



HAL
open science

Recycling Krylov subspace strategies for sequences of sampled stochastic elliptic equations

Nicolas Venkovic, Paul Mycek, Luc Giraud, Olivier Le Maitre

► **To cite this version:**

Nicolas Venkovic, Paul Mycek, Luc Giraud, Olivier Le Maitre. Recycling Krylov subspace strategies for sequences of sampled stochastic elliptic equations. [Research Report] RR-9425, Inria Bordeaux - Sud Ouest. 2021. hal-03366966

HAL Id: hal-03366966

<https://hal.science/hal-03366966v1>

Submitted on 4 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Recycling Krylov subspace strategies for sequences of sampled stochastic elliptic equations

Nicolas Venkovic, Paul Mycek, Luc Giraud, Olivier P. Le Maître

**RESEARCH
REPORT**

N° 9425

October 2021

Project-Teams HiePACS



Recycling Krylov subspace strategies for sequences of sampled stochastic elliptic equations

Nicolas Venkovic*, Paul Mycek*, Luc Giraud†, Olivier P. Le
Maître‡

Project-Teams HiePACS

Research Report n° 9425 — October 2021 — 24 pages

This work is a revised and extended version of a short paper accepted for the Student Paper Competition of CMCIM 2020.

This work was funded by Cerfacs.

* Cerfacs, Toulouse, France (venkovic@gmail.com, mycek@cerfacs.fr).

† HiePACS, Inria Bordeaux, Sud-Ouest, Talence, France (luc.giraud@inria.fr).

‡ CNRS, CMAP, École Polytechnique, Palaiseau, France (olivier.le-maitre@polytechnique.edu).

**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200 avenue de la Vieille Tour
33405 Talence Cedex

Abstract: We are interested in the quantification of uncertainties in discretized elliptic partial differential equations with a random coefficient field. In sampling-based approaches, this relies on solving large numbers of symmetric positive definite (SPD) linear systems with different matrices. In particular, we consider the case in which these operators are sampled by Markov chain Monte Carlo, which leads to sequences of correlated matrices. We investigate recycling Krylov subspace strategies for the iterative solution of sequences of linear systems formed with such matrices. The linear systems are solved using initialized conjugate gradient (Init-CG) methods, where approximate eigenvectors of the previously sampled operator are used to set an initial guess, and deflated conjugate gradient (Def-CG) methods, where the Krylov subspace is augmented with these vectors. The following aspects of eigenvector approximation, and their effect on deflation and initialization, are investigated in this context: (i) projection technique, and (ii) refreshing strategy of the eigen-search space. Our numerical experiments show that, when not using a preconditioner, these aspects only impact convergence behaviors of Def-CG at the early stages of the sampling sequence. Second, unlike sequences with multiple right-hand sides and a constant operator, our experiments with multiple matrices show that, even for highly correlated matrices, Init-CG does not reproduce the convergence behavior of Def-CG. Finally, the limits of deflation used as a means to compensate for the inefficiency of block-Jacobi (bJ) preconditioners are investigated. For small systems, using a bJ preconditioner while deflating with at least as many approximate eigenvectors as the number of bJ blocks achieves similar convergence behaviors to PCG with a constant algebraic multigrid (AMG) preconditioner. For larger systems, although the effect of deflation is improved when using the right refreshing strategy of the eigen-search space, the combination of deflation with bJ preconditioners does not scale as well as using PCG with a constant AMG preconditioner based on the median realization of the coefficient field.

Key-words: Deflation, recycling Krylov subspaces, uncertainty quantification, stochastic PDEs.

Stratégies de recyclage de sous-espaces de Krylov pour des suites d'équations elliptiques stochastiques échantillonnées

Résumé : Nous nous intéressons à la quantification des incertitudes dans les équations aux dérivées partielles elliptiques discrétisées avec un champ de coefficients aléatoires. Dans les approches basées sur l'échantillonnage, celle-ci repose sur la résolution d'un grand nombre de systèmes linéaires symétriques définis positifs (SPD) avec des matrices différentes. En particulier, nous considérons le cas où ces opérateurs sont échantillonnés par une méthode de Monte Carlo par chaînes de Markov, ce qui produit des suites de matrices corrélées. Nous étudions des stratégies de recyclage de sous-espaces de Krylov pour la résolution itérative de suites de systèmes linéaires provenant de telles matrices. Les systèmes linéaires sont résolus en utilisant des méthodes de gradient conjugué initialisé (Init-CG), où les vecteurs propres approximatifs de l'opérateur précédemment échantillonné sont utilisés pour définir une estimation initiale, et des méthodes de gradient conjugué déflaté (Def-CG), où le sous-espace de Krylov est augmenté de ces vecteurs propres. Les aspects suivants de l'approximation des vecteurs propres, et leur effet sur la déflation et l'initialisation, sont étudiés dans ce contexte : (i) la technique de projection, et (ii) la stratégie de rafraîchissement de l'espace de recherche des vecteurs propres. Nos expériences numériques montrent que, lorsqu'on n'utilise pas de préconditionneur, ces aspects n'ont un impact sur les comportements de convergence de Def-CG qu'au début de la suite d'échantillonnage. Deuxièmement, contrairement aux séquences avec des second-membres multiples et un opérateur constant, nos expériences avec des matrices multiples montrent que, même pour des matrices fortement corrélées, Init-CG ne reproduit pas le comportement de convergence de Def-CG. Enfin, nous étudions les limites de la déflation utilisée comme moyen de compenser l'inefficacité des préconditionneurs bloc-Jacobi (bJ). Pour les petits systèmes, l'utilisation d'un préconditionneur bJ tout en déflatant avec au moins autant de vecteurs propres approximatifs que le nombre de blocs bJ permet d'obtenir des comportements de convergence similaires à ceux de PCG avec un préconditionneur multigrille algébrique (AMG) constant. Pour les systèmes plus grands, bien que l'effet de la déflation soit amélioré en utilisant la bonne stratégie de rafraîchissement de l'espace de recherche des vecteurs propres, la combinaison de la déflation avec les préconditionneurs bJ ne s'adapte pas aussi bien que l'utilisation du PCG avec un préconditionneur AMG constant basé sur la réalisation médiane du champ de coefficients.

Mots-clés : Déflation, recyclage de sous-espaces de Krylov, quantification d'incertitudes, EDP stochastique.

Contents

1	Introduction	5
2	Discretization and sampling of stochastic PDEs	5
3	Deflated Krylov subspace strategies	7
3.1	Deflation	8
3.2	Recycling strategy	9
3.3	Projection techniques for the approximation of eigenvectors	9
3.4	Refreshing the eigen-search space	10
3.4.1	Thick-refresh	10
3.4.2	Locally optimal thick-refresh	11
3.5	Practical considerations	11
3.5.1	Assembly of the reduced eigenvalue problem	12
3.5.2	Refreshing the eigen-search space	13
3.6	Numerical results	13
4	Deflated Krylov subspace strategies with fixed preconditioners	16
4.1	Deflation	16
4.2	Projection techniques for the approximation of eigenvectors	17
4.3	Practical considerations	18
4.4	Numerical results	19
5	Conclusion	22

1 Introduction

We are interested in solving sequences of symmetric positive definite (SPD) linear systems that arise when sampling discretized elliptic PDEs with uncertain coefficients. For this purpose, we use INIT-CG [2] and DEF-(P)CG [19] algorithms, with deflation subspaces spanned by approximate eigenvectors of the sampled operator. Eigenvector approximations are carried out by projection techniques with respect to Krylov subspaces of the previous operator in the sampling sequence augmented with previous eigenvector approximations. Hence, precise eigenvector approximations serve the purpose of deflation if the next system in the sequence has a matrix with similar eigenvectors. To improve the effect of this recycling on deflation, we consider sequences in which the latent random variables of the coefficient field are sampled by Markov chain Monte Carlo (MCMC), as it introduces a correlation between subsequently sampled operators.

Different technical aspects arise when solving such MCMC-sampled sequences of linear systems with recycling Krylov subspace strategies. First, eigenvector approximations can be performed using different projection techniques. While some authors favor the use of harmonic Rayleigh-Ritz (HR) projections for sequences of linear systems with a constant operator [19], and multiple operators [12], other recycling strategies used for a constant operator are based on Rayleigh-Ritz (RR) projections [20]. Second, the search space in which the approximate eigenvectors are sought, i.e., the eigen-search space, may only make use of a fraction of the matrix-vector products generated by the solver [12, 19], or be *refreshed* periodically throughout the linear solve so as to make use of all these products in hope of improving the eigenvector approximation. This is for instance the case in eigCG [20]. Note that we use the term *refresh* rather than restart to clarify that, in such procedures, the linear solver is not restarted. Third, while DEF-(P)CG [19] forces the iterated residual of the linear system to remain orthogonal to a deflation subspace, this step, which entails an additional computational cost at every iteration, is sometimes successfully bypassed by using INIT-CG methods for sequences of linear systems with multiple right-hand sides and a fixed operator [3, 20]. However, it is not obvious whether a similar approach can be adopted for MCMC-sampled sequences with multiple operators.

In this work, we address the aforementioned technical aspects. The problem at hand is stated in Section 2. Section 3 presents how approximate eigenvectors can be computed to deflate sampled sequences of linear systems and initialize the linear solves. Different refreshing strategies of the eigen-search space are presented, and their effect on deflation and initialization is investigated through numerical experiments without preconditioner. The combined action of deflation and preconditioning is discussed in Section 4, where the use of deflation as a means to compensate for the inefficiency of constant bJ preconditioners is compared to PCG solves using an algebraic multigrid (AMG) solver as a constant preconditioner. Finally, conclusions are drawn based on the results of our numerical experiments.

2 Discretization and sampling of stochastic PDEs

Let (Θ, Σ, μ) be a probability triplet and $\Omega \subset \mathbb{R}^d$ be a bounded open domain for $d \in \{1, 2, 3\}$. We want to find $u : \bar{\Omega} \times \Theta \rightarrow \mathbb{R}$, such that

$$\nabla \cdot (a(x; \theta) \nabla u(x; \theta)) = -f(x) \quad \forall x \in \Omega, \quad (1)$$

and deterministic boundary conditions for all $x \in \partial\Omega$. For almost all $\theta \in \Theta$, realizations $a(\cdot; \theta)$ of the random coefficient field are strictly positive and bounded above almost everywhere in the domain Ω . In the present work, we restrict ourselves to the case of random coefficients with log-normal distributions; denoting $g(x; \theta) = \log a(x; \theta)$ the underlying Gaussian field, the stochastic

discretization of a relies on the Karhunen-Loève [5] expansion of g ,

$$g(x; \theta) = \mathbb{E}[g(x; \theta)] + \sum_{l=0}^{\infty} \sqrt{\gamma_l} g_l(x) \xi_l(\theta), \quad (2)$$

where (γ_l, g_l) are the eigen-pairs of the covariance function of the Gaussian process. Ordering the normalized eigen-pairs, $\gamma_1 \geq \gamma_2 \geq \dots \geq 0$, the KL expansion can be truncated to retain the first $n_s > 1$ dominant modes. In addition, the random variables ξ_l are independent and identically distributed standard Gaussian variables: $\boldsymbol{\xi} := (\xi_1 \dots \xi_{n_s})^T \sim \mathcal{N}(0, \mathbf{I}_{n_s})$. Therefore, the field a can be sampled by sampling $\boldsymbol{\xi}$. In the present work, we consider a with unit median value, $\mathbb{E}[g(x; \theta)] = 0$, and two cases. Case 1 corresponds to a 1D stationary coefficient field with exponential covariance function for g , given by $\mathbb{E}[g(x)g(y)] = \sigma^2 \exp(-|x - y|/L)$, in which $\sigma^2 = 0.5$ and $L = 0.05$. Case 2 corresponds to a 2D stationary coefficient field with squared exponential covariance function for g given by $\mathbb{E}[g(x)g(y)] = \sigma^2 \exp(-\|x - y\|^2/L^2)$ in which $\sigma^2 = 1$ and $L = 0.1$. For Case 1 we used $n_s = 1,802$ while for Case 2 $n_s = 176$. Figures 1 and 2 report few realizations of a for Case 1 and 2 respectively.

