φ is

$$\Phi'(\theta) = \frac{2a_1}{(a_1\theta^2 + a_3)^3}(a_2\theta - a_3)(a^2\theta^3 + a_1 a_2 \theta^2 + a_1 a_3 \theta + a_2 a_3).$$

Since $\Phi'(\theta) < 0$ on $]0, \frac{a_3}{a_2}[$ and $\Phi'(\theta) > 0$ on $]\frac{a_3}{a_2}, +\infty[$, then $\frac{a_3}{a_2}$ is the global minimizer of Φ on $\mathbb{R}_+^*$ and is unique. Hence,

$$\theta_r = \arg\min_{\theta > 0} \Phi(\theta) = \frac{a_3}{a_2} = \frac{\sum_{i=k+1}^n \lambda_i \eta_i^2}{\sum_{i=k+1}^n \eta_i^2}.$$

Moreover,

$$\lambda_n = \frac{\sum_{i=k+1}^n \lambda_n \eta_i^2}{\sum_{i=k+1}^n \eta_i^2} \leq \theta_r \leq \frac{\sum_{i=k+1}^n \lambda_i \eta_i^2}{\sum_{i=k+1}^n \eta_i^2} = \lambda_{k+1}.$$

The expression for θ_r can be rewritten in terms of S_k, Λ_k, and r_0 as follows:

$$\theta_r = \frac{\sum_{i=1}^n \lambda_i \eta_i^2 - \sum_{i=1}^k \lambda_i \eta_i^2}{\sum_{i=1}^n \eta_i^2 - \sum_{i=1}^k \eta_i^2} = \frac{r_0^\top A r_0 - r_0 S_k \Lambda_k S_k^\top r_0}{r_0^\top r_0 - r_0^\top S_k S_k^\top r_0}. \qquad \square$$

Note that θ_r can be interpreted as the center of mass for the remaining part of the spectrum in which the weights are determined by η_i^2, i.e.

$$\sum_{i=k+1}^n \eta_i^2 (\theta_r - \lambda_i) = 0.$$

Let us now look at the first iterate,

$$\hat{x}_1(\theta_r) = x_0 + \frac{r_0^\top F_{\theta_r} r_0}{r_0^\top F_{\theta_r} A F_{\theta_r} r_0} F_{\theta_r} r_0, \tag{22}$$

to better understand the effect of θ_r. Using (21) and the value of θ_r,

$$\frac{r_0^\top F_{\theta_r} r_0}{r_0^\top F_{\theta_r} A F_{\theta_r} r_0} = \frac{\sum_{i=k+1}^n \eta_i^2}{\sum_{i=k+1}^n \lambda_i \eta_i^2} = \frac{1}{\theta_r}.$$

Therefore, (22) simplifies to

$$\hat{x}_1(\theta_r) = x_0 + \frac{1}{\theta_r}\left(\bar{S}_k \bar{S}_k^\top + \theta_r S_k \Lambda_k^{-1} S_k^\top\right) r_0 = x_0 + S_k \Lambda_k^{-1} S_k^\top r_0 + \frac{1}{\theta_r} \bar{S}_k \bar{S}_k^\top r_0.$$

Then, the residual of the first iteration is given by

$$b - A\hat{x}_1(\theta_r) = r_0 - S_k S_k^\top r_0 - \frac{1}{\theta_r} \bar{S}_k \bar{\Lambda}_k \bar{S}_k^\top r_0 = \bar{S}_k \bar{S}_k^\top r_0 - \frac{1}{\theta_r} \bar{S}_k \bar{\Lambda}_k \bar{S}_k^\top r_0. \tag{23}$$

Given (23), we conclude that, from the first iteration, we can remove all components of the residual with respect to S_k, see Appendix A. We now provide an upper bound for the error in the energy norm for later iterations, $\ell > 1$, beginning with $\hat{x}_1(\theta_r)$. With this initial point, we ensure that all iterates yield a residual within $\operatorname{Span}(\bar{S}_k)$.

Theorem 5. *Let $\hat{x}_\ell(\theta_r)$ be the ℓ-th iterate obtained from PCG when solving $Ax = b$ using the preconditioner F_{θ_r} with an arbitrary initial guess x_0. Let x_ℓ^{Init} be the ℓ-th iterate generated by CG for solving $Ax = b$ starting from $\hat{x}_1(\theta_r)$ as defined in (22). Then, for all $\ell \in \{1, \ldots, n\}$, $\|x^* - \hat{x}_{\ell+1}(\theta_r)\|_A \leq \|x^* - x_\ell^{Init}\|_A$.*

Proof. From (23), the components of $b - A\hat{x}_1(\theta_r)$ in the eigenspace of A are

$$0 \quad (i = 1, \ldots, k), \quad \text{and} \quad \eta_i(1 - \lambda_i/\theta_r) \quad (i > k).$$

