University of Groningen

A spectrally preconditioned and initially deflated variant of the restarted block GMRES method for solving multiple right-hand sides linear systems

Sun, Dong-Lin; Carpentieri, Bruno; Huang, Ting-Zhu; Jing, Yan-Fei

Published in:
International Journal of Mechanical Sciences

DOI:
[10.1016/j.ijmecsci.2018.06.033](https://doi.org/10.1016/j.ijmecsci.2018.06.033)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Sun, D-L., Carpentieri, B., Huang, T-Z., & Jing, Y-F. (2018). A spectrally preconditioned and initially deflated variant of the restarted block GMRES method for solving multiple right-hand sides linear systems. *International Journal of Mechanical Sciences*, 144, 775-787. <https://doi.org/10.1016/j.ijmecsci.2018.06.033>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

A spectrally preconditioned and initially deflated variant of the restarted block GMRES method for solving multiple right-hand sides linear systems

Dong-Lin Sun^{a,b}, Bruno Carpentieri^{c,*}, Ting-Zhu Huang^b, Yan-Fei Jing^b

^a Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Nijenborgh 9, Groningen 9747 AG, The Netherlands

^b School of Mathematical Sciences/Institute of Computational Science, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

^c Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3 39100, Italy

ARTICLE INFO

Keywords:

Multiple right-hand sides linear systems
Block Krylov subspace methods
Two-level preconditioners
Deflation
Boundary integral equations
Quantum chromodynamics

ABSTRACT

The solution of large linear systems with multiple right-hand sides given simultaneously is required in many large-scale scientific and engineering applications modelled by either partial differential or boundary integral equations. Block Krylov subspace methods are attractive to use for this problem class as they can overcome the memory bottleneck of direct methods and they perform block matrix-vector products, achieving high computational efficiency on modern cache-based computer architectures. In this paper we introduce variants of the block GMRES method that combine initial deflation and eigenvalue recycling strategies to remedy some typical convergence problems of block Krylov solvers. The new class of block iterative solvers has the ability to handle the approximate linear dependence of the block of right-hand sides and exploits approximate invariant subspaces recycled over the iterations to mitigate the bad effects that small eigenvalues can have on the convergence, by adapting an existing preconditioner. We illustrate the numerical behavior of the spectrally updated and initially deflated block GMRES method on a set of linear systems arising from the discretization of the Dirac equation and of boundary integral equations in electromagnetics scattering.

1. Introduction

In this paper we consider the simultaneous solution of multiple right-hand sides linear systems of the form

$$AX = B, \quad (1)$$

where $A \in \mathbb{C}^{n \times n}$ is a large square nonsingular matrix of dimension n , $B = [b_1, b_2, \dots, b_p] \in \mathbb{C}^{n \times p}$ is a full rank matrix of the p right-hand side vectors b_i , $i = 1, 2, \dots, p$ and $X \in \mathbb{C}^{n \times p}$ is the unknown solution matrix. An efficient solution of systems (1) is an important computational task required in many large-scale scientific and engineering simulations, e.g., in the radar cross section (RCS) calculation of realistic targets [12], in lattice quantum chromodynamics (QCD) applications [32], in computational acoustics [19] and in circuit analysis using model reduction techniques [18]. In some of these problems the coefficient matrix A is large and sparse, for example for lattice QCD discretizations of the Dirac operator. On the other hand, boundary element discretizations of integral equations in computational electromagnetics, medical imaging and computational acoustic problems lead to large and fully populated matrices. Although some efficient variants of Gaussian elimination have been proposed for solving blocks of right-hand sides simultaneously, see

e.g. [2,14], the huge memory requirements of direct methods often remain a bottleneck in large-scale simulations. Krylov subspace methods are matrix-free, and thus they can solve the memory burdens of direct methods. They can be attractive alternatives to use for solving sequences of linear systems of the form (1) on parallel computers especially when the coefficient matrix A is not explicitly available.