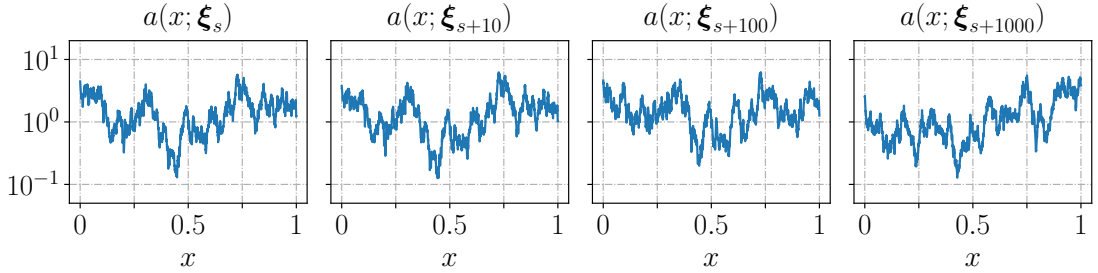


Figure 1: Realizations of $a(x) := \exp(g(x))$ for Case 1.

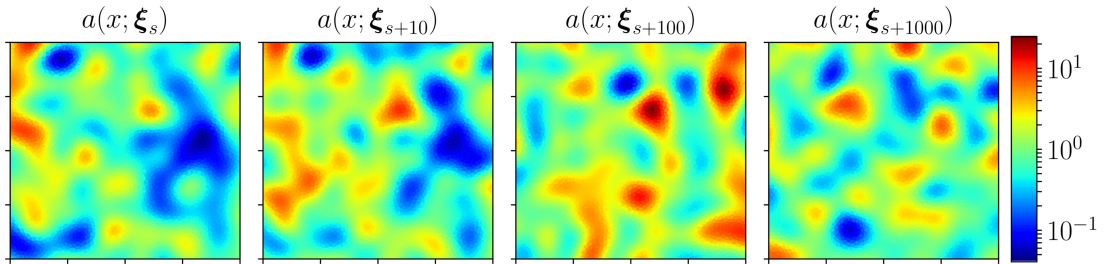


Figure 2: Realizations of $a(x)$ for Case 2.

Upon subsequent spatial discretization, the approximation of u reduces to an n -by- n SPD linear system $\mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi})$. In this work, we present and use iterative methods to solve linear systems $\mathbf{A}_s \mathbf{x}_s = \mathbf{b}_s$ in which $\mathbf{A}_s := \mathbf{A}(\boldsymbol{\xi}_s)$, and so on, where $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots$ are samples of $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I}_{n_s})$. Because these methods are best behaved when \mathbf{A}_{s-1} and \mathbf{A}_s are similar, we consider the case where $\boldsymbol{\xi}$ is sampled by MCMC using a Gaussian proposal distribution with covariance $2.38^2 \mathbf{I}_{n_s} / n_s$. The similarities between the successive samples of the coefficient field $a(x; \boldsymbol{\xi}_s)$ can be appreciated from Figures 1 and 2. As a result, $\mathbf{A}_1, \mathbf{A}_2, \dots$ are correlated matrices, and a reduction of covariance in $\delta \mathbf{A}_s := \mathbf{A}_s - \mathbf{A}_{s-1}$ is induced, which does not occur

when sampling by standard Monte Carlo. While $\delta\mathbf{A}_s$ is generally not low rank, the improved similarity between \mathbf{A}_{s-1} and \mathbf{A}_s still provides better working conditions for the iterative methods we design to solve the linear systems $\mathbf{A}_s\mathbf{x}_s = \mathbf{b}_s$. The coefficient field being lognormal, we refer to $\hat{\mathbf{A}} := \mathbf{A}(\mathbf{0})$ as the median operator.

3 Deflated Krylov subspace strategies

In case of a spatial discretization using finite elements, \mathbf{A}_s is sparse, and its fast application is leveraged when searching for an approximation $\mathbf{x}_s^{(j)}$ of \mathbf{x}_s in the affine subspace $\mathbf{x}_s^{(0)} + \mathcal{K}_j(\mathbf{A}_s, \mathbf{r}_s^{(0)})$, where

$$\mathcal{K}_j(\mathbf{A}_s, \mathbf{r}_s^{(0)}) := \text{Span}\{\mathbf{r}_s^{(0)}, \mathbf{A}_s\mathbf{r}_s^{(0)}, \dots, \mathbf{A}_s^{j-1}\mathbf{r}_s^{(0)}\} \quad (3)$$

is the j -th Krylov subspace of \mathbf{A}_s generated by $\mathbf{r}_s^{(0)} := \mathbf{b}_s - \mathbf{A}_s\mathbf{x}_s^{(0)}$ for a given initial guess $\mathbf{x}_s^{(0)}$. Then, the orthogonal projection obtained by letting $\mathbf{r}_s^{(j)} \perp \mathcal{K}_j(\mathbf{A}_s, \mathbf{r}_s^{(0)})$, leads to an optimal iterate in the sense that $\|\mathbf{x}_s^{(j)} - \mathbf{x}_s\|_{\mathbf{A}_s}^2 := (\mathbf{x}_s^{(j)} - \mathbf{x}_s)^T \mathbf{A}_s (\mathbf{x}_s^{(j)} - \mathbf{x}_s)$ is minimized by the approximation over the search space. In this work, \mathbf{A}_s is SPD, and the sequence $\{\mathbf{x}_s^{(j)}\}_{j=1}^m$ of these optimal iterates is obtained by the CG algorithm, i.e., $\text{CG}(\mathbf{A}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)})$.

While the rate of convergence of $\text{CG}(\mathbf{A}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)})$ is governed by the distribution of the eigenvalues of \mathbf{A}_s , it admits the following bound:

$$\|\mathbf{x}_s^{(m)} - \mathbf{x}_s\|_{\mathbf{A}_s} \leq 2\|\mathbf{x}_s^{(0)} - \mathbf{x}_s\|_{\mathbf{A}_s} \left(\frac{\sqrt{\kappa(\mathbf{A}_s)} - 1}{\sqrt{\kappa(\mathbf{A}_s)} + 1} \right)^m \quad (4)$$

where $\kappa(\mathbf{A}_s) = \lambda_n(\mathbf{A}_s)/\lambda_1(\mathbf{A}_s)$ is the condition number of \mathbf{A}_s with eigenvalues $0 < \lambda_1(\mathbf{A}_s) \leq \dots \leq \lambda_n(\mathbf{A}_s)$. While the tightness of this bound also depends on the distribution of the eigenvalues, Eq. (4) provides a way to understand the relation between the convergence behavior of CG and the eigenvalues at both ends of the spectrum $\text{Sp}(\mathbf{A}_s)$. In particular, it is understood that an increase of $\lambda_1(\mathbf{A}_t)$ (or a decrease of $\lambda_n(\mathbf{A}_s)$) results in a decrease of the bound on the rate of convergence. Also, if the eigenvalue $\lambda_1(\mathbf{A}_s)$ (resp. $\lambda_n(\mathbf{A}_s)$) is moved towards the center of the spectrum past its closest neighbor, then the upper bound of Eq. (4) is scaled by $\lambda_1(\mathbf{A}_s)/\lambda_2(\mathbf{A}_s)$ (resp. $\lambda_{n-1}(\mathbf{A}_s)/\lambda_n(\mathbf{A}_s)$). Hence, well separated eigenvalues located at either end of the spectrum of \mathbf{A}_s can significantly hinder convergence and, canceling, or at least, attenuating their effect may result in a substantially faster convergence [4].

If k eigenvectors of \mathbf{A}_s are known, they can be used to force the CG procedure to work with subspaces which are convenient enough to enable a convergence to \mathbf{x}_s at a rate bounded as in Eq. (4), but with a potentially smaller condition number. Let these k eigenvectors be stored by columns in $\mathbf{U}_s \in \mathbb{R}^{n \times k}$, and paired with the eigenvalues in $\Lambda_s \subset \text{Sp}(\mathbf{A}_s)$. We want $\{\mathbf{x}_s^{(j)}\}_{j=1}^m$ to be such that $\mathbf{r}_s^{(j)} \perp \mathcal{R}(\mathbf{U}_s)$, where $\mathcal{R}(\mathbf{U}_s)$ is the range of \mathbf{U}_s , and $\mathbf{r}_s^{(j)} := \mathbf{b}_s - \mathbf{A}_s\mathbf{x}_s^{(j)}$ for all $0 \leq j \leq m$. Since $\mathcal{R}(\mathbf{U}_s)^\perp$ is invariant under the action of \mathbf{A}_s , this is achieved by setting $\mathbf{x}_s^{(0)}$ such that $\mathbf{r}_s^{(0)} \perp \mathcal{R}(\mathbf{U}_s)$ in $\text{CG}(\mathbf{A}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)})$, i.e., by letting $\mathbf{x}_s^{(0)} := \mathbf{U}_s(\mathbf{U}_s^T \mathbf{A}_s \mathbf{U}_s)^{-1} \mathbf{U}_s^T \mathbf{b}_s$. Then, the sequence $\{\mathbf{x}_s^{(j)}\}_{j=1}^m$ converges to \mathbf{x}_s at a rate bounded by Eq. (4), but where the condition number is now given by $\kappa = \max\{\text{Sp}(\mathbf{A}_s) \setminus \Lambda_s\} / \min\{\text{Sp}(\mathbf{A}_s) \setminus \Lambda_s\}$. Essentially, the effect of the eigenvalues in Λ_s is canceled at the expense of an initial \mathbf{A}_s -orthogonal projection of the solution \mathbf{x}_s onto $\mathcal{R}(\mathbf{U}_s)$, which does not require much computation as long as $k \ll n$. The resulting procedure, referred to as INIT-CG, was introduced in [2] motivated by [13, 15, 21], and can lead

to significantly improved convergence behaviors if a small number of well separated eigenpairs are properly selected at the end(s) of the spectrum.

In practice, \mathbf{U}_s is not known. Instead, it is possible to construct $\mathbf{W}_s \in \mathbb{R}^{n \times k}$, where $\mathcal{R}(\mathbf{W}_s)$ somehow approximates the subspace associated with the eigenvalues of interest. If this approximation is good enough, using \mathbf{W}_s in place of \mathbf{U}_s in INIT-CG may yield improved convergence behaviors similar to when using \mathbf{U}_s , depending on the target accuracy [3]. As the quality of this approximation deteriorates, the \mathbf{A}_s -invariance of $\mathcal{R}(\mathbf{W}_s)$ is no longer guaranteed, causing the residuals $\mathbf{r}_s^{(j)}$ generated by INIT-CG to lose their orthogonality with respect to $\mathcal{R}(\mathbf{W}_s)$, and the original behavior of CG is recovered. Deflation can be used as a means to circumvent the effects of this loss of invariance, by forcing the residuals to remain orthogonal to the subspace.