Thus,

$$\|x - x_\ell^{\text{Init}}\|_A^2 = \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i}\left(1 - \frac{\lambda_i}{\theta_r}\right)^2 p_\ell^{*,\text{Init}}(\lambda_i)^2, \tag{24}$$

where $p_\ell^{*,\text{Init}}$ is the polynomial that minimizes $p \mapsto \|p(A)(x^* - \hat{x}_1(\theta_r))\|_A^2$ over $\mathbb{P}_\ell(0)$.
Define

$$\bar{q}(\lambda) = \left(1 - \frac{\lambda}{\theta_r}\right) p_\ell^{*,\text{Init}}(\lambda),$$

and note that $\bar{q} \in \mathbb{P}_\ell(0)$. Now we have

$$\begin{aligned}
\|x^* - \hat{x}_{\ell+1}(\theta_r)\|_A^2 &= \min_{q \in \mathbb{P}_{\ell+1}(0)} \left(\sum_{i=1}^k \frac{\eta_i^2}{\lambda_i} q(\theta_r)^2 + \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i} q(\lambda_i)^2\right) \\
&\leq \sum_{i=1}^k \frac{\eta_i^2}{\lambda_i} \bar{q}(\theta_r)^2 + \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i} \bar{q}(\lambda_i)^2 \\
&= \sum_{i=k+1}^n \frac{\eta_i^2}{\lambda_i}\left(1 - \frac{\lambda_i}{\theta_r}\right)^2 p_\ell^{*,\text{Init}}(\lambda_i)^2 = \|x - x_\ell^{\text{Init}}\|_A^2.
\end{aligned}$$

□

Note that, one can interpret $\hat{x}_1(\theta_r)$ as the first iteration of CG when solving the unpreconditioned system, starting from $x_0 + S_k \Lambda_k^{-1} S_k^\top r_0$, since the search direction at the first iteration is equal to:

$$b - A\left(x_0 + S_k \Lambda_k^{-1} S_k^\top r_0\right) = b - Ax_0 - S_k S_k^\top r_0 = r_0 - S_k S_k^\top r_0 = \bar{S}_k \bar{S}_k^\top r_0, \tag{25}$$

and the step-length α_0 is given as

$$\alpha_0 = \frac{1}{\theta_r} = \frac{r_0^\top \bar{S}_k \bar{S}_k^\top r_0}{r_0^\top \bar{S}_k^\top \bar{S}_k A \bar{S}_k \bar{S}_k^\top r_0}.$$

This highlights the strong connection between preconditioning, CG with different initial point and deflation techniques [23, 24]. This connection will be explored in detail in the next subsection, providing another choice for the scaling parameter.

4.4 θ as the mid-range between λ_k and λ_n

We focus now on choosing a scaling parameter θ to obtain approximate iterates to those of deflated CG (see Algorithm 3). The deflation technique, with S_k as the deflation subspace, is equivalent to standard CG applied to $Ax = b$ with initial guess

$$x_0^{\text{Def}} = x_0 + S_k \Lambda_k^{-1} S_k^\top (b - Ax_0).$$

From (25), the residual of x_0^{Def} is given as

$$b - Ax_0^{\text{Def}} = \bar{S}_k \bar{S}_k^\top r_0.$$

One can see that this initial guess gives a residual which is an orthogonal projection of r_0 onto $\operatorname{span}(\bar{S}_k)$, so that the ℓ-th iterate of CG, x_ℓ^{Def}, starting with x_0^{Def} satisfies

$$\left\|x^* - x_\ell^{\text{Def}}\right\|_A^2 = \min_{q \in \mathbb{P}_\ell(0)} \left(\sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q(\lambda_i)^2 \right).$$

We now provide the main result of this section.

Theorem 6. *Let $\hat{x}_\ell(\theta)$ be the ℓ-th iterate obtained from PCG iterates when solving $Ax = b$ using F_θ starting from an arbitrary initial guess $x_0 \in \mathbb{R}^n$. Let x_ℓ^{Def} be the ℓ-th iterate generated with CG when solving $Ax = b$ starting with $x_0^{Def} = x_0 + S_k \Lambda_k^{-1} S_k^\top (b - Ax_0)$. Then, in exact arithmetic,*

$$\left\|x^* - x_{\ell+1}^{Def}\right\|_A \le \|x^* - \hat{x}_{\ell+1}(\theta)\|_A \le \frac{\alpha(\theta)}{\theta} \left\|x^* - x_\ell^{Def}\right\|_A, \tag{26}$$

with $\alpha(\theta) = \max\left(|\lambda_{k+1} - \theta|, |\theta - \lambda_n|\right)$.

Proof. Let us start by showing the first inequality. From Theorem 1

$$\begin{aligned}
\|x^* - \hat{x}_{\ell+1}(\theta)\|_A^2 &= \sum_{i=1}^{k} \frac{\eta_i^2}{\lambda_i} q_{\ell+1,\theta}^*(\theta)^2 + \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q_{\ell+1,\theta}^*(\lambda_i)^2 \\
&\ge \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q_{\ell+1,\theta}^*(\lambda_i)^2 \\
&\ge \min_{q \in \mathbb{P}_{\ell+1}(0)} \left(\sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q(\lambda_i)^2 \right) = \left\|x^* - x_{\ell+1}^{\text{Def}}\right\|_A^2.
\end{aligned}$$

Now, to prove the second inequality, we consider $p_\ell^{*,\text{Def}}$ the polynomial that minimizes $p \mapsto \|p\,(A)\left(x^* - x_0^{\text{Def}}\right)\|_A^2$ over $\mathbb{P}_\ell(0)$., i.e.,

$$\left\|x^* - x_\ell^{\text{Def}}\right\|_A^2 = \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} p_\ell^{*,\text{Def}}(\lambda_i)^2.$$