Block variants of Krylov subspace methods [1,7,21] can offer noticeable advantages over conventional Krylov algorithms for the solution of multiple right-hand sides linear systems. They can solve the whole sequence at once using much larger search spaces and they perform block matrix-vector products maximizing computational efficiency on modern cache-based computer architectures. The block GMRES method (BGMRES) [31] is one of the most popular block Krylov subspace solvers for this problem class. Let $X_0 \approx X$ be the initial block guess in (1) and $R_0 = B - AX_0$ the corresponding initial block residual. The BGMRES algorithm computes the minimal residual norm over the block Krylov subspace

$$\mathbb{K}_m(A, R_0) = \text{blockspan}\{R_0, AR_0, \dots, A^{m-1}R_0\}. \quad (2)$$

As in finite arithmetic the spanning vectors $\{R_0, AR_0, \dots, A^{m-1}R_0\}$ tend to be numerically unstable (i.e. they become approximately linearly dependent), in the BGMRES method an orthogonal basis for the space

* Corresponding author.

E-mail addresses: donglin1220@126.com (D.-L. Sun), bruno.carpentieri@unibz.it (B. Carpentieri), tingzhuang@126.com (T.-Z. Huang), 00jyfvictory@163.com (Y.-F. Jing).

<https://doi.org/10.1016/j.ijmecsci.2018.06.033>

Received 25 January 2018; Received in revised form 1 June 2018; Accepted 22 June 2018

Available online 23 June 2018

0020-7403/© 2018 Elsevier Ltd. All rights reserved.

$\mathbb{K}_m(A, R_0)$ is computed from the Modified Block Gram-Schmidt process, by running m steps of the block Arnoldi procedure presented in Algorithm 1. The process is started with a matrix V_1 computed from

Algorithm 1 Block Arnoldi procedure with blockwise modified Gram-Schmidt orthogonalization.

```

1: Choose a unitary matrix  $V_1$  of size  $n \times p$ .
2: for  $j = 1, 2, \dots, m$  do
3:   Compute  $W_j = AV_j$ 
4:   for  $i = 1, 2, \dots, j$  do
5:      $H_{i,j} = V_i^H W_j$ 
6:      $W_j = W_j - V_i H_{i,j}$ 
7:   end for
8:    $W_j = V_{j+1} H_{j+1,j}$  (reduced QR-decomposition of  $W_j$ )
9: end for
10: Define  $\mathcal{V}_{m+1} = [V_1, V_2, \dots, V_{m+1}]$ ,  $\tilde{H}_m = (H_{k,l})_{1 \leq k \leq m+1, 1 \leq l \leq m}$ 

```

the QR-decomposition of R_0 , that is $R_0 = V_1 R$ where matrix $V_1 \in \mathbb{C}^{n \times p}$ has orthonormal columns and $R \in \mathbb{C}^{p \times p}$ is a full rank upper triangular matrix. After m steps, the block Arnoldi algorithm yields the so-called block Arnoldi factorization

$$A\mathcal{V}_m = \mathcal{V}_{m+1}\tilde{H}_m, \quad (3)$$

where $\mathcal{V}_m = [V_1, V_2, \dots, V_{m+1}] \in \mathbb{C}^{n \times mp}$ is the computed block orthonormal sequence spanning $\mathbb{K}_m(A, R_0)$, with $\mathcal{V}_m^H \mathcal{V}_m = I_m$ and $\tilde{H}_m \in \mathbb{C}^{(m+1)p \times mp}$ of the form

$$\tilde{H}_m = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,m} \\ H_{2,1} & H_{2,2} & \dots & H_{2,m} \\ 0_{p \times p} & H_{3,2} & \dots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0_{p \times p} & 0_{p \times p} & 0_{p \times p} & H_{m+1,m} \end{bmatrix} = \begin{bmatrix} H_m \\ H_{m+1,m} E_m^H \end{bmatrix}. \quad (4)$$

The matrix $H_m \in \mathbb{C}^{mp \times mp}$ in (4) is a block Hessenberg matrix with $p \times p$ blocks $H_{i,j}$, while E_m is the matrix of the last p columns of I_{mp} .

The approximate BGMRES solution to systems (1) can be written as $X_m = X_0 + \mathcal{V}_m Y_m$, where Y_m minimizes the norm of the block residual matrix $R_m = R_0 - A\mathcal{V}_m Y_m$ over $\mathbb{K}_m(A, R_0)$. The minimization leads us to solving the small least-squares problem

$$Y_m = \operatorname{argmin}_Y \|ER - \tilde{H}_m Y\|_F, \quad (5)$$

with $R_0 = V_1 R = \mathcal{V}_{m+1} ER$ and E is the $(m+1)p \times p$ matrix whose upper $p \times p$ principal block is an identity matrix.