3.1 Deflation

Deflation consists of splitting the approximation space into two complementary subspaces with a projector such that the projected linear system, referred to as the deflated system, is more amenable to iterative solving than the original system. Given $\mathbf{W}_s \in \mathbb{R}^{n \times k}$ whose range approximates the subspace associated with some of the eigenvalues hindering the convergence of $\text{CG}(\mathbf{A}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)})$, we consider the \mathbf{A}_s -orthogonal projector onto $\mathcal{R}(\mathbf{W}_s)^{\perp \mathbf{A}_s}$ given by

$$\mathbf{\Pi}_s := \mathbf{I}_n - \mathbf{W}_s (\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s)^{-1} \mathbf{W}_s^T \mathbf{A}_s \quad (5)$$

where $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s$ is invertible as \mathbf{W}_s is full column rank. The solution \mathbf{x}_s is then uniquely decomposed into

$$\mathbf{x}_s = (\mathbf{I}_n - \mathbf{\Pi}_s) \mathbf{x}_s + \mathbf{\Pi}_s \mathbf{x}_s, \quad (6)$$

whose parts respectively lie in $\mathcal{R}(\mathbf{W}_s)$ and $\mathcal{R}(\mathbf{W}_s)^{\perp \mathbf{A}_s}$. It follows from the definition of $\mathbf{\Pi}_s$ in Eq. (5) that the first projection can be recast in

$$(\mathbf{I}_n - \mathbf{\Pi}_s) \mathbf{x}_s = \mathbf{W}_s \boldsymbol{\mu}_s \quad (7)$$

where $\boldsymbol{\mu}_s$ is such that $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s \boldsymbol{\mu}_s = \mathbf{W}_s^T \mathbf{b}_s$, and is not difficult to solve for as long as $k \ll n$. From Eq. (5), we have $\mathbf{A}_s \mathbf{\Pi}_s = \mathbf{\Pi}_s^T \mathbf{A}_s$ so that $\mathbf{A}_s \mathbf{\Pi}_s \mathbf{x}_s = \mathbf{\Pi}_s^T \mathbf{A}_s \mathbf{x}_s = \mathbf{\Pi}_s^T \mathbf{b}_s$. Because $\mathbf{\Pi}_s^T$ is also a projector, it is idempotent, and the second part $\mathbf{\Pi}_s \mathbf{x}_s$ of Eq. (6) can be obtained from solutions $\hat{\mathbf{x}}_s$ of the so called *deflated system* given by

$$\mathbf{\Pi}_s^T \mathbf{A}_s \hat{\mathbf{x}}_s = \mathbf{\Pi}_s^T \mathbf{b}_s. \quad (8)$$

Although $\mathbf{\Pi}_s^T \mathbf{A}_s$ is semi-positive definite with null space $\mathcal{R}(\mathbf{W}_s)$, Eq. (8) is still consistent in that $\mathbf{\Pi}_s^T \mathbf{b}_s \in \mathcal{R}(\mathbf{\Pi}_s^T \mathbf{A}_s)$. Actually, Eq. (8) admits infinitely many solutions, all of which are such that $\mathbf{\Pi}_s \hat{\mathbf{x}}_s = \mathbf{\Pi}_s \mathbf{x}_s$. Because $\mathcal{R}(\mathbf{W}_s)^{\perp}$ is invariant under the action of the symmetric operator $\mathbf{\Pi}_s^T \mathbf{A}_s$, setting $\hat{\mathbf{x}}_s^{(0)}$ such that $\hat{\mathbf{r}}_s^{(0)} \perp \mathcal{R}(\mathbf{W}_s)$ implies that the sequence $\{\hat{\mathbf{x}}_s^{(j)}\}_{j=1}^m$ generated by $\text{CG}(\mathbf{\Pi}_s^T \mathbf{A}_s, \mathbf{\Pi}_s^T \mathbf{b}_s, \hat{\mathbf{x}}_s^{(0)})$ satisfies $\hat{\mathbf{r}}_s^{(j)} \perp \mathcal{R}(\mathbf{W}_s)$ for all $0 \leq j \leq m$. Moreover, the post-processed sequence $\mathbf{x}_s^{(j)} := \mathbf{W}_s \boldsymbol{\mu}_s + \mathbf{\Pi}_s \hat{\mathbf{x}}_s^{(j)}$ converges to \mathbf{x}_s at a rate bounded as in Eq. (4), but where the *active* condition number¹ κ is evaluated only over the strictly positive eigenvalues of $\mathbf{\Pi}_s^T \mathbf{A}_s$. Then, deflation guarantees a reduction of the active condition number of the system, i.e., $\kappa(\mathbf{\Pi}_s^T \mathbf{A}_s) < \kappa(\mathbf{A}_s)$. The resulting sequence $\{\mathbf{x}_s^{(j)}\}_{j=1}^m$ can be obtained in different ways, one of

¹The active condition number κ is taken as the ratio of the maximum and minimum strictly positive eigenvalues of the operator.

which, presented in [19], is here referred to as DEF-CG($\mathbf{A}_s, \mathbf{W}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)}$) and given by Algo. 3 when setting $\mathbf{M} := \mathbf{I}_n$. The iterates of this procedure are

$$\mathbf{x}_s^{(j)} \in \mathbf{x}_s^{(0)} + \mathcal{K}_j(\mathbf{A}_s, \mathbf{W}_s, \mathbf{r}_s^{(0)}) \quad (9)$$

$$\text{s.t. } \mathbf{r}_s^{(j)} \perp \mathcal{K}_j(\mathbf{A}_s, \mathbf{W}_s, \mathbf{r}_s^{(0)}) \quad (10)$$

with $\mathbf{r}_s^{(0)} \perp \mathcal{R}(\mathbf{W}_s)$ and where $\mathcal{K}_j(\mathbf{A}_s, \mathbf{W}_s, \mathbf{r}_s^{(0)}) := \mathcal{K}_j(\mathbf{A}_s, \mathbf{r}_s^{(0)}) \oplus \mathcal{R}(\mathbf{W}_s)$.

3.2 Recycling strategy

As we intend to solve the sampled linear systems either by DEF-CG, or by INIT-CG, a procedure needs to be defined in order to compute \mathbf{W}_{s+1} prior to solving $\mathbf{A}_{s+1}\mathbf{x}_{s+1} = \mathbf{b}_{s+1}$. For this, a key observation is that a basis of $\mathcal{K}_j(\mathbf{A}_s, \mathbf{W}_s, \mathbf{r}_s^{(0)})$ can be recycled from the linear solve of $\mathbf{A}_s\mathbf{x}_s = \mathbf{b}_s$. Moreover, different quantities needed to construct projection-based approximations of the eigenvectors of \mathbf{A}_s in these subspaces, such as the tridiagonalization of \mathbf{A}_s , are readily accessible byproducts of the linear solver. Therefore, approximate eigenpairs of \mathbf{A}_s can be promptly generated while solving $\mathbf{A}_s\mathbf{x}_s = \mathbf{b}_s$. Now, because the operators sampled by MCMC are correlated, the eigenvectors of \mathbf{A}_{s+1} are correlated with those of \mathbf{A}_s . For this reason, we let the columns of \mathbf{W}_{s+1} be approximate eigenvectors of \mathbf{A}_s .

All the matrices sampled in this work show greater ratios $\lambda_{\bullet+1}(\mathbf{A}_s)/\lambda_{\bullet}(\mathbf{A}_s)$ in the lower end of the spectrum than in the upper part. Therefore, a more significant decrease of the active condition number can be expected when deflating a few least dominant (LD) eigenpairs, rather than as many of the most dominant (MD) ones. Moreover, the LD eigenvectors correspond to low frequency modes of the sampled operators, as opposed to the MD eigenvectors, which capture more detailed features of the solution. It was observed that the LD eigenvectors of subsequent matrices sampled by MCMC are more correlated than the MD ones. For these reasons, we let the columns of \mathbf{W}_{s+1} be projection-based approximations of LD eigenvectors of \mathbf{A}_s . Note that this strategy is analogous to the approach used in [12].

3.3 Projection techniques for the approximation of eigenvectors

We only intend to approximate a small number $k \ll n$ of the LD eigenvectors of every sampled operator \mathbf{A}_s . Iterative methods based on projection techniques are well suited for this task [18]. In particular, given a full column rank matrix $\mathbf{V}_s \in \mathbb{R}^{n \times m}$ with $k \leq m < n$, we consider both RR and HR projection techniques with respect to the eigen-search space $\mathcal{R}(\mathbf{V}_s)$.

RR procedures—perhaps the most commonly used projection-based techniques for eigenvector computation—are well known to provide optimal eigenvalue approximations at the extremities of the spectrum [14]. They are defined as follows.

RR projection A RR vector \mathbf{w} of \mathbf{A}_s with respect to $\mathcal{R}(\mathbf{V}_s)$ is such that $\mathbf{A}_s\mathbf{w} - \vartheta\mathbf{w} \perp \mathcal{R}(\mathbf{V}_s)$ for some ϑ . As the RR vector is recast as $\mathbf{w} := \mathbf{V}_s\hat{\mathbf{w}}$, the pair $(\vartheta, \hat{\mathbf{w}})$ becomes solution of the reduced (generalized) eigenvalue problem

$$\mathbf{V}_s^T \mathbf{A}_s \mathbf{V}_s \hat{\mathbf{w}} = \vartheta \mathbf{V}_s^T \mathbf{V}_s \hat{\mathbf{w}}. \quad (11)$$

HR procedures were introduced as an alternative to RR projections in order to better approximate the interior eigenvalues of Hermitian operators [7, 11]. They are now commonly defined as follows.

HR projection A HR vector \mathbf{w} of \mathbf{A}_s with respect to $\mathcal{R}(\mathbf{V}_s)$ is such that $\mathbf{A}_s \mathbf{w} - \vartheta \mathbf{w} \perp \mathcal{R}(\mathbf{A}_s \mathbf{V}_s)$ for some ϑ . As the HR vector is recast as $\mathbf{w} := \mathbf{V}_s \hat{\mathbf{w}}$, the pair $(\vartheta, \hat{\mathbf{w}})$ becomes solution of the following reduced generalized eigenvalue problem:

$$(\mathbf{A}_s \mathbf{V}_s)^T \mathbf{A}_s \mathbf{V}_s \hat{\mathbf{w}} = \vartheta (\mathbf{A}_s \mathbf{V}_s)^T \mathbf{V}_s \hat{\mathbf{w}}. \quad (12)$$

In this work, we let the columns of \mathbf{V}_s be previous eigenvector approximations, if any, along with search directions or residuals generated by the linear solve. Approximations of the k LD eigenvectors of \mathbf{A}_s are formed by, first, solving for the k LD eigenpairs $\{(\vartheta_i, \hat{\mathbf{w}}_i)\}_{i=1}^k$ of the reduced (RR or HR) eigenvalue problem, and then, letting $\mathbf{W}_{s+1} := \mathbf{V}_s [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_k]$.

A procedure based on HR projections was favored in [19] to approximate the LD eigenvectors of SPD matrices. Notwithstanding, whether or not HR vectors better approximate the LD eigenvectors of an SPD matrix than their RR counterparts remains, to our knowledge, an open question. Irrespective of which projection technique is used, high-dimensional eigen-search spaces are generally needed in order to obtain accurate eigenvector approximations. However, since the computed approximate eigenvectors of \mathbf{A}_s are only used to deflate \mathbf{A}_{s+1} , the need for highly accurate approximations is not justified in the context of this study.

3.4 Refreshing the eigen-search space

Although the computed vectors need not be exact eigenvectors of \mathbf{A}_s , the better these vectors approximate the eigenvectors of \mathbf{A}_{s+1} , the better the convergence behaviors of DEF-CG [6] and INIT-CG [3]. Up to some level, which depends on $\delta \mathbf{A}_{s+1}$, this may be achieved by increasing the dimension of the eigen-search space, which might cause memory problems because \mathbf{V}_s , which is dense, needs to be stored in order to compute \mathbf{W}_{s+1} . Moreover, due to the effects of finite arithmetic, the linear independence of the recycled residuals (or search directions) tends not to hold anymore as the dimension of the eigen-search space increases. This, in turn, can cause a loss of rank in the matrices of the reduced eigenvalue problem, eventually leading to spurious eigenvector approximations [10]. As a means to circumvent these difficulties, the recycled vectors can be orthogonalized, at a computational cost which increases quadratically with the dimension of the eigen-search space. An alternative which allows for a better use of computational resources is to periodically refresh the eigen-search space with updated eigenvector approximations of \mathbf{A}_s , every time the dimension of the eigen-search space reaches a certain limit. We refer to these approaches as refreshed methods. To avoid any confusion, note that the linear solve is *not* restarted, as opposed to what is done when restarting FOM or GMRES [17].

3.4.1 Thick-refresh

The term *thick-restart* was first coined in [22] to refer to an explicitly restarted Lanczos procedure which we summarize as follows. First, a sequence of Lanczos vectors $\mathbf{v}_1, \dots, \mathbf{v}_{j+1}$ is generated, which is used to compute k RR vectors $\mathbf{w}_1, \dots, \mathbf{w}_k$ of \mathbf{A}_s with respect to $\mathcal{R}(\mathbf{V}_s)$, where $\mathbf{V}_s := [\mathbf{v}_1, \dots, \mathbf{v}_j]$ and $k < j$. A restart is then instantiated by setting the new basis $\mathbf{V}_s := [\mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{v}_{j+1}]$, to which new vectors are subsequently appended by an augmented Krylov subspace method [1, 16] while using a Gram-Schmidt procedure to enforce the orthogonality of the whole basis. Every time the basis reaches a certain size *spdim*, a new set of RR vectors is computed and a restart is instantiated. As a result, the eigen-search space $\mathcal{R}(\mathbf{V}_s)$ is and remains a Krylov subspace of \mathbf{A}_s throughout the procedure. Moreover, this restart strategy ensures that the growing eigen-search space simultaneously contains growing Krylov subspaces generated by every single of the most recently computed RR vectors. This property is believed to explain the effectiveness of the method [8].