Consider $\widetilde{q}_{\ell+1} \in \mathbb{P}_{\ell+1}(0)$ such as for all $\lambda \in \mathbb{R}, \widetilde{q}_{\ell+1}(\lambda) = \left(1 - \dfrac{\lambda}{\theta}\right) p_\ell^{*,\text{Def}}(\lambda)$. Hence,

$$\begin{aligned}
\|x^* - \hat{x}_{\ell+1}(\theta)\|_A^2 &= \sum_{i=1}^{k} \frac{\eta_i^2}{\lambda_i} q_{\ell+1,\theta}^*(\theta)^2 + \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} q_{\ell+1,\theta}^*(\lambda_i)^2 \\
&\le \sum_{i=1}^{k} \frac{\eta_i^2}{\lambda_i} \widetilde{q}_{\ell+1}(\theta)^2 + \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} \widetilde{q}_{\ell+1}(\lambda_i)^2
\end{aligned}$$

$$= \sum_{i=k+1}^{n} \frac{\eta_i^2}{\lambda_i} p_\ell^{\mathrm{Def},*}(\lambda_i) \left(1 - \frac{\lambda_i}{\theta}\right)^2$$

$$\leq \max_{k+1 \leq i \leq n} \left(1 - \frac{\lambda_i}{\theta}\right)^2 \left\|x^* - x_\ell^{\mathrm{Def}}\right\|_A^2 = \frac{\alpha(\theta)}{\theta} \left\|x^* - x_\ell^{\mathrm{Def}}\right\|_A^2 . \qquad \square$$

Choosing $\theta > 0$ such that $\alpha(\theta)/\theta > 1$ in (26) would give a pessimistic upper bound. For a better bound, we select $\theta > 0$ such that $\alpha(\theta)/\theta \leq 1$, which is equivalent to impose $\theta \geq \lambda_{k+1}/2$. The value of θ that minimizes $\alpha(\theta)/\theta$ is $\theta^* = (\lambda_{k+1} + \lambda_n)/2$.

Given that λ_{k+1} is unknown, and λ_n can be predetermined in various applications, e.g., in data assimilation problems $\lambda_n = 1$, a practical approach for selecting θ (the closest to θ^*) is by choosing the average between the λ_k and λ_n, i.e., $\theta_m = (\lambda_k + \lambda_n)/2$, for which we have $\alpha(\theta_m)/\theta_m = (\lambda_k - \lambda_n)/(\lambda_k + \lambda_n) < 1$. Note that the choice $\theta = \lambda_k$ yields in (26) to a worst upper bound compared to θ_m, i.e., $\alpha(\lambda_k)/\lambda_k > \alpha(\theta_m)/\theta_m$.

4.5 Discussion

The analysis in this section raises two key questions. The first is: why use a scaled spectral preconditioner when we know that deflated CG iterations using the deflated subspace S_k, or using an initial guess as defined in (22), produce better results in exact arithmetic (see Theorem 6)? The assumption in this section is that the eigenpairs used to construct the deflated subspace or the initial guess are exact, ensuring that components of the initial residual within the eigenspace of S_k are eliminated. However, when an approximate eigen-spectrum is used, such as the eigen-spectrum of A is applied to solve a system involving a perturbed matrix, $\widetilde{A}$, the initial guess may fail to remove the components of the initial residual within the eigenspace of $\widetilde{A}$. For instance, consider the perturbed matrix $\widetilde{A} = A + E$, A is modified by a small perturbation matrix E. This results in the following expression:

$$b - \widetilde{A}x_0^{\mathrm{Def}} = b - Ax_0^{\mathrm{Def}} + Ex_0^{\mathrm{Def}},$$

where the value of $b - Ax_0^{\mathrm{Def}}$ from (25) becomes: $b - \widetilde{A}x_0^{\mathrm{Def}} = \bar{S}_k \bar{S}_k^\top (b - Ax_0) + Ex_0^{\mathrm{Def}}$.

This illustrates that the perturbation E introduces additional components to the residual, which the initial guess fails to fully eliminate, unlike in the exact case. When the perturbation exists, we show in numerical experiments that using a scaled spectral LMP becomes advantageous over deflated CG.

The second question is: why not combine the initial guess (22) with the scaled spectral LMP using $\theta = 1$. When the initial guess fails to eliminate components of the initial residual within the eigenspace of $\widetilde{A}$, these components influence the convergence of PCG. Their impact on the energy norm of the error can be reduced by appropriately positioning the largest eigenvalues.

5 Numerical Experiments

In this section, we illustrate the performance of the scaled spectral LMP, as defined in (11), within the context of a nonlinear weighted least-squares problem arising in data assimilation, i.e.,

$$\min_{w_0 \in \mathbb{R}^n} f(w_0) = \min_{w_0 \in \mathbb{R}^n} \frac{1}{2} \|w_0 - w_b\|_{B^{-1}}^2 + \frac{1}{2} \sum_{i=1}^{N_t} \|y_i - \mathcal{H}_i(\mathcal{M}_{t_0,t_i}(w_0))\|_{R_i^{-1}}^2 . \tag{27}$$

Here, $w_0 = w(t_0)$, is the state at the initial time t_0, for instance temperature value, $w_b \in \mathbb{R}^n$ is a priori information at time t_0 and $y_i \in \mathbb{R}^{m_i}$ represents the observation vector at time t_i for $i = 1, \dots, N_t$. $\mathcal{M}_{t_0,t_i}(\cdot)$ is a nonlinear physical dynamical model which propagates the state w_0 at time t_0 to the the state w_i at time t_i by solving the partial differential equations. $\mathcal{H}_i(\cdot)$ maps the state vector w_i to a m_i-dimensional vector representing the state vector in the observation space. $B \in \mathbb{R}^{n \times n}$, $R_i \in \mathbb{R}^{m_i \times m_i}$

are symmetric positive definite error covariance matrices corresponding to the a priori and observation model error, respectively.