Although generally robust, the BGMRES algorithm can be memory demanding due to the heavy memory requirements of the block orthogonalization process. The problem can be remedied by restarting periodically the block Arnoldi procedure using the last computed approximation as m increases, at the price of slowing down the convergence to some extent. It has been shown that at each iteration of GMRES, basically the next smallest eigenvalue in magnitude is removed from the spectrum of the linear system. Therefore, the restarting procedure can destroy information on the very small eigenvalues of A and the super-linear convergence may be lost [35]. It is known that the convergence of Krylov methods is heavily reliant on the spectral properties of the coefficient matrix A . Krylov methods build a polynomial expansion of A that is equal to one at zero and has to approximate zero on the set of eigenvalues. This means that eigenvalues tightly clustered around a single point away from the origin are good, while clusters close to zero and widely spread eigenvalues, especially if they lie on both sides of the origin, are bad because a low-degree polynomial with value 1 at the origin cannot be small at a large number of such points [20].

Preconditioning can be used to improve the spectral properties of the linear systems (1) by transforming them into the equivalent form

$$MAX = MB, \quad (6)$$

(or $AMY = B$) where M is a nonsingular matrix close to A^{-1} in some sense, cheap to compute and to apply, and is such that the eigenvalues

of the preconditioned matrix MA (or AM for right preconditioning) are well clustered away from the origin. Consequently, solving (6) may be significantly faster than solving (1). Many of the preconditioners proposed in the literature succeed in clustering most of the eigenvalues of the preconditioned matrix MA far from the origin. However, they often tend to leave a few eigenvalues close to zero in the spectrum of MA . This may happen especially for indefinite problems when some of the eigenvalues, in their migration process towards the real axis under the action of a good preconditioner, may group close to the origin of the complex plane potentially degrading the convergence.

There are essentially two different approaches for exploiting information related to the smallest eigenvalues in the iterative solution. The first idea is to compute a few, k say, approximate eigenvectors of MA corresponding to the k smallest eigenvalues in magnitude, and augment the Krylov subspace with those directions. This approach is referred to as the *augmented subspace approach* (see [8,25–27,33,36]). The standard implementation of the restarted GMRES algorithm is based on the Arnoldi process, and spectral information of MA can be recovered during the iterations. The second idea exploits spectral information gathered during the Arnoldi process to determine an approximation of an invariant subspace of A associated with the eigenvalues nearest the origin, and uses this information to construct a preconditioner or to adapt an existing one [3,4,11,13,16,23].

Another practical difficulty in the implementation of block Krylov methods is to handle efficiently the approximate linear dependence of the block of right-hand sides. The problem has been addressed by other authors, see e.g. [1,6,7]. The explicit reduction technique described in [21,30] proposes to compute a singular value decomposition (SVD) of matrix B or of the initial block residual matrix R_0 before starting the iterative solution, and to filter the smallest singular values according to a certain deflation tolerance retaining only the most relevant contributions to the block Krylov subspace [5,7,21]. Another approach, known as Arnoldi deflation, removes the linearly dependent columns of the matrix W_j by a rank-revealing QR algorithm at step 8 of Algorithm 1, ensuring that V_{j+1} has full rank [21].

There is an inherent difficulty to combine the augmented subspace approach and the initial deflation strategy, as discussed in [33]. Since the residuals of the linear systems and the residuals of the Ritz vectors to be recycled at restart are not in the same subspace, the Arnoldi-like relation cannot be preserved at restart. In this paper we propose a spectrally preconditioned and initially deflated version of the BGMRES method (shortly, SPID-BGMRES) that attempts to overcome the two numerical problems described above. To accelerate the convergence of the standard restarted BGMRES method, the SPID-BGMRES method gathers spectral information during the block Arnoldi process to determine an approximation of an invariant subspace of MA associated with the eigenvalues nearest the origin, and uses this information to adapt an existing preconditioner M . The spectral preconditioning formulation is presented in Section 2. The eigenspace approximation and the recycling procedure are presented in Section 3. In Section 4, the spectrally preconditioned BGMRES method is combined with an initial deflation strategy to monitor the approximate linear dependence of the block of right-hand sides over the iterates. Numerical experiments illustrated in Sections 5 show that the SPID-BGMRES method can solve sequences of multiple right-hand sides linear systems arising in lattice QCD applications and from boundary integral equations discretization in electromagnetics scattering more efficiently than standard BGMRES. Some concluding remarks close the paper in Section 6. To the best of our knowledge, our method is the first block Krylov solver that combines initial deflation with eigenspace recycling.