The thick-restart Lanczos method of [22] is a dedicated eigensolver. Here, we merely intend to make use of all the residuals (or search directions) generated by the linear solver without applying an augmented Krylov subspace strategy to the eigenvector approximations, as this would entail an unwanted restart of the linear solver. Therefore, the eigen-search space is simply *refreshed* by appending the residuals (or search directions) of the linear solver to the most recently computed eigenvector approximations. The resulting strategy, referred to as thick-refresh (TR) and summarized in Algo. 1, is run concurrently with CG, DEF-CG or INIT-CG. As a result, the refreshed eigen-search space does not remain a Krylov subspace, and the convenient property of thick-restart Lanczos is not satisfied. Nevertheless, this strategy does allow to compute eigenvector approximations of \mathbf{A}_s by making use of all the residuals (or search directions) generated by the linear solver while avoiding the difficulties described in Section 3.4.

```

1: if  $s = 1$  then
2:    $\mathbf{V}_s := []$ ,  $d := 0$ 
3: else if  $s > 1$  then
4:    $\mathbf{V}_s := \mathbf{W}_s$ ,  $d := k$ 
5: end if
6: for  $j = 0, 1, \dots$  do
7:    $\mathbf{V}_s := [\mathbf{V}_s, \mathbf{r}_s^{(j)}]$ ,  $d := d + 1$ 
8:   if  $d = \text{spdim}$  then
9:     Compute the  $k$  LD RR (or HR) vectors  $\{\mathbf{w}_i\}_{i=1}^k$  of  $\mathbf{A}_s$  w.r.t.  $\mathcal{R}(\mathbf{V}_s)$ 
10:     $\mathbf{V}_s := [\mathbf{w}_1, \dots, \mathbf{w}_k]$ ,  $d := k$ 
11:   end if
12: end for
13:  $\mathbf{W}_{s+1} := [\mathbf{w}_1, \dots, \mathbf{w}_k]$ 

```

Algorithm 1: Thick-refresh (TR) of the eigen-search space

3.4.2 Locally optimal thick-refresh

An alternative use of all the information generated by the linear solver is described in Algo. 2, which we analogously refer to as a locally optimal thick-refresh (LO-TR) of the eigen-search space. LO-TR essentially works as follows. Every time the eigen-search space $\mathcal{R}(\mathbf{V}_s)$ reaches a dimension spdim , it is used to generate eigenvector approximations $\mathbf{y}_1, \dots, \mathbf{y}_k$, whereas additional eigenvector approximations $\bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_k$ are computed with respect to $\mathcal{R}(\bar{\mathbf{V}}_s)$, where $\bar{\mathbf{V}}_s := \mathbf{V}_s[:, 1 : \text{spdim} - 1]$ consists of all but the last column of \mathbf{V}_s . All the eigenvector approximations of \mathbf{A}_s with respect to $\mathcal{R}([\mathbf{y}_1, \bar{\mathbf{y}}_1, \dots, \mathbf{y}_k, \bar{\mathbf{y}}_k])$ are then used to refresh the eigen-search space. When applied concurrently with INIT-CG using RR projections, this strategy is equivalent to eigCG [20]. Not only does this method outperform thick-restart Lanczos procedures for eigenvalue approximation, but it was also shown to perform as well as orthogonalized un-restarted Lanczos procedures under favorable conditions [20].

To the best of our knowledge, among all the refreshing strategies described in this Section, RR-LO-TR-INIT-CG (i.e. eigCG) is the only one documented in the literature.

3.5 Practical considerations

We assume that $k \ll n$ so that the extra computational cost of DEF-CG compared to CG is mainly due to the computation of $\mathbf{\Pi}_s \mathbf{r}_s^{(j)}$, which can be done at a cost of $O(kn)$ per iteration when both \mathbf{W}_s and $\mathbf{A}_s \mathbf{W}_s$ are stored [19]. However, INIT-CG, which is CG with a particular initial

```

1: if  $s = 1$  then
2:    $\mathbf{V}_s := []$ ,  $d := 0$ 
3: else if  $s > 1$  then
4:    $\mathbf{V}_s := \mathbf{W}_s$ ,  $d := k$ 
5: end if
6: for  $j = 0, 1, \dots$  do
7:    $\mathbf{V}_s := [\mathbf{V}_s, \mathbf{r}_s^{(j)}]$ ,  $d := d + 1$ 
8:   if  $d = \text{spdim}$  then
9:      $\overline{\mathbf{V}}_s := \mathbf{V}_s[:, 1 : \text{spdim} - 1]$   $\triangleright \overline{\mathbf{V}}_s$  contains all but the last column of  $\mathbf{V}_s$ 
10:    Compute the  $k$  LD RR (or HR) vectors  $\{\mathbf{y}_i\}_{i=1}^k$  of  $\mathbf{A}_s$  w.r.t.  $\mathcal{R}(\mathbf{V}_s)$ 
11:    Compute the  $k$  LD RR (or HR) vectors  $\{\overline{\mathbf{y}}_i\}_{i=1}^k$  of  $\mathbf{A}_s$  w.r.t.  $\mathcal{R}(\overline{\mathbf{V}}_s)$ 
12:    Get an orthonormal basis  $\mathbf{Q} := [\mathbf{q}_1, \dots, \mathbf{q}_{nvec}]$  of  $\mathcal{R}([\mathbf{y}_1, \overline{\mathbf{y}}_1, \dots, \mathbf{y}_k, \overline{\mathbf{y}}_k])$ 
13:    Compute the RR (or HR) vectors  $\{\mathbf{w}_i\}_{i=1}^{nvec}$  of  $\mathbf{A}_s$  w.r.t.  $\mathcal{R}(\mathbf{Q})$ 
14:     $\mathbf{V}_s := [\mathbf{w}_1, \dots, \mathbf{w}_{nvec}]$ ,  $d := nvec$ 
15:   end if
16: end for
17:  $\mathbf{W}_{s+1} := [\mathbf{w}_1, \dots, \mathbf{w}_k]$ 

```

Algorithm 2: Locally optimal thick-refresh (LO-TR) of the eigen-search space

guess, only requires one extra computation to set $\mathbf{x}_s^{(0)}$ such that $\mathbf{r}_s^{(0)} \perp \mathcal{R}(\mathbf{W}_s)$. Hence, there is an incentive to investigate whether the convergence behavior of DEF-CG can be reproduced by INIT-CG, when applied to sequences of correlated operators.

3.5.1 Assembly of the reduced eigenvalue problem

The assembly of the reduced eigenvalue problem depends on the type of projection (RR or HR) used, as well as on whether residuals or search directions are stored in \mathbf{V}_s . Indeed, although Algos. 1 and 2 both suggest to extract residuals from the solver run, these vectors can be replaced by search directions as they both span the same subspace Krylov subspace. Irrespective of which approach is taken, the matrices $\mathbf{A}_s \mathbf{W}_s$ and $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s$ need to be computed prior to assembly.

Then, for RR projections, the cost of assembly is minimized when we extract sequences of $\ell := \text{spdim} - k$ normalized residuals $\mathbf{r}_s^{(j)} / \|\mathbf{r}_s^{(j)}\|_2$, which we store by columns in \mathbf{R}_s . Then, we have $\mathbf{V}_s = [\mathbf{W}_s, \mathbf{R}_s]$ so that Eq. (11) becomes:

$$\begin{bmatrix} \mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{W}_s^T \mathbf{A}_s \mathbf{R}_s \\ \mathbf{R}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{R}_s^T \mathbf{A}_s \mathbf{R}_s \end{bmatrix} \hat{\mathbf{w}} = \vartheta \begin{bmatrix} \mathbf{I}_k & \mathbf{W}_s^T \mathbf{R}_s \\ \mathbf{R}_s^T \mathbf{W}_s & \mathbf{I}_\ell \end{bmatrix} \hat{\mathbf{w}} \quad (13)$$

where $\mathbf{R}_s^T \mathbf{A}_s \mathbf{R}_s$ is tridiagonal with coefficients which can be recovered as a byproduct of the solver. Computing the product $\mathbf{W}_s^T \mathbf{A}_s \mathbf{R}_s$ can be done column-by-column as the residuals are being generated by the solver, for a total cost of $O(k \ell n)$. The off-diagonal block $\mathbf{W}_s^T \mathbf{R}_s$ vanishes as a property of DEF-CG. For INIT-CG, this term vanishes if $\mathcal{R}(\mathbf{W}_s)$ is invariant under the action of \mathbf{A}_s , which would be the case if the columns of \mathbf{W}_s were exact eigenvectors of \mathbf{A}_s . Note that this term is ignored in [20].

When using HR projections, a simpler problem is obtained when extracting sequences of ℓ search directions $\mathbf{p}_s^{(j)}$, which we store by columns in \mathbf{P}_s . Then, we have $\mathbf{V}_s = [\mathbf{W}_s, \mathbf{P}_s]$ so that Eq. (12) is recast as:

$$\begin{bmatrix} (\mathbf{A}_s \mathbf{W}_s)^T \mathbf{A}_s \mathbf{W}_s & (\mathbf{A}_s \mathbf{W}_s)^T \mathbf{A}_s \mathbf{P}_s \\ (\mathbf{A}_s \mathbf{P}_s)^T \mathbf{A}_s \mathbf{W}_s & (\mathbf{A}_s \mathbf{P}_s)^T \mathbf{A}_s \mathbf{P}_s \end{bmatrix} \hat{\mathbf{w}} = \vartheta \begin{bmatrix} \mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{W}_s^T \mathbf{A}_s \mathbf{P}_s \\ \mathbf{P}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{P}_s^T \mathbf{A}_s \mathbf{P}_s \end{bmatrix} \hat{\mathbf{w}} \quad (14)$$

in which the computation of $(\mathbf{A}_s \mathbf{W}_s)^T \mathbf{A}_s \mathbf{W}_s$ costs $O(k^2 n)$. The matrices $(\mathbf{A}_s \mathbf{P}_s)^T \mathbf{A}_s \mathbf{P}_s$ and $\mathbf{P}_s^T \mathbf{A}_s \mathbf{P}_s$ are tridiagonal and diagonal, respectively, and may be recovered as byproducts of the solver. Then, by property of DEF-CG, $\mathbf{W}_s^T \mathbf{A}_s \mathbf{P}_s$ vanishes, while $(\mathbf{A}_s \mathbf{W}_s)^T \mathbf{A}_s \mathbf{P}_s$ may be computed at negligible cost, in the way described by [19]. Otherwise, each of these two products has an additional cost of $O(k \ell n)$.

3.5.2 Refreshing the eigen-search space

Every refresh of the eigen-search space entails a new assembly at a cost comparable to what is described in Section 3.5.1. Note that the orthogonalization step (line 12) of Algo. 2 can conveniently be performed between the reduced eigenvectors, as described in [20].

3.6 Numerical results

We consider a 1,000-long sequence of 500-by-500 linear systems obtained by setting $u(0) := 0$ and $\partial_x u(1) := 0$ with the 1D coefficient fields presented in Fig. 1 and a constant f . These systems are solved by CG, and the convergence histories of the norm of the iterated residual are presented in Fig. 3. Because the sampled coefficients are highly heterogeneous, the corresponding operators are ill-conditioned, and CG systematically converges after more than n iterations. Note that, while the first linear systems of this particular sampled sequence require more iterations to solve, this is not generally true of all sequences.

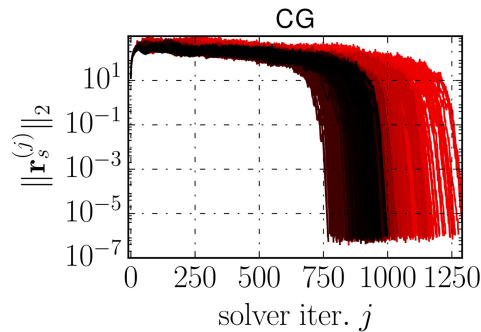


Figure 3: Norm of iterated residuals obtained by CG for a sequence of 1,000 linear systems with 500 degrees of freedom (DoFs). Coefficient field of Case 1. System index s gradually color coded from first ($s = 1$, \color{red}) to last ($s = 1000$, \color{black}) in sequence.

The same sequence of linear systems is solved by DEF-CG and INIT-CG using $k = 10$ RR eigenvector approximations with and without refreshing the eigen-search space. The resulting convergence histories are presented in Fig. 4. The results obtained by DEF-CG are on the first row of the graph, and those of INIT-CG on the second row. Analogous results obtained using as many HR eigenvector approximations are presented in Fig. 5. All the eigen-search spaces are kept at a dimension $spdim = 40$.