The TGN method [11] is widely used to solve the nonlinear optimization problem (27). At each iteration j of the TGN method, the linearized least-squares approximation to the nonlinear least-squares problem (27) is solved. This quadratic cost function at the j-th iterate is formulated as

$$Q^{(j)}(s) = \frac{1}{2}\left\|s - (w_b - w_0^{(j)})\right\|^2_{B^{-1}} + \frac{1}{2}\sum_{i=1}^{N_t} \|G_i^{(j)} s_i - d_i^{(j)}\|^2_{R_i^{-1}}, \tag{28}$$

where $s \in \mathbb{R}^n$, $d_i^{(j)} = y_i - \mathcal{G}_i(w_0^{(j)})$ with $\mathcal{G}_i(w_0^{(j)}) = \mathcal{H}_i(\mathcal{M}_{t_0,t_i}(w_0^{(j)}))$ and $G_i^{(j)}$ represents the Jacobian of $\mathcal{G}_i$ at a given iterate $w_0^{(j)}$. The quadratic cost function (28) is minimized with respect to s which is then used to update the current iterate, i.e. $w_0^{(j+1)} = w_0^{(j)} + s^{(j)}$, where $s^{(j)}$ is an approximate solution of the problem (28). This process continues till the convergence criterion is met. For large scale problems with computationally expensive models $\mathcal{M}_{t_0,t_i}(\cdot)$, a limited number of TGN iterations are applied. The solution to the quadratic problem (28) can be found by solving

$$\left(B^{-1} + (G^{(j)})^\top R^{-1} G^{(j)}\right) s = B^{-1}(w_b - w_0^{(j)}) - (G^{(j)})^\top R^{-1} d^{(j)}. \tag{29}$$

where $d^{(j)}$ is a m-dimensional concatenated vector of $d_i^{(j)}$ with $m = \sum_{i=1}^{N_t} m_i$, $G^{(j)} \in \mathbb{R}^{m\times n}$ represents a concatenation of $G_i^{(j)} \in \mathbb{R}^{m_i \times n}$, and $R \in \mathbb{R}^{m\times m}$ is a block diagonal matrix, i.e. $R = \text{diag}(R_1, \ldots, R_N)$. The matrix $B^{-1} + (G^{(j)})^\top R^{-1} G^{(j)}$ is SPD, matrix-vector products with it are accessible only through operators, and n can be large for data assimilation problems. Hence, CG is widely used to solve such systems.

Let us assume that a square root factorization of $B = LL^\top$ is available. The linear system (29) can be then preconditioned by using this *first-level* split preconditioner,

$$\left(I_n + L^\top (G^{(j)})^\top R^{-1} G^{(j)} L\right) x = L^\top \left(B^{-1}(w_b - w_0^{(j)}) - (G^{(j)})^\top R^{-1} d^{(j)}\right). \tag{30}$$

CG at the ℓ-th iteration provides an approximate solution $x_\ell^{(j)}$ which is then used to obtain an approximate solution of the linear system (29), i.e. $s_\ell^{(j)} = L x_\ell^{(j)}$. In operational data assimilation problems, in general $m \ll n$. Consequently, the preconditioned matrix $A^{(j)} = I_n + L^\top (G^{(j)})^\top R^{-1} G^{(j)} L$ has $n - m$ eigenvalues clustered around 1, while the remaining eigenvalues are greater than 1.

Since in the context of TGN, a sequence of closely related linear systems is solved, it is common to update the first-level preconditioner L by using approximate eigenspectrum of the previous linear system [6, 10]. Let us denote $b^{(j)} := L^\top \left(B^{-1}(w_b - w_0^{(j)}) - (G^{(j)})^\top R^{-1} d^{(j)}\right)$. For $j = 1$, CG Algorithm 1 solves the linear system $A^{(1)} x = b^{(1)}$, for the variable x. Using the recurrences of CG, we can easily compute approximate eigenpairs of $A^{(1)}$ (see [22, p.174] for more details). These pairs can then be used to construct a second-level preconditioner, $U_{\theta_1}^{(1)}$, by using the formula (11). Consequently, $(U_{\theta_1}^{(1)})^2$ is an approximation to the inverse of the matrix $A^{(1)}$. Then, assuming that $A^{(2)}$ is close to the matrix $A^{(1)}$, for $j = 2$, CG Algorithm 1 is applied to the preconditioned system, $U_\theta^{(1)} A^{(2)} U_{\theta_1}^{(1)} x = U_{\theta_1}^{(1)} b^{(2)}$. The approximate solution at ℓ-iterate is obtained from the relation $s_\ell^{(2)} = L U_{\theta_1}^{(1)} x_\ell^{(2)}$. At the end of the CG, we can obtain approximate eigenpairs of $U_\theta^{(1)} A^{(2)} U_{\theta_1}^{(1)}$ and use it to construct a preconditioner for the next linear system. At the j-th outer loop of TGN, CG is applied to the preconditioned linear system:

$$(U_{\theta_{j-1}}^{(j-1)} \ldots U_{\theta_1}^{(1)} A^{(j)} U_{\theta_1}^{(1)} \ldots U_{\theta_{j-1}}^{(j-1)})\, x = U_{\theta_{j-1}}^{(j-1)} \ldots U_{\theta_1}^{(1)} b^{(j)}, \tag{31}$$

and the approximate solution to (29) is obtained from $s_\ell^{(j)} = L U_{\theta_{j-1}}^{(j-1)} \ldots U_{\theta_1}^{(1)} x_\ell^{(j)}$.