Throughout the paper, the symbol $\|\cdot\|_q$ is used to denote the Euclidean norm when $q = 2$ and the Frobenius norm when $q = F$. The superscript H denotes the transpose conjugate of a vector or a matrix and $0_{i \times j} \in \mathbb{C}^{i \times j}$ is the zero rectangular matrix with i rows and j columns. In addition, MATLAB notation is adopted; for example, $R(i, j)$ denotes

the (i, j) th entry of matrix R , and $R(1:m, 1:j)$ refers to the submatrix consisting of the first m rows and the first j columns of R .

2. Spectral two-level preconditioning

In [11,13], the authors presented a spectral preconditioning formulation for solving general linear systems where invariant subspaces associated to the smallest eigenvalues of MA , precomputed prior to the iterative solution using the MATLAB function `eigs`, were deflated from the Krylov subspace resulting in faster convergence. In this paper, we pursue a different strategy. We recover approximate eigenspaces of MA adaptively from the block Arnoldi process at low computational cost, and we recycle those eigenspaces to adapt an existing preconditioner M . The recycling procedure is described in Section 3. In this section we present the spectral preconditioning formulation.

We denote by M the first-level preconditioner that is used to solve systems (1), meaning that we are solving the linear systems (6). Let $\{\lambda_1, \dots, \lambda_n\}$ be the set of eigenvalues of MA in any order. Let the columns of V_k be the basis of a right invariant subspace of MA of dimension k associated to the eigenvalues $\{\lambda_1, \dots, \lambda_k\}$, that is $MAV_k = V_k J_k$ where the eigenvalues of J_k are $\{\lambda_1, \dots, \lambda_k\}$. Finally, let $V_k^\perp$ be an orthogonal complement of V_k . Then we can write

$$MAV = V \begin{pmatrix} J_k & E \\ 0 & F \end{pmatrix},$$

where $V = [V_k, V_k^\perp]$, the eigenvalues of J_k are $\{\lambda_1, \dots, \lambda_k\}$ and those of F are $\{\lambda_{k+1}, \dots, \lambda_n\}$.

2.1. Spectral low-rank preconditioning update

In the first formulation, we update the preconditioner matrix in the form

$$M_{slru} = M + M_c, \quad (7)$$

where $M_c = V_k A_c^{-1} W^H$ is the coarse-grid correction, and W is chosen such that $A_c = W^H A V_k$ is nonsingular. Then we have $M_{slru} A V_k = V_k (J_k + I_k)$ and $M_{slru} A V_k^\perp = V_k E + V_k^\perp F + V_k P$ with $P \in \mathbb{R}^{k \times (n-k)}$. Altogether, we can write

$$M_{slru} A V = V \begin{pmatrix} J_k + I_k & * \\ 0 & F \end{pmatrix},$$

where the asterisk denotes a nonzero block whose actual expression is unimportant, proving the following result

Proposition 1. *The preconditioner M_{slru} is such that the preconditioned matrix $M_{slru} A$ is similar to a matrix whose eigenvalues are*

$$\begin{cases} \eta_i = \lambda_i & \text{if } i > k, \\ \eta_i = 1 + \lambda_i & \text{if } i \leq k. \end{cases}$$

The coarse-grid correction M_c results in shifting some of the eigenvalues from close to the origin of the complex plane to near to one for the new preconditioner. It should be noted that for unsymmetric linear systems in real arithmetic some of the eigenvectors can be complex. However, since if a complex eigenvector exists its conjugate is also an eigenvector, the problem can be remedied by replacing the eigenvectors with a real basis of the plane spanned by those two conjugate eigenvectors; that are, the two real vectors defined by the real part and the imaginary part of those vectors. In this case all the calculations would still be performed in real arithmetic.

2.2. Multiplicative two-grid spectral preconditioning

In the same spirit of standard multigrid cycles, the first-level preconditioner M can be used to define a smoother for a weighted stationary iterative method. Multigrid methods are amongst the fastest techniques for solving certain classes of partial differential equations on parallel

computers. A geometric two-grid algorithm typically consists of three stages: a pre-smoothing, a coarse grid correction and a post-smoothing step. In the pre-smoothing phase, a few iterations are applied on the fine grid to damp the high frequencies of the error. These components of the error are in the space spanned by the vectors associated with the largest eigenvalues of MA . The smooth components (i.e., the components associated with the smallest eigenvalues) are represented on a coarser grid by projecting the residual on the coarse grid where they can be captured by solving the coarse error equation. The error on the coarse mesh is then interpolated back into the fine grid for the solution update. Finally, in the post-smoothing phase, a few additional smoothing iterations are run to damp some high frequency components that might have been introduced back in the error by the interpolation process. If the new iterate is not accurate enough, the two-grid cycle is applied iteratively.