We observe the following. First, despite a high correlation between realizations of the coefficient field—see the similarity between $a(x; \boldsymbol{\xi}_{s-10})$, $a(x; \boldsymbol{\xi}_s)$ and $a(x; \boldsymbol{\xi}_{s+10})$ in Fig. 1—all the INIT-CG procedures recover the behavior of regular CG once the norm of the iterated residual reaches a value smaller than 10^{-1} . RR-LO-TR-INIT-CG, known as eigCG [20], provides the best convergence behavior before the norm of the iterated residual reaches this value. Second, all the DEF-CG procedures show significant improvements in their convergence behavior between

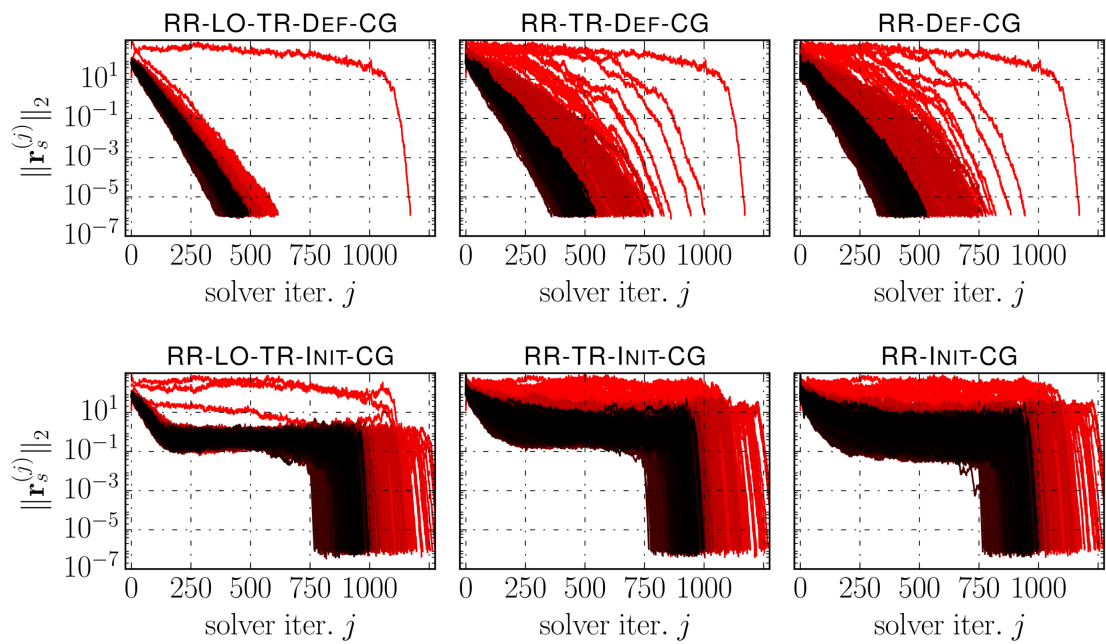


Figure 4: Norm of iterated residuals obtained by DEF-CG (top row) and INIT-CG (bottom row) procedures using RR projections for a sequence of 1,000 linear systems with 500 DoFs. Coefficient field of Case 1. System index s gradually color coded from first ($s = 1$, —) to last ($s = 1000$, —) in sequence.

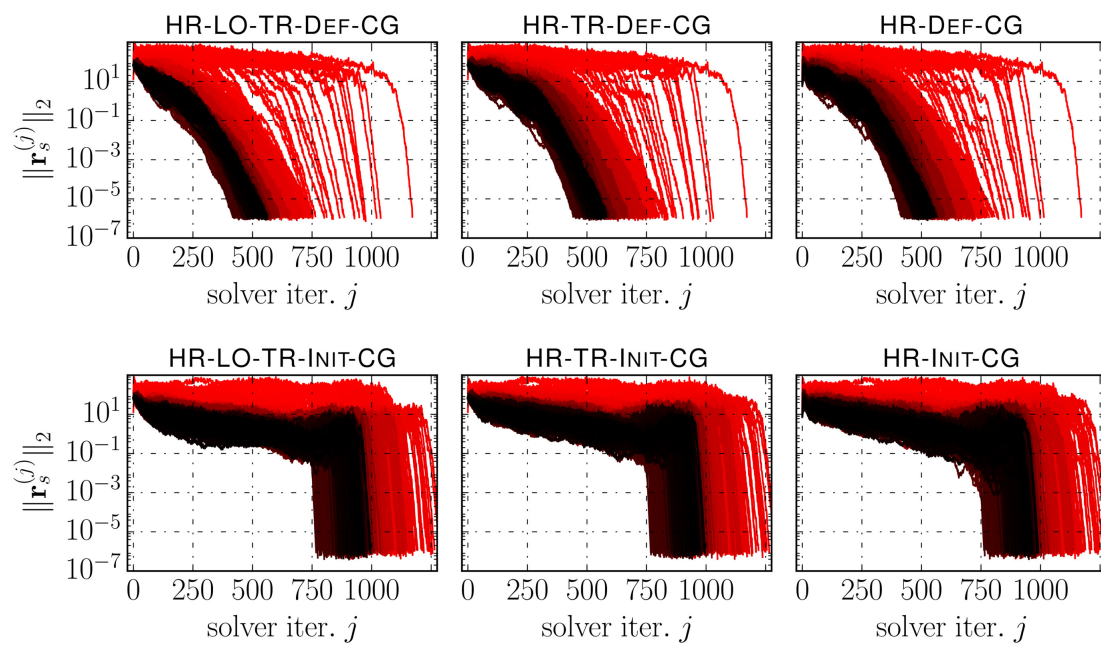


Figure 5: Norm of iterated residuals obtained by DEF-CG (top row) and INIT-CG (bottom row) procedures using HR projections for a sequence of 1,000 linear systems with 500 DoF. Coefficient field of Case 1. System index s gradually color coded from first ($s = 1$, \color{red}) to last ($s = 1000$, \color{black}) in sequence.

the beginning and the end of the sampled sequence. Despite the fact that the operators are different in consecutive linear systems, it does seem that those remain sufficiently correlated so that the approximated eigenvectors are accurate enough for the deflation to be effective. However, the convergence behaviors observed toward the end of the sampled sequence, i.e., the black curves, have a different shape depending on whether RR or HR projections are used—RR projections showing slightly faster convergence. Third, only one refreshing strategy seems to be effective, namely, RR-LO-TR-DEF-CG.

4 Deflated Krylov subspace strategies with fixed preconditioners

Let \mathbf{M} be an SPD preconditioner, constant throughout the sequence, with Cholesky decomposition $\mathbf{L}\mathbf{L}^T$. The preconditioned conjugate gradient algorithm (PCG) formally consists of a transformed sequence of iterates approximating the solution of the split preconditioned system $\dot{\mathbf{A}}_s \dot{\mathbf{x}}_s = \dot{\mathbf{b}}_s$ where $\dot{\mathbf{A}}_s := \mathbf{L}^{-1} \mathbf{A}_s \mathbf{L}^{-T}$ and $\dot{\mathbf{b}}_s := \mathbf{L}^{-1} \mathbf{b}_s$ so that $\dot{\mathbf{x}}_s = \mathbf{L}^T \mathbf{x}_s$ in which \mathbf{x}_s is the solution of the original system $\mathbf{A}_s \mathbf{x}_s = \mathbf{b}_s$.

4.1 Deflation

Here, we present how preconditioning is combined with deflation to form the DEF-PCG algorithm [19] as a means to guide us in the formulation of relevant approximate spectral information. To do so, we rely on the definition of a projector $\dot{\mathbf{\Pi}}_s := \mathbf{I}_n - \dot{\mathbf{W}}_s (\dot{\mathbf{W}}_s^T \dot{\mathbf{A}}_s \dot{\mathbf{W}}_s)^{-1} \dot{\mathbf{W}}_s^T \dot{\mathbf{A}}_s$ with a full column rank matrix $\dot{\mathbf{W}}_s \in \mathbb{R}^{n \times k}$. Then, $\dot{\mathbf{x}}_s$ admits the decomposition

$$\dot{\mathbf{x}}_s = (\mathbf{I}_n - \dot{\mathbf{\Pi}}_s) \dot{\mathbf{x}}_s + \dot{\mathbf{\Pi}}_s \dot{\mathbf{x}}_s. \quad (15)$$

It follows from the definition of $\dot{\mathbf{\Pi}}_s$ that, in Eq. (15), we have

$$(\mathbf{I}_n - \dot{\mathbf{\Pi}}_s) \dot{\mathbf{x}}_s = \dot{\mathbf{W}}_s \dot{\boldsymbol{\mu}}_s \quad (16)$$

where $\dot{\boldsymbol{\mu}}_s$ is such that $\dot{\mathbf{W}}_s^T \dot{\mathbf{A}}_s \dot{\mathbf{W}}_s \dot{\boldsymbol{\mu}}_s = \dot{\mathbf{W}}_s^T \dot{\mathbf{b}}_s$. Meanwhile, $\dot{\mathbf{\Pi}}_s \dot{\mathbf{x}}_s$ is equivalently given by $\dot{\mathbf{\Pi}}_s \hat{\mathbf{x}}_s$, where $\hat{\mathbf{x}}_s$ is any of the infinitely many solutions of the preconditioned and deflated system

$$\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s \hat{\mathbf{x}}_s = \dot{\mathbf{\Pi}}_s^T \dot{\mathbf{b}}_s. \quad (17)$$

Let $\{\hat{\mathbf{x}}_s^{(j)}\}_{j=0}^m$ be the sequence generated by CG($\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s$, $\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{b}}_s$, $\hat{\mathbf{x}}_s^{(0)}$) for some $\hat{\mathbf{x}}_s^{(0)}$ such that $\hat{\mathbf{r}}_s^{(0)} := \dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s \hat{\mathbf{x}}_s^{(0)} - \dot{\mathbf{\Pi}}_s^T \dot{\mathbf{b}}_s \perp \mathcal{R}(\dot{\mathbf{W}}_s)$. Then, the post-processed sequence $\mathbf{x}_s^{(j)} := \mathbf{L}^{-T} (\dot{\mathbf{W}}_s \dot{\boldsymbol{\mu}}_s + \dot{\mathbf{\Pi}}_s \hat{\mathbf{x}}_s^{(j)})$ converges to the solution \mathbf{x}_s of the original system at a rate bounded as in Eq. (4), but where the active condition number is evaluated over the strictly positive eigenvalues of $\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s$. In practice, the sequence $\{\mathbf{x}_s^{(j)}\}_{j=0}^m$ is obtained using the DEF-PCG algorithm, which is derived upon specifying $\mathbf{W}_s := \mathbf{L}^{-T} \dot{\mathbf{W}}_s$ among other changes of variables [19]. Note that one does not need to know the Cholesky factor \mathbf{L} to execute DEF-PCG, nor to specify $\mathbf{x}_s^{(0)}$ such that $\hat{\mathbf{r}}_s^{(0)} \perp \mathcal{R}(\dot{\mathbf{W}}_s)$, because the latter is implied by $\mathbf{r}_s^{(0)} := \mathbf{A}_s \mathbf{x}_s^{(0)} - \mathbf{b}_s \perp \mathcal{R}(\mathbf{W}_s)$. Hence, if an initial guess $\mathbf{x}_s^{(0)}$ is specified such that $\mathbf{r}_s^{(0)} \perp \mathcal{R}(\mathbf{W}_s)$, then the iterates generated by DEF-PCG(\mathbf{A}_s , \mathbf{M} , \mathbf{W}_s , \mathbf{b}_s , $\mathbf{x}_s^{(0)}$), see Algo. 3, are given by:

$$\mathbf{x}_s^{(j)} \in \mathbf{x}_s^{(0)} + \mathbf{L}^{-T} \mathcal{K}_j(\dot{\mathbf{A}}_s, \dot{\mathbf{W}}_s, \mathbf{L}^{-1} \mathbf{r}_s^{(0)}) \quad (18)$$

$$\text{s.t. } \mathbf{r}_s^{(j)} \perp \mathbf{L}^{-T} \mathcal{K}_j(\dot{\mathbf{A}}_s, \dot{\mathbf{W}}_s, \mathbf{L}^{-1} \mathbf{r}_s^{(0)}) \quad (19)$$

where $\mathcal{K}_j(\dot{\mathbf{A}}_s, \dot{\mathbf{W}}_s, \mathbf{L}^{-1}\mathbf{r}_s^{(0)}) = \mathcal{K}_j(\dot{\mathbf{A}}_s, \mathbf{L}^{-1}\mathbf{r}_s^{(0)}) \oplus \mathcal{R}(\dot{\mathbf{W}}_s)$.