5.1 Setup

In our numerical experiments, we use the Lorenz-96 [16] model as the physical dynamical system, $\mathcal{M}_{t_0,t_i}(\cdot)$, which is commonly used as a reference model in data assimilation. The observation operator $\mathcal{H}(\cdot)$ is defined as a uniform selection operator, meaning $\mathcal{H}(x)$ extracts a subset of x that is uniformly selected. B is chosen as a discretized diffusion operator with a standard deviation $\sigma_b = 0.8$ [9]. We consider $R_1 = R_2 = \sigma_r^2 I_m$ with $\sigma_r = 0.2$. We choose $n = 1000$ and $N_t = 2$, and we consider two different scenarios, with a different number of observations: (1) *LowObs* with $m_1 = m_2 = 150$ and (2) *HighObs* with $m_1 = m_2 = 300$. For both cases, 2 outer loops are performed within TGN. CG is applied to the first linear system $A^{(1)}x = b^{(1)}$ with 100 iterations. Then, approximate largest eigen-pairs of $A^{(1)}$, (S_k, Λ_k), are computed and selected based on convergence criteria with a tolerance of $\varepsilon = 10^{-3}$ (See [Section 1.3][24] for further details). With this criteria, the number of selected eigen-pairs is 45 in the *LowObs* case and 26 in the *HighObs* case. Using these pairs, the scaled LMP, $U_{\theta_1}^{(1)}$, is applied as a preconditioner for $j = 2$. Matrix-vector products with the preconditioner are carried out via an operator using the selected pairs, meaning the preconditioner matrix is not explicitly constructed.

5.2 Numerical results

In this section, we present numerical results only for the second outer loop ($j = 2$) of the TGN method. We compare the performance of the methodologies of Table 1 in terms of convergence rate and computational cost.

Table 1: Description of methods used in the numerical experiments

Method	Description	Initial guess
BPrec	Algorithm 1 applied to (30)	$x_0 = 0$
sLMP-Base	Algorithm 1 applied to (31), $\theta_1 = 1$	$x_0 = 0$
Init-sLMP-Base	Algorithm 1 applied to (31), $\theta_1 = 1$	$x_0 = U_{\theta_1}^{-1} S_k \Lambda_k^{-1} S_k^\top b^{(2)}$
sLMP-λ_k	Algorithm 1 applied to (31), $\theta_1 = \lambda_k$	$x_0 = 0$
sLMP-θ_r	Algorithm 1 applied to (31), $\theta_1 = \theta_r$	$x_0 = 0$
sLMP-θ_m	Algorithm 1 applied to (31), $\theta_1 = (\lambda_k + 1)/2$	$x_0 = 0$
DefCG	Algorithm 3 applied to (30), $W = S_k$	$x_{-1} = 0$

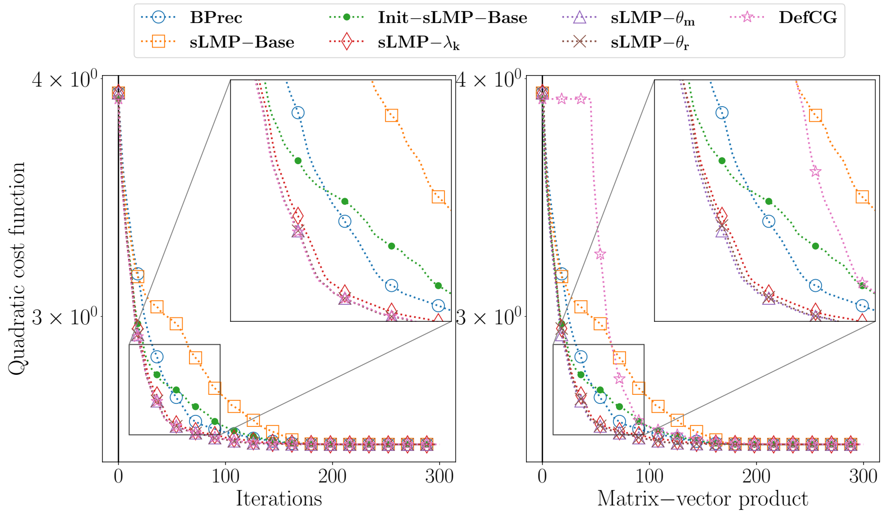

Figure 1: Quadratic cost function values along all CG iterates (left) and with respect to the number of matrix-vector product with the matrix $A^{(1)}$ and $A^{(2)}$ (right).

We can easily see that **sLMP-Base** is not necessarily better than **BPrec** especially in the early iterations. This means that the scaled spectral LMP, clustering the largest k eigenvalues around 1, might reduce the total number iterations to converge, however it does not guarantee better convergence for early iterations. The slow convergence of **sLMP-Base** can be partly explained by the fact that perturbations may cause some eigenvalues to appear near zero, as depicted in Figure 2. When changing the clustering position from 1 to λ_k by using **sLMP-**$\boldsymbol{\lambda_k}$, we can see that the method performs better than **BPrec**. In this case, however the gap between the cluster and the remaining spectrum as defined in Theorem 6, i.e. $\alpha(\theta_1^{(1)})/\theta_1^{(1)}$, can be large. When clustering around θ_r and θ_m is applied with **sLMP-**$\boldsymbol{\theta_r}$ and **sLMP-**$\boldsymbol{\theta_m}$ respectively, the value of $\alpha(\theta_1^{(1)})/\theta_1^{(1)}$ reduces for both cases (see Figure 2). This improves the convergence compared to **sLMP-**$\boldsymbol{\lambda_k}$ as seen from Figure 1.