In standard multigrid cycles, the coarse space is not defined explicitly in terms of the Eigen components but instead by selecting a space that can capture those components effectively [17,22,34]. Therefore, we can define the coarse space as the span of the eigenvectors associated with the smallest eigenvalues of MA , the prolongation operator as $P = V_k$, the restriction operator R as $R = W^H$, and finally the coarse matrix problem by a Petrov–Galerkin formula $A_c = RAP$. The preconditioning algorithm takes as input the residual vector r to precondition. After, say μ_1 , pre-smoothing steps using the restriction operator $R = W^H$, we project the residual onto the coarse subspace and we solve the coarse space error equation defined by the matrix $A_c = W^H A V_k$. Finally, we project back the error to the original space using V_k and we smooth again the new approximation. The preconditioning operation is presented in Algorithm 2.

Algorithm 2 Multiplicative spectral preconditioner. Preconditioning operation: $z = M_{mul} r$.

```

1: Pre-smoothing: damp the high frequencies of the error
2:  $e^0 = 0$ 
3: for  $j = 1, \mu_1$  do
4:    $e^j = e^{j-1} + \omega M(r - A e^{j-1})$ 
5: end for
6: Coarse grid correction
7:  $e^{\mu_1} = e^{\mu_1} + V_k A_c^{-1} W^H (r - A e^{\mu_1})$ 
8: Post-smoothing: damp again the high frequencies of the error
9: for  $j = 1, \mu_2$  do
10:    $e^{\mu_1+j} = e^{\mu_1+j-1} + \omega M(r - A e^{\mu_1+j-1})$ 
11: end for
12:  $z = e^{\mu_1+\mu_2}$ 

```

For the sake of simplicity, we will simply denote this preconditioner as M_{mul} or $M_{mul}(A, M)$.

Using a standard result from multigrid theory (see, for instance [13]), the preconditioning operator M_{mul} has the following expression

$$M_{mul} = A^{-1} - (I - \omega MA)^{\mu_2} (I - M_c A) (I - \omega MA)^{\mu_1} A^{-1}, \quad (8)$$

where, as in Section 2.1, $M_c = V_k (W^H A V_k)^{-1} W^H$, $MAV_k = V_k J_k$ and $M_c A V_k^\perp = V_k^\perp$.

It is easy to show by induction on j that $(I - \omega MA)^j V_k = V_k (I - \omega J_k)^j$ and $(I - \omega MA)^j V_k^\perp = V_k^\perp (I - \omega F)^j + V_k P_j$, where $P_j \in \mathbb{R}^{k \times (n-k)}$. We also have $(I - M_c A) V_k^\perp = V_k^\perp - V_k P_c$ with $P_c \in \mathbb{R}^{k \times (n-k)}$. Therefore, the following relation can be established

$$M_{mul} A V = V \begin{pmatrix} I & * \\ 0 & I - (I - \omega F)^{\mu_1+\mu_2} \end{pmatrix}, \quad (9)$$

where again the asterisk denotes a nonzero block whose actual expression is unimportant. The result is summarized in the following proposition.

Proposition 2. *The preconditioner M_{mul} is such that the preconditioned matrix $M_{mul} A$ has eigenvalues*

$$\begin{cases} \eta_i = 1 & \text{if } i \leq k, \\ \eta_i = 1 - (1 - \omega\lambda_i)^{\mu_1 + \mu_2} & \text{if } i > k. \end{cases}$$

The result shows that, if M has the ability to cluster most of the eigenvalues close to one, possibly leaving relatively few outliers near zero, due to the choice of V_k the eigenvalues closest to the origin are shifted exactly to one and those that were grouped close to one get more clustered closer to this point after applying M_{mul} . Therefore, we can expect a significant reduction of the number of BGMRES iterations.