From hereon, there remains to specify procedures for the computation of \mathbf{W}_s . In doing so, our intent is to decrease the effective condition number $\kappa(\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s)$ as much as possible in comparison to $\kappa(\mathbf{A}_s)$. However, since $\mathbf{W}_s = \mathbf{L}^{-T} \dot{\mathbf{W}}_s$ implies $\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s = \mathbf{L}^{-1} \dot{\mathbf{\Pi}}_s^T \mathbf{A}_s \mathbf{L}^{-T}$, two types of approaches arise. In the first approach, we let the columns of \mathbf{W}_s be approximate eigenvectors of \mathbf{A}_s , in which case it is convenient to think of $\kappa(\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s)$ as the effective condition number of an operator which is deflated prior to be split preconditioned. The second approach consists of letting the columns of $\dot{\mathbf{W}}_s$ be approximate eigenvectors of $\dot{\mathbf{A}}_s$, while still only using \mathbf{W}_s in DEF-PCG. In this case, $\kappa(\dot{\mathbf{\Pi}}_s^T \dot{\mathbf{A}}_s)$ is thought of as the effective condition number of an operator which is split preconditioned prior to be deflated. Note that both approaches are not equivalent. Indeed, if \mathbf{w} is an eigenvector of \mathbf{A}_s , then $\dot{\mathbf{w}} := \mathbf{L}^T \mathbf{w}$ is a right eigenvector of $\mathbf{L}^T \mathbf{A}_s \mathbf{L}^{-T}$, but generally not of $\dot{\mathbf{A}}_s$. Conversely, if $\dot{\mathbf{w}}$ is an eigenvector of $\dot{\mathbf{A}}_s$, then $\mathbf{w} := \mathbf{L}^{-T} \dot{\mathbf{w}}$ is a right eigenvector of $\mathbf{M}^{-1} \mathbf{A}_s$, but generally not of \mathbf{A}_s .

Both approaches were investigated in the work that lead to this study, and no conclusive observation could be made on which method leads to the best behavior of DEF-PCG. However, the following can be said about the second approach: (i) unlike the first method, it seeks to approximate eigenvectors which are consistent with the Krylov subspace spanned by the vectors extracted from DEF-PCG; (ii) it yields reduced generalized eigenvalue problems which are less computationally demanding to assemble. Therefore, we only focus on the second approach, which corresponds to the approach that was originally proposed in [19].

```

1:  $\mathbf{r}_s^{(0)} := \mathbf{b}_s - \mathbf{A}_s \mathbf{x}_s^{(0)}$ 
2:  $\mathbf{z}_s^{(0)} := \mathbf{M}^{-1} \mathbf{r}_s^{(0)}$ 
3: Solve for  $\boldsymbol{\mu}_s^{(0)}$  in  $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s \boldsymbol{\mu}_s^{(0)} = \mathbf{W}_s^T \mathbf{A}_s \mathbf{z}_s^{(0)}$ 
4:  $\mathbf{p}_s^{(0)} := \mathbf{z}_s^{(0)} - \mathbf{W}_s \boldsymbol{\mu}_s^{(0)}$ 
5: for  $j = 1, 2, \dots$  do
6:    $\alpha_s^{(j-1)} := \mathbf{r}_s^{(j-1)T} \mathbf{z}_s^{(j-1)} / \mathbf{p}_s^{(j-1)T} \mathbf{A}_s \mathbf{p}_s^{(j-1)}$ 
7:    $\mathbf{x}_s^{(j)} := \mathbf{x}_s^{(j-1)} + \alpha_s^{(j-1)} \mathbf{p}_s^{(j-1)}$ 
8:    $\mathbf{r}_s^{(j)} := \mathbf{r}_s^{(j-1)} - \alpha_s^{(j-1)} \mathbf{A}_s \mathbf{p}_s^{(j-1)}$ 
9:    $\mathbf{z}_s^{(j)} := \mathbf{M}^{-1} \mathbf{r}_s^{(j)}$ 
10:  Solve for  $\boldsymbol{\mu}_s^{(j)}$  in  $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s \boldsymbol{\mu}_s^{(j)} = \mathbf{W}_s^T \mathbf{A}_s \mathbf{z}_s^{(j)}$ 
11:   $\beta_s^{(j-1)} := \mathbf{r}_s^{(j)T} \mathbf{z}_s^{(j)} / \mathbf{r}_s^{(j-1)T} \mathbf{z}_s^{(j-1)}$ 
12:   $\mathbf{p}_s^{(j)} := \beta_s^{(j-1)} \mathbf{p}_s^{(j-1)} + \mathbf{z}_s^{(j)} - \mathbf{W}_s \boldsymbol{\mu}_s^{(j)}$ 
13: end for

```

$\triangleright \mathbf{x}_s^{(0)}$ is set s.t. $\mathbf{W}_s^T \mathbf{r}_s^{(0)} = \mathbf{0}$

Algorithm 3: DEF-PCG($\mathbf{A}_s, \mathbf{M}, \mathbf{W}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)}$)

4.2 Projection techniques for the approximation of eigenvectors

Let us extract sequences of preconditioned residuals (or search directions) from the linear solver and store them by columns in \mathbf{V}_s , along with \mathbf{W}_s which may be refreshed or not. Then, setting $\dot{\mathbf{V}}_s := \mathbf{L}^T \mathbf{V}_s$ allows us to construct an eigen-search space $\mathcal{R}(\dot{\mathbf{V}}_s)$ consisting of a Krylov subspace of $\dot{\mathbf{A}}_s$ augmented by $\mathcal{R}(\dot{\mathbf{W}}_s)$. We then intend to compute approximations $\{\mathbf{w}_i\}_{i=1}^k$ of the k LD right eigenvectors of $\mathbf{M}^{-1} \mathbf{A}_s$, which we store by columns in \mathbf{W}_{s+1} . This is done by first computing LD approximate eigenvectors $\dot{\mathbf{w}}_i$ of $\dot{\mathbf{A}}_s$ with respect to $\mathcal{R}(\dot{\mathbf{V}}_s)$, before setting $\mathbf{w}_i := \mathbf{L}^{-T} \dot{\mathbf{w}}_i$. As we proceed to do so using RR and HR projections, we note that the Cholesky factor \mathbf{L} is in fact not needed in either of the resulting procedures.

RR projection Let $\dot{\mathbf{w}}$ be a RR eigenvector approximation of $\dot{\mathbf{A}}_s$ with respect to $\mathcal{R}(\dot{\mathbf{V}}_s)$. We then search for $\dot{\mathbf{w}} \in \mathcal{R}(\dot{\mathbf{V}}_s)$ such that $\dot{\mathbf{A}}_s \dot{\mathbf{w}} - \vartheta \dot{\mathbf{w}} \perp \mathcal{R}(\dot{\mathbf{V}}_s)$, for some ϑ . Recasting $\dot{\mathbf{w}}$ as $\dot{\mathbf{V}}_s \hat{\mathbf{w}}$, we obtain $\dot{\mathbf{V}}_s^T \dot{\mathbf{A}}_s \dot{\mathbf{V}}_s \hat{\mathbf{w}} = \vartheta \dot{\mathbf{V}}_s^T \dot{\mathbf{V}}_s \hat{\mathbf{w}}$, which can be recast as

$$\mathbf{V}_s^T \mathbf{A}_s \mathbf{V}_s \hat{\mathbf{w}} = \vartheta \mathbf{V}_s^T \mathbf{M} \mathbf{V}_s \hat{\mathbf{w}}. \quad (20)$$

Then, we have $\mathbf{w} = \mathbf{L}^{-T} \dot{\mathbf{V}}_s \hat{\mathbf{w}} = \mathbf{V}_s \hat{\mathbf{w}}$, so that this procedure is equivalent to searching for $\mathbf{w} \in \mathcal{R}(\mathbf{V}_s)$ such that $\mathbf{A}_s \mathbf{w} - \vartheta \mathbf{M} \mathbf{w} \perp \mathcal{R}(\mathbf{V}_s)$. Note that, in practice, \mathbf{M} does not need to be applied to assemble the matrix on the right hand side of Eq. (20), as explained in Section 4.3.

HR projection Alternatively, Saad et al. [19] consider $\dot{\mathbf{w}}$ to be an HR eigenvector approximation of $\dot{\mathbf{A}}_s$ with respect to $\mathcal{R}(\dot{\mathbf{V}}_s)$. Then, $\dot{\mathbf{w}} \in \mathcal{R}(\dot{\mathbf{V}}_s)$ is such that $\dot{\mathbf{A}}_s \dot{\mathbf{w}} - \vartheta \dot{\mathbf{w}} \perp \mathcal{R}(\dot{\mathbf{A}}_s \dot{\mathbf{V}}_s)$ for some ϑ . Recasting $\dot{\mathbf{w}}$ as $\dot{\mathbf{V}}_s \hat{\mathbf{w}}$ leads to a reduced eigenvalue problem $(\dot{\mathbf{A}}_s \dot{\mathbf{V}}_s)^T \dot{\mathbf{A}}_s \dot{\mathbf{V}}_s \hat{\mathbf{w}} = \vartheta (\dot{\mathbf{A}}_s \dot{\mathbf{V}}_s)^T \dot{\mathbf{V}}_s \hat{\mathbf{w}}$, which can be recast as

$$(\mathbf{A}_s \mathbf{V}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{V}_s \hat{\mathbf{w}} = \vartheta (\mathbf{A}_s \mathbf{V}_s)^T \mathbf{V}_s \hat{\mathbf{w}}. \quad (21)$$

Then, once again, we have $\mathbf{w} = \mathbf{V}_s \hat{\mathbf{w}}$ so that this procedure is equivalent to searching for $\mathbf{w} \in \mathcal{R}(\mathbf{V}_s)$ such that $\mathbf{M}^{-1} \mathbf{A}_s \mathbf{w} - \vartheta \mathbf{w} \perp \mathcal{R}(\mathbf{A}_s \mathbf{V}_s)$.

In summary, \mathbf{W}_{s+1} is computed by first solving for the k LD eigenpairs $\{(\vartheta_i, \hat{\mathbf{w}}_i)\}_{i=1}^k$ of the reduced (RR or HR) eigenvalue problem, and then, letting $\mathbf{W}_{s+1} := \mathbf{V}_s [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_k]$. Finally, the refreshing strategies of Section 3.4 are applied similarly using the projection methods described this Section.

4.3 Practical considerations

Given the poor convergence behaviors of the INIT-CG solvers reported in Section 3.6, we only consider the implementation details of DEF-PCG($\mathbf{A}_s, \mathbf{M}, \mathbf{W}_s, \mathbf{b}_s, \mathbf{x}_s^{(0)}$). First, both matrices $\mathbf{A}_s \mathbf{W}_s$ and $\mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s$ are computed prior to the first solver iteration and stored at memory costs of kn and k^2 , respectively. Then, the computational cost of assembling the reduced (generalized) eigenvalue problems is minimized as follows for RR and HR projections.