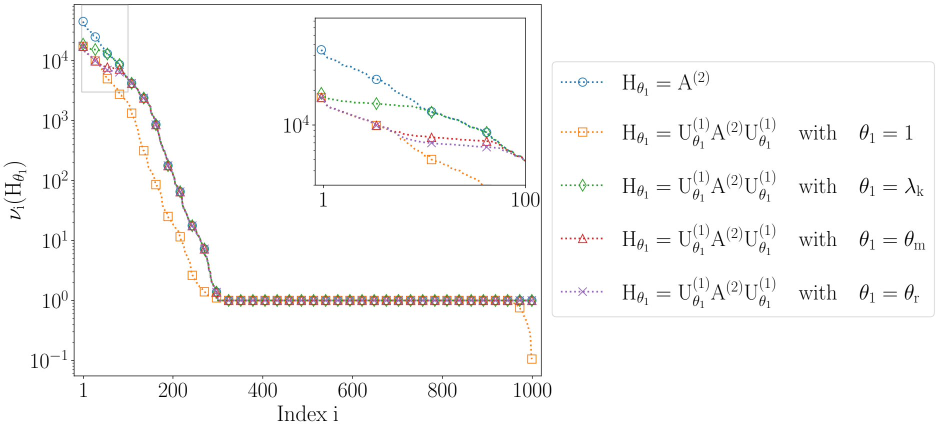

Figure 2: Spectrum of $U_{\theta_1}^{(1)}A^{(2)}U_{\theta_1}^{(1)}$ for different values of θ_1 on a logarithmic scale. LowObs scenario ($k = 45$).

Init-sLMP-Base performs better than **sLMP-Base**, i.e. starting from $x_0 = S_k\Lambda_k^{-1}S_k^\top b^{(2)}$ improves performance compared to starting from $x_0 = 0$. This improvement arises because the initial residual's components in the eigenbasis of $A^{(2)}$ are reduced. In fact, without any perturbation, these components would be completely eliminated. Although, the performance is improved with this initial guess, it can not reach the performance of **DefCG**. This demonstrates that modifying the initial guess enhances convergence; however, the placement of the eigenvalue clustering can have an even more significant impact. This is evident from the fact that the performance of **sLMP-**$\boldsymbol{\theta_m}$ and **sLMP-**$\boldsymbol{\theta_r}$ are very close to that of **DefCG**.

The right panel of Figure 1 shows the values of the quadratic cost function as a function of the number of matrix-vector products performed with $A^{(j)}$ for $j = 1, 2$ across different methods. Although **DefCG** performs better, it is computationally expensive as it requires forming the projected matrix $S_k^\top A^{(2)} S_k$. Among the other techniques, **sLMP-**$\boldsymbol{\theta_r}$ requires one additional matrix-vector product with $A^{(1)}$ to compute θ_r. However, as shown in Figure 1, **sLMP-**$\boldsymbol{\theta_m}$ and **sLMP-**$\boldsymbol{\lambda_k}$ do not require any extra matrix-vector products either $A^{(1)}$ or $A^{(2)}$.

These results indicate that the performance of CG, when used with scaled spectral LMP, can be significantly improved, approaching that of deflated CG, by selecting the position of the eigenvalue clusters based on CG's convergence properties. The cluster position is determined by θ, whose computation incurs no additional cost for **sLMP-**$\boldsymbol{\theta_m}$ and **sLMP-**$\boldsymbol{\lambda_k}$. Conclusions from experiments with *HighObs* are very similar, the obtained results are depicted in Figures 3 and 4 in Appendix B.

6 Conclusion

We have proposed a *scaled* spectral LMP to accelerate the solution of a sequence of SPD systems $A^{(j)}x^{(j)} = b^{(j)}$ for $j \geq 1$. The *scaled* LMP incorporates a low-rank update based on k eigenpairs of the matrix A. We have provided theoretical analysis of the *scaled* spectral LMP when $A^{(j)} = A$. We have shown that the scaled spectral LMP (10) clusters k eigenvalues around the scaling parameter θ, and leaves the rest of the spectrum untouched.

We have focused on the choice of θ to ensure that PCG achieves faster convergence, particularly in the early iterations. In the first approach, we have proposed choosing θ to guarantee a lower energy norm of the error at each iteration of PCG. In the second approach, we have obtained an optimum θ in the sense that it minimizes the energy norm of the error at the first iteration. Our analysis reveals that, with the optimal θ, the components of the first residual is eliminated from the eigenspace of A, which aligns with the core principle of deflated CG. Lastly, we have also explored a scaling parameter that approximates the iterates of deflated CG. We have provided the link between the deflated CG and PCG with the scaled spectral LMP.

We have compared different methods for solving a nonlinear weighted least-squares problem arising in data assimilation. In our numerical experiments, we used approximate eigenpairs to construct the scaled spectral LMP. First, we have demonstrated that selecting θ based on PCG convergence properties significantly accelerates early convergence compared to the conventional choice of $\theta = 1$. Then, we have shown that θ values that reduce the spectral gap between θ and the remaining eigenvalues lead to faster convergence. Additionally, we have compared the scaled spectral LMP with deflated CG, showing that the scaled spectral LMP produces iterates similar to deflated CG, but at a negligible computational cost and memory, unlike deflated CG. These numerical results clearly highlight the importance of selecting the preconditioner not only as an approximation to the inverse of A, but also with consideration of its role within PCG. In particular, we have demonstrated the significance of the placement of clustered eigenvalues, an often overlooked factor in the literature, on the early convergence of PCG.

As the next step, we will provide a detailed theoretical perturbation analysis in a forthcoming paper. Additionally, we aim to validate the proposed preconditioner in an operational weather prediction system.