For RR projections, we extract sequences of normalized preconditioned residuals $\mathbf{z}_s^{(j)} / (\mathbf{r}_s^{(j)T} \mathbf{z}_s^{(j)})^{1/2}$ which we store by columns in \mathbf{Z}_s . Then, we have $\mathbf{V}_s = [\mathbf{W}_s, \mathbf{Z}_s]$ so that Eq. (20) is recast as:

$$\begin{bmatrix} \mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{W}_s^T \mathbf{A}_s \mathbf{Z}_s \\ \mathbf{Z}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{Z}_s^T \mathbf{A}_s \mathbf{Z}_s \end{bmatrix} \hat{\mathbf{w}} = \vartheta \begin{bmatrix} \mathbf{W}_s^T \mathbf{M} \mathbf{W}_s & \mathbf{W}_s^T \mathbf{M} \mathbf{Z}_s \\ \mathbf{Z}_s^T \mathbf{M} \mathbf{W}_s & \mathbf{Z}_s^T \mathbf{M} \mathbf{Z}_s \end{bmatrix} \hat{\mathbf{w}}. \quad (22)$$

Note that, by property of DEF-PCG, we have $\mathbf{W}_s^T \mathbf{M} \mathbf{z}_s^{(j)} = \mathbf{W}_s^T \mathbf{r}_s^{(j)} = 0$, and $\mathbf{z}_s^{(i)T} \mathbf{M} \mathbf{z}_s^{(j)} = \delta_{ij} \mathbf{r}_s^{(i)T} \mathbf{z}_s^{(j)}$. Moreover, we have $\mathbf{W}_2 := \mathbf{V}_1 \hat{\mathbf{W}}_1 = \mathbf{Z}_1 \hat{\mathbf{W}}_1$ so that $\mathbf{W}_2^T \mathbf{M} \mathbf{W}_2 = \hat{\mathbf{W}}_1^T \hat{\mathbf{W}}_1$. Therefore, as long as the reduced eigenvectors are orthonormal, the generalized eigenvalue problem simplifies to

$$\begin{bmatrix} \mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{W}_s^T \mathbf{A}_s \mathbf{Z}_s \\ \mathbf{Z}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{Z}_s^T \mathbf{A}_s \mathbf{Z}_s \end{bmatrix} \hat{\mathbf{w}} = \vartheta \hat{\mathbf{w}} \quad (23)$$

for all s , where $\mathbf{Z}_s^T \mathbf{A}_s \mathbf{Z}_s$ is tridiagonal and recovered as a byproduct of the solver. The computation of $\mathbf{W}_s^T \mathbf{A}_s \mathbf{Z}_s$ does not simplify and entails an additional cost of $O(k \ell n)$ every time a reduced eigenvalue problem is assembled. Note that the time complexity of assembling the reduced RR eigenvalue problem is *not* affected by the use of a preconditioner. A more detailed implementation can be found in [20].

For HR projections, the most efficient assembly of the reduced generalized eigenvalue problem is obtained when extracting sequences of search directions $\mathbf{p}_s^{(j)}$ stored by columns in \mathbf{P}_s . Then, we have $\mathbf{V}_s = [\mathbf{W}_s, \mathbf{P}_s]$ so that Eq. (21) is recast as:

$$\begin{bmatrix} (\mathbf{A}_s \mathbf{W}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{W}_s & (\mathbf{A}_s \mathbf{W}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{P}_s \\ (\mathbf{A}_s \mathbf{P}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{W}_s & (\mathbf{A}_s \mathbf{P}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{P}_s \end{bmatrix} \hat{\mathbf{w}} = \vartheta \begin{bmatrix} \mathbf{W}_s^T \mathbf{A}_s \mathbf{W}_s & \mathbf{0}_{k,\ell} \\ \mathbf{0}_{\ell,k} & \mathbf{P}_s^T \mathbf{A}_s \mathbf{P}_s \end{bmatrix} \hat{\mathbf{w}}, \quad (24)$$

where $\mathbf{P}_s^T \mathbf{A}_s \mathbf{P}_s$ and $(\mathbf{A}_s \mathbf{P}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{P}_s$ are diagonal and tridiagonal matrices, respectively, whose coefficients may be extracted as byproducts of the solver. The off-diagonal block $(\mathbf{A}_s \mathbf{W}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{P}_s$ is assembled in the same fashion as the corresponding block in Eq. (14), see [19]. However, $(\mathbf{A}_s \mathbf{W}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{W}_s$ does not simplify. Therefore, the time complexity of assembling the reduced HR generalized eigenvalue value problem is affected by the necessity to perform k preconditioner applications for the computation of $(\mathbf{A}_s \mathbf{W}_s)^T \mathbf{M}^{-1} \mathbf{A}_s \mathbf{W}_s$. Even though these preconditioner applications can be avoided when using RR projections, this was not pointed out in [19].

4.4 Numerical results

We first consider a 200-long sequence of 2,000-by-2,000 linear systems obtained by setting $u(0) := 0$ and $\partial_x u(1) := 0$ with the 1D coefficient fields presented in Fig. 1 and a constant f . These systems are solved by PCG while preconditioning with a single V-cycle of an AMG solver [9] based on the operator $\hat{\mathbf{A}}$ constructed with the median realization, as well as with HR-DEF-PCG using constant block-Jacobi (bJ) preconditioners with 10, 20 and 30 non-overlapping diagonal blocks of $\hat{\mathbf{A}}$. For each instance of HR-DEF-PCG, the number k of approximated eigenvectors is set equal to the number of blocks, whereas the dimension of the eigen-search space is set to $spdim := 2k$. The number of solver iterations to reach a backward error smaller than 10^{-7} is presented on the left side of Fig. 6, whereas the right side of Fig. 6 presents the spectra of \mathbf{A}_s , $\hat{\mathbf{A}}_s$ and, for the instances of HR-DEF-PCG, of $\hat{\Pi}_s^T \hat{\mathbf{A}}_s$, at the end of the sequence, i.e., for $s = 200$.

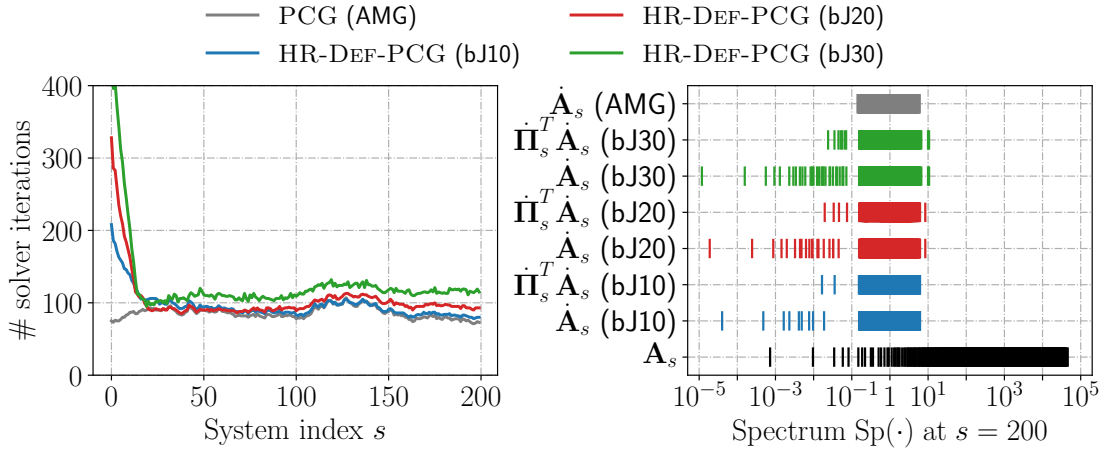


Figure 6: Left side: number of solver iterations needed to reach a backward error smaller than 10^{-7} using HR-DEF-PCG (bJ10, bJ20, bJ30) and PCG (AMG). Right side: original, preconditioned and deflated spectra of \mathbf{A}_s at $s = 200$. Linear systems with 2,000 DoFs and coefficient field of Case 1.

Even though PCG (AMG) uses a constant preconditioner, the resulting spectrum $\text{Sp}(\dot{\mathbf{A}}_s)$ is efficiently condensed between 10^{-1} and 10 for all the realizations, leading to a roughly constant iteration number for this sequence. On the other hand, the spectra $\text{Sp}(\dot{\mathbf{A}}_s)$ associated with the bJ preconditioners mostly consist of two parts: (i) a dense set of eigenvalues ranging from 10^{-1} to 10, and (ii) a trail of as many eigenvalues as the number of blocks used for the preconditioner, spread throughout the lower end of the spectrum. The iteration numbers obtained by HR-DEF-PCG (bJ) at $s = 1$ are the same as what would be obtained using PCG (bJ). As expected, this number increases with the number of blocks. However, as the k approximate LD eigenvectors generated throughout the sequence are used for deflation, the iteration numbers of HR-DEF-PCG (bJ) decrease significantly, and most of the trailing eigenvalues of $\text{Sp}(\dot{\mathbf{A}}_s)$ are “removed” in $\text{Sp}(\ddot{\mathbf{I}}_s^T \dot{\mathbf{A}}_s)$. Note that the few remaining trailing eigenvalues can be removed by increasing the number k of approximated eigenvectors, in which case HR-DEF-PCG (bJ) slightly outperforms PCG (AMG). However this is not the case when the number of blocks is increased beyond 20 or 30 because spdim , which must be larger than k , can not be set to values significantly larger than 60 or 70 without potentially causing some loss of rank in the matrices of the reduced generalized eigenvalue problem.

We now consider a 1,000-long sequence of 4,000-by-4,000 linear systems obtained by setting $u(x) := 0$ on the left edge of a 2D square domain $\Omega := (0, 1)^2$ and a homogeneous Neumann boundary condition on the remaining of $\partial\Omega$, with the coefficient fields presented in Fig. 2 and a constant f . These systems are solved by PCG (AMG) as well as by PCG (bJ), HR-DEF-PCG (bJ) and RR-LO-TR-DEF-PCG (bJ) using constant preconditioners with 10, 20 and 30 non-overlapping blocks of $\hat{\mathbf{A}}$. Note that HR-DEF-PCG and RR-LO-TR-DEF-PCG are considered in particular because thorough details of their implementation can be found in [19] and [20], respectively, and none of the other attempted methods lead to consistently better convergence behaviors than these two. For each instance of deflated solver, the number k of approximated eigenvectors is set equal to the number of blocks, and the dimension of the eigen-search space is set to $\text{spdim} := 2k + 10$. The convergence histories of PCG (bJ10), HR-DEF-PCG (bJ10) and RR-LO-TR-DEF-PCG (bJ10) are presented in Fig. 7. The number of necessary solver iterations to reach a backward error smaller than 10^{-7} using PCG (AMG) and RR-LO-TR-DEF-PCG (bJ10, 20, 30) is presented on the left side of Fig. 8 for $s \leq 200$, whereas the right side of Fig. 8 presents the corresponding spectra of \mathbf{A}_s , $\dot{\mathbf{A}}_s$ and $\ddot{\mathbf{I}}_s \dot{\mathbf{A}}_s$ at $s = 100$.

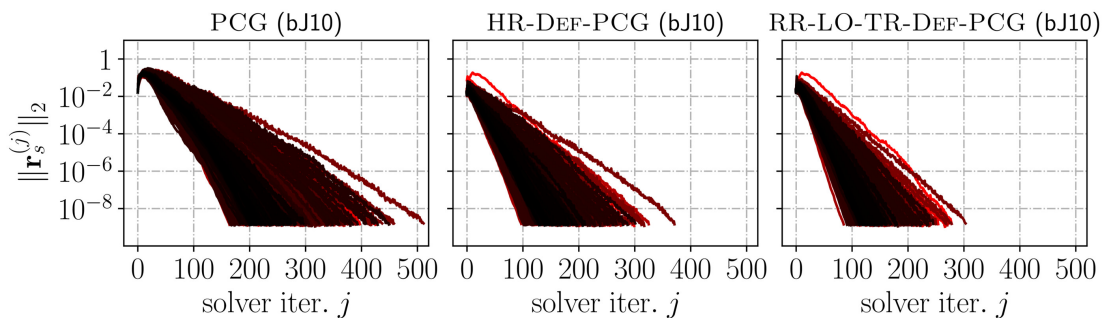


Figure 7: Norm of iterated residuals obtained by PCG (bJ10), HR-DEF-PCG (bJ10) and RR-LO-TR-PCG (bJ10) for a sequence of 1,000 linear systems with 4,000 DoFs. Coefficient field of Case 2. System index s gradually color coded from first ($s = 1$, ---) to last ($s = 1,000$, ---) in sequence.

Unlike the non-preconditioned results of Figs. 4–5, the results of Fig. 7 do not show a signif-

icant difference between the convergence behavior of the linear systems solved at the beginning and at the end of the sampled sequence. This could be due to the fact that the 2D coefficient fields of this example (Fig. 2) are less correlated than the ones used for the 1D example (Fig. 1), thus making the approximated eigenvectors of the previous operator less likely to result in an effective deflation of the current linear system. The preconditioned spectra of Fig. 8 are also denser than those of Fig. 6, due to the fact that the coefficient fields of this example are significantly smoother than those of Fig. 1. However, despite a smaller correlation between coefficient fields, and less separated eigenvalues after the application of the bJ preconditioners, the deflated spectra $\text{Sp}(\dot{\Pi}_s^T \dot{\mathbf{A}}_s)$ obtained by RR-LO-DEF-PCG (bJ) are nearly as condensed as the spectrum of $\dot{\mathbf{A}}_s$ obtained with the constant AMG preconditioner. Consequently, the iteration numbers (left side of Fig. 8) are also similar to those obtained by PCG (AMG)—for linear systems with 4,000 DoFs.