A Deflated CG with S_k

The deflation technique outlined in Algorithm 3 is defined for any deflation subspace W, see [23] for more details. The main idea is to speed-up the CG starting from an initial point such that the initial residual does not have components in the deflation subspace W and to update the search directions such that $W^\top A p_j = 0$. A widely used approach is to choose W as the eigenvectors corresponding to the eigenvalues that slows down the CG convergence.

Algorithm 3 Deflated-CG

```
Choose $k$ linearly independent vectors $w_1, w_2, \ldots, w_k$.
Define $W = [w_1, w_2, \ldots, w_k]$, and choose $x_{-1}$.
Set $x_0^{\mathrm{Def}} = x_{-1} + W(W^\top AW)^{-1}W^\top r_{-1}$, where $r_{-1} = b - Ax_{-1}$.        $W^\top r_0 = 0$
Set $p_0 = r_0 - W\left(W^\top AW\right)^{-1} W^\top Ar_0$.        $W^\top Ap_0 = 0$
for $j = 1, 2, \ldots$ do
    $\alpha_{j-1} = r_{j-1}^\top r_{j-1}/(p_{j-1}^\top Ap_{j-1})$
    $x_j^{\mathrm{Def}} = x_{j-1}^{\mathrm{Def}} + \alpha_{j-1}p_{j-1}$
    $r_j = r_{j-1} - \alpha_{j-1}Ap_{j-1}$        $W^\top r_j = 0$
    $\beta_{j-1} = r_j^\top r_j/(r_{j-1}^\top r_{j-1})$
    $p_j = \beta_{j-1}p_{j-1} + r_j - W\left(W^\top AW\right)^{-1} W^\top Ar_j$        $W^\top Ap_j = 0$
end for
```

If we choose $W = S_k$, and using the fact that $S_k^\top AS_k = \Lambda_k$ and $AS_k = S_k\Lambda_k$, we can achieve the following simplifications:

- $x_0^{\mathrm{Def}} = x_{-1} + S_k\Lambda_k^{-1}S_k^\top r_{-1}$,
- $p_0 = r_0 - S_kS_k^\top r_0$.
- $p_j = \beta_{j-1}p_{j-1} + r_j - S_kS_k^\top r_j$.

Lemma 2. *The residual r_j and the direction p_j are orthogonal to span(S_k).*

Proof. We proceed by induction. For $j = 0$, $r_0 = r_{-1} - S_k S_k^\top r_{-1}$, from which it follows that $S_k^\top r_0 = 0$. As a consequence, $S_k^\top p_0 = 0$. Assume that r_j and p_j are orthogonal to $\text{span}(S_k)$ for j. We have $r_{j+1} = r_j - \alpha_j A p_j$. From [23, Proposition 3.3], replacing W by S_k, we have $S_k^T A p_j = 0$. Since $p_j, r_j \perp \text{span}(S_k)$ by assumption, it follows that $r_{j+1} \perp \text{span}(S_k)$. For $p_{j+1} = \beta_j p_j + r_{j+1} - S_k S_k^\top r_{j+1} = \beta_j p_j + r_{j+1}$, we get $p_{j+1} \perp \text{Span}(S_k)$ since $S_k^\top r_{j+1} = 0$ as shown and $p_j \perp \text{Span}(S_k)$ by assumption. □

From Lemma 2, it follows that $p_j = \beta_{j-1} p_{j-1} + r_j - S_k S_k^\top r_j = \beta_{j-1} p_{j-1} + r_j$. With these simplifications, it is clear that in exact arithmetic, deflated CG, when used with the deflated subspace consisting of a set of eigenvectors of A, generates iterates equivalent to those generated by using the initial guess x_0^{Def} in standard CG.

B Results for the HighObs scenario

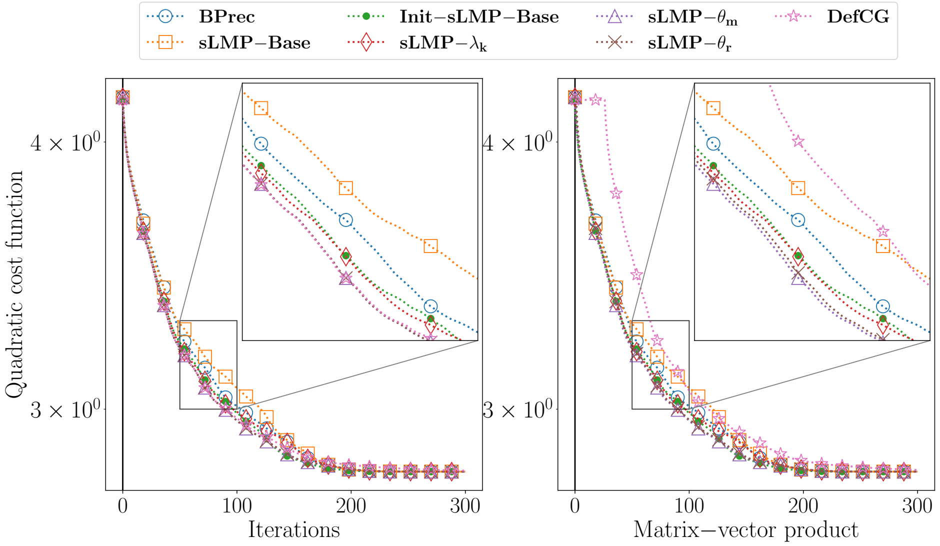

Figure 3: Quadratic cost function values along all CG iterates and with respect to the number of matrix-vector product for the HighObs scenario.

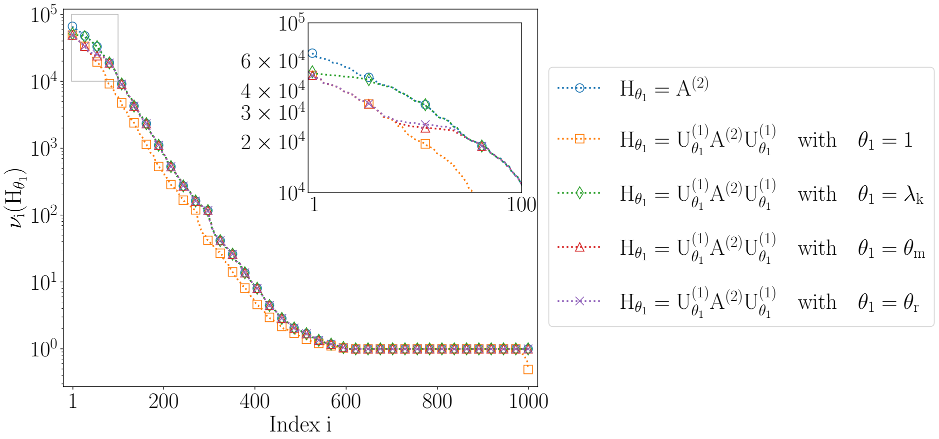

Figure 4: Spectrum of $U_{\theta_1}^{(1)} A^{(2)} U_{\theta_1}^{(1)}$ with different θ_1 for the HighObs scenario $(k = 26)$.

References

[1] M Benzi. Preconditioning techniques for large linear systems: A survey. Journal of Computational Physics, 182(2):418–477, 2002.

[2] Erin Carson, Jörg Liesen, and Zdeněk Strakoš. Towards understanding cg and gmres through examples. Linear Algebra and its Applications, 2024.

[3] Erin Carson and Zdeněk Strakoš. On the cost of iterative computations. Philosophical Transactions of the Royal Society A, 378(2166):20190050, 2020.

[4] Roger Daley. Atmospheric data analysis. Cambridge University Press, 1991.

[5] Van der Sluis and Van der Vorst. The rate of convergence of conjugate gradients. Numerische Mathematik, 48(5):543–560, 1986.

[6] M. Fisher, J. Nocedal, Y. Trémolet, and Stephen J. Wright. Data assimilation in weather forecasting: a case study in pde-constrained optimization. Optimization and Engineering, 10(3):409–426, 2009.

[7] L. Giraud and S. Gratton. On the sensitivity of some spectral preconditioners. SIAM Journal on Matrix Analysis and Applications, 27(4):1089–1105, 2006.

[8] Gene H. Golub and Charles F. Van Loan. Matrix Computations. The Johns Hopkins University Press, Baltimore, 4th edition, 2013.

[9] Olivier Goux, Selime Gürol, Anthony T Weaver, Youssef Diouane, and Oliver Guillet. Impact of correlated observation errors on the conditioning of variational data assimilation problems. Numerical Linear Algebra with Applications, 31(1):e2529, 2024.

[10] S. Gratton, A. Sartenaer, and J. Tshimanga. On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. SIAM Journal on Optimization, 21(3):912–935, 2011.

[11] Serge Gratton, Amos S Lawless, and Nancy K Nichols. Approximate gauss–newton methods for nonlinear least squares problems. SIAM Journal on Optimization, 18(1):106–132, 2007.

[12] A Greenbaum. Comparison of splittings used with the conjugate gradient algorithm. Numerische Mathematik, 33(2):181–193, june 1979.

[13] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards, 49:409–435, 1952.

[14] K. Kahl and H. Rittich. The deflated conjugate gradient method: Convergence, perturbation and accuracy. Linear Algebra and its Applications, 515:111–129, 2017.

[15] Eugenia Kalnay. Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press, 2002.

[16] Edward N Lorenz. Predictability: A problem partly solved. In Proc. Seminar on predictability, volume 1. Reading, 1996.

[17] José Luis. Morales and Jorge. Nocedal. Automatic preconditioning by limited memory quasi-newton updating. SIAM Journal on Optimization, 10(4):1079–1096, 2000.

[18] Reinhard Nabben and Cornelis Vuik. A comparison of deflation and the balancing preconditioner. SIAM Journal on Scientific Computing, 27(5):1742–1759, 2006.

[19] Stephen G Nash and Jorge Nocedal. A numerical study of the limited memory bfgs method and the truncated-newton method for large scale optimization. SIAM Journal on Optimization, 1(3):358–372, 1991.

[20] Jorge Nocedal and Stephen J. Wright. Numerical Optimization. Springer, New York, NY, USA, 2nd edition, 2006.

[21] John W. Pearson and Jennifer Pestana. Preconditioners for Krylov subspace methods: An overview. GAMM-Mitteilungen, 43(4):e202000015, 2020.

[22] Y Saad. Iterative methods for sparse linear systems. PWS Publishing Company, Boston, USA, 1996.

[23] Y. Saad, M. Yeung, J. Erhel, and F Guyomarc'h. A deflated version of the conjugate gradient algorithm. SIAM Journal on Scientific Computing, 21(5):1909–1926, january 2000.

[24] J. Tshimanga. On a class of limited memory preconditioners for large-scale nonlinear least-squares problems (with application to variational ocean data assimilation). PhD thesis, Department of Mathematics, University of Namur, Namur, Belgium, 2007.

[25] Andrew J Wathen. Preconditioning. Acta Numerica, 24:329–376, 2015.