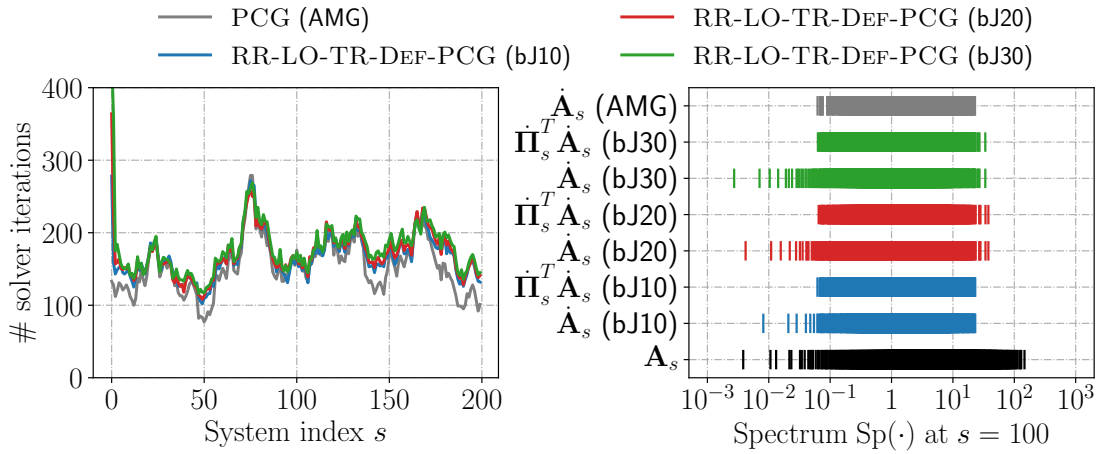


Figure 8: Left side: number of solver iterations needed to reach a backward error smaller than 10^{-7} using RR-LO-TR-DEF-PCG (bJ10, bJ20, bJ30) and PCG (AMG). Right side: original, preconditioned and deflated spectra of \mathbf{A}_s at $s = 100$. Linear systems with 4,000 DoFs and coefficient field of Case 2.

Finally, the same sequence of coefficient fields is considered with different spatial discretizations, along with the same f and boundary conditions, leading to linear systems of size 4,000, 16,000 and 64,000. These linear systems are solved by PCG (AMG), PCG (bJ10), HR-DEF-PCG (bJ10) and RR-LO-TR-DEF-PCG (bJ10) using $k := 20$ approximate eigenvectors with $\text{spdim} := 50$. The resulting numbers of solver iterations are presented in Fig. 9. We observe that increasing the number of DoFs results in a more significant degradation of the convergence for PCG (bJ10) than for PCG (AMG). For small numbers of DoFs, this difference is successfully compensated by the use of deflation, irrespective of whether the eigen-search space is refreshed or not. However, as the number of DoFs increases, deflation does not work as well, and the refreshing of the eigen-search space positively impacts more and more the convergence behavior of the deflated solver, as shown by the iteration number of RR-LO-TR-DEF-PCG compared to those of HR-DEF-PCG.

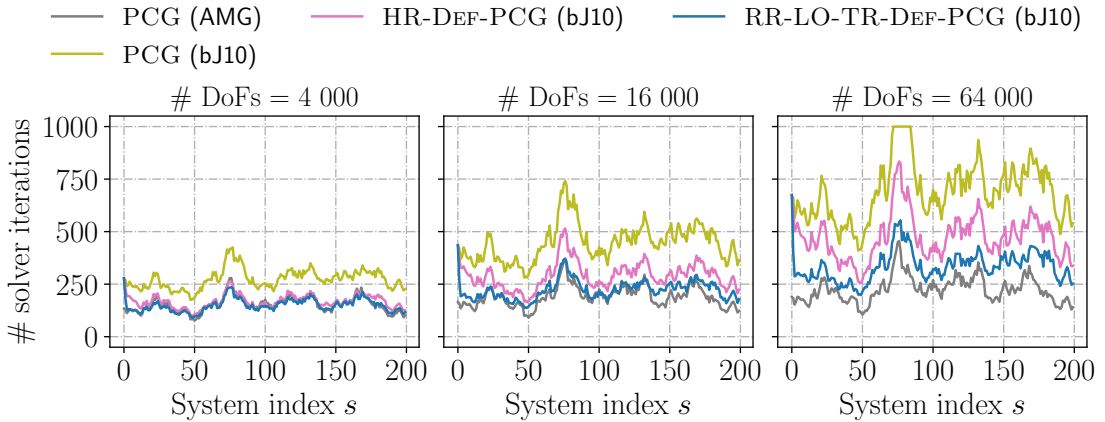


Figure 9: Number of solver iterations needed to reach a backward error smaller than 10^{-7} using HR-DEF-PCG (bJ10), RR-LO-TR-DEF-PCG (bJ10) and PCG (AMG, bJ10) for increasing numbers of DoFs. Coefficient field of Case 2.

5 Conclusion

Recycling Krylov subspace strategies were investigated as a means to accelerate the iterative solution of SPD linear systems in sequences of discretized elliptic PDEs with random coefficient fields sampled by MCMC. Every sampled linear system was deflated with LD eigenvector approximations of the previously sampled operator. These approximate eigenvectors were obtained using RR and HR projections with respect to eigen-search spaces spanned by sequences of matrix-vector products of the previous linear solve along with previous approximate eigenvectors. Different strategies were investigated to make use of all the matrix-vector products generated by the solver while keeping the dimension of the eigen-search space to a certain spd_{dim} in order to avoid problems of loss of orthogonality and memory consumption. These strategies were referred to as *refreshing* the eigen-search space so as to make clear that the linear solver is not restarted.

For the non-preconditioned cases, both DEF-CG and INIT-CG were used, and our experiments showed that, even if the coefficient fields are highly correlated, all the strategies based on INIT-CG eventually yield similar convergence behaviors to what is obtained with regular CG. For small numbers of DoFs, it was observed that the asymptotic convergence behavior of DEF-CG does not depend on the refreshing strategy of the eigen-search space. However, using eigCG [20] while explicitly orthogonalizing the residual against $\mathcal{R}(\mathbf{W}_s)$ at each solver iteration (method referred to as RR-LO-TR-DEF-CG), albeit incurring an additional cost of $O(kn)$ per iteration, allows for a significantly faster transition to better convergence behaviors than all the other refreshing strategies.

Two types of constant preconditioners were considered based on the operator $\hat{\mathbf{A}}$ constructed with the median realization: (i) a single V-cycle of an AMG solver, and (ii) bJ preconditioners with non-overlapping blocks. While the constant AMG preconditioner efficiently condenses the spectra of the sampled operators, numerical experiments (not reported here) showed that the behavior of PCG (AMG) is not significantly improved by deflation. However, the constant bJ preconditioners tend to segregate the spectrum into two parts, leaving a trail of as many eigenvalues as the number of bJ blocks spread throughout the lower end of the spectrum. These eigenvalues being well separated, the bJ preconditioners are well suited for deflation. Both 1D and 2D examples show that deflating with as many approximate eigenvectors as the number of bJ

blocks significantly improves the convergence behavior of DEF-PCG (bJ) compared to PCG (bJ). For small numbers of DoFs, the convergence behaviors of DEF-PCG (bJ) are comparable to those of PCG (AMG). However this is only the case for moderate numbers of blocks, say 30 or less. Indeed, by increasing the number of bJ blocks, one needs to increase the number k of approximated eigenvectors, and thus $spdim$, which can lead to undefined reduced (generalized) eigenvalue problems and larger memory requirements. As the number of DoFs increases, DEF-PCG (bJ) does not work as well as PCG (AMG), and the refreshing strategy of the eigen-search space starts to matter. It is then recommended to use eigCG [20] with explicit orthogonalization of the iterated residual (i.e., RR-LO-TR-DEF-CG) over the recycling strategy of [19] adapted for multiple operators (i.e., HR-DEF-PCG).

As long as constant preconditioners are used, deflation does not seem to allow for significantly better convergence behaviors than simply using PCG (AMG), particularly for large numbers of DoFs. However, it should be noted that other classes of elliptic PDEs such as, e.g., the Helmholtz equation, for which there is no obvious choice of a robust and scalable preconditioner comparable to AMG for the diffusion equation considered here, may still benefit from deflation techniques provided that they manage to compensate for a less efficient preconditioning strategy. The application of recycling Krylov subspace strategies to such equations may be investigated in future work. Furthermore, deflation may also prove useful in legacy software and/or when problem-specific, highly tuned preconditioners are used. Besides deflation, a potentially more promising option to improve the iterative solve of SPD linear systems arising from the discretization of stochastic elliptic PDEs is to periodically re-define the preconditioner for small groups (clusters) or subsequences of realizations. This is currently the focus of ongoing work and will be reported on in future communications.

References

- [1] A. CHAPMAN AND Y. SAAD, *Deflated and augmented Krylov subspace techniques*, Numerical linear algebra with applications, 4 (1997), pp. 43–66.
- [2] J. ERHEL AND F. GUYOMARC'H, *An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1279–1299.
- [3] L. GIRAUD, D. RUIZ, AND A. TOUHAMI, *A comparative study of iterative solvers exploiting spectral information for SPD systems*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1760–1786.
- [4] A. GREENBAUM, *Iterative methods for solving linear systems*, SIAM, 1997.
- [5] M. KAC AND A. SIEGERT, *An explicit representation of a stationary gaussian process*, Ann. Math. Stat., 18 (1947), pp. 438–442.
- [6] K. KAHL AND H. RITTICH, *The deflated conjugate gradient method: Convergence, perturbation and accuracy*, Linear Algebra and its Applications, 515 (2017), pp. 111–129.
- [7] R. B. MORGAN, *Computing interior eigenvalues of large matrices*, Linear Algebra and its Applications, 154-156 (1991), pp. 289 – 309.
- [8] R. B. MORGAN AND M. ZENG, *A harmonic restarted Arnoldi algorithm for calculating eigenvalues and determining multiplicity*, Linear algebra and its applications, 415 (2006), pp. 96–113.

-
- [9] L. N. OLSON AND J. B. SCHRODER, *PyAMG: Algebraic multigrid solvers in Python v4.0*, 2018, <https://github.com/pyamg/pyamg>. Release 4.0.
- [10] C. C. PAIGE, *The computation of eigenvalues and eigenvectors of very large sparse matrices.*, PhD thesis, University of London, 1971.
- [11] C. C. PAIGE, B. N. PARLETT, AND H. A. VAN DER VORST, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical linear algebra with applications, 2 (1995), pp. 115–133.
- [12] M. L. PARKS, E. DE STURLER, G. MACKEY, D. D. JOHNSON, AND S. MAITI, *Recycling Krylov subspaces for sequences of linear systems*, SIAM Journal on Scientific Computing, 28 (2006), pp. 1651–1674.
- [13] B. N. PARLETT, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, Linear algebra and its applications, 29 (1980), pp. 323–346.
- [14] B. N. PARLETT, *The symmetric eigenvalue problem*, vol. 20, SIAM, 1998.
- [15] Y. SAAD, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Mathematics of computation, 48 (1987), pp. 651–662.
- [16] Y. SAAD, *Analysis of augmented Krylov subspace methods*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 435–449.
- [17] Y. SAAD, *Iterative methods for sparse linear systems*, vol. 82, SIAM, 2003.
- [18] Y. SAAD, *Numerical methods for large eigenvalue problems: revised edition*, vol. 66, SIAM, 2011.
- [19] Y. SAAD, M. YEUNG, J. ERHEL, AND F. GUYOMARC'H, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (1999), pp. 1909–1926.
- [20] A. STATHOPOULOS AND K. ORGINOS, *Computing and deflating eigenvalues while solving multiple right-hand side linear systems with an application to quantum chromodynamics*, SIAM Journal on Scientific Computing, 32 (2010), pp. 439–462.
- [21] H. A. VAN DER VORST, *An iterative solution method for solving $f(A)x=b$, using Krylov subspace information obtained for the symmetric positive definite matrix A* , Journal of Computational and Applied Mathematics, 18 (1987), pp. 249–263.
- [22] K. WU AND H. SIMON, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, 22 (2000), pp. 602–616.



**RESEARCH CENTRE
BORDEAUX – SUD-OUEST**

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399