2.3. Additive two-grid spectral preconditioning

In the additive variant of the two-grid preconditioning cycle, the coarse grid correction and the smoothing operation are decoupled. The coarse grid correction computes only the residual components in the space spanned by the eigenvectors associated with the smallest eigenvalues of MA . The μ smoothing steps damp all the frequencies. At the end of the smoothing step, however, the preconditioned residual is filtered so that only the components in the complementary subspace are retained. These two contributions are summed together for the solution update. A simple additive two-level multigrid cycle is illustrated in Algorithm 3. Following [17,34], we define the filtering operator

Algorithm 3 Additive spectral preconditioner. Preconditioning operation: $z = M_{add} r$.

```

1: Damp all the frequency components of the error
2:  $e^0 = 0$ 
3: for  $j = 1, \mu$  do
4:    $e^j = e^{j-1} + \omega M(r - Ae^{j-1})$ 
5: end for
6: Filter the high frequency components of the error
7:  $c_1 = (I - V_k W^H) e^\mu$ 
8: Low frequency correction
9:  $c_2 = V_k A_c^{-1} W^H r$ 
10: Compute the preconditioned residual
11:  $z = c_1 + c_2$ 

```

$(I - V_k W^H)$ that is supposed to remove all the components in the V_k directions as $(I - V_k W^H)V_k = 0$.

The preconditioning operation described in Algorithm 3 can be written in the form $z = M_{add} r$, where the preconditioner matrix M_{add} has the following expression,

$$M_{add} = V_k A_c^{-1} W^H + (I - V_k W^H)(I - (I - \omega M A)^\mu) A^{-1}. \quad (10)$$

Using arguments similar to those for the proof of Proposition 2, it can be shown the following result [13].

Proposition 3. The preconditioner M_{add} is such that the preconditioned matrix $M_{add}A$ has eigenvalues

$$\begin{cases} \eta_i = 1 & \text{if } i \leq k, \\ \eta_i = 1 - (1 - \omega\lambda_i)^\mu & \text{if } i > k. \end{cases}$$

See [9,13] for discussions on the optimal choice of the parameter ω . The multiplicative and the additive two-grid spectral preconditioners have the ability to shift to one a selected set of eigenvalues and to cluster near one most of the others.

3. Subspace recycling in the SPID-BGMRES method

In SPID-BGMRES, the eigenvectors V_k 's defined in Section 2 are approximated from the block Arnoldi process at low computational cost. More precisely, we compute the so-called harmonic Ritz vectors. We recall the following definition of a harmonic Ritz pair from [28].

Definition 1. (Harmonic Ritz pair [28]). Consider a subspace $\mathcal{U}$ of $\mathbb{C}^n$. Given a matrix $B \in \mathbb{C}^{n \times n}$, $\lambda \in \mathbb{C}$, and $y \in \mathcal{U}$, (λ, y) is called a harmonic Ritz pair of B with respect to $\mathcal{U}$ if and only if

$$By - \lambda y \perp BU \quad (11)$$

or equivalently, using the canonical scalar product, if and only if

$$\forall \omega \in \text{Range}(BU^*) \quad \omega^H (By - \lambda y) = 0. \quad (12)$$

The vector y is called the harmonic Ritz vector associated with the harmonic Ritz value λ . $\square$

The following lemma shows how to compute the harmonic Ritz vectors from the block Arnoldi process.

Lemma 1. Let $\mathcal{U} = \text{span}\{\mathcal{V}_m\}$, where $\mathcal{V}_m$ is an orthonormal matrix built by BGMRES at the end of a cycle. The harmonic Ritz pairs (θ_i, g_i) associated with $\mathcal{V}_m$ satisfy the following property

$$\left(\tilde{H}_m \right)^H \left(\tilde{H}_m g_i - \theta_i \begin{bmatrix} g_i \\ 0_p \end{bmatrix} \right) = 0, \quad (13)$$

where $\mathcal{V}_m g_i$ are the harmonic Ritz vectors associated with the corresponding harmonic Ritz values θ_i .

Proof. See [33]. $\square$

Using (4), the above relation can be rewritten as

$$\left((H_m)^H H_m + E_m H_{m+1,m}^H H_{m+1,m} E_m^H \right) g_i = \theta_i (H_m)^H g_i, \quad (14)$$

and consequently the k targeted harmonic Ritz pairs can be computed during the BGMRES cycle. We denote by $V_k = \mathcal{V}_m G_k$, with $G_k = [g_1, \dots, g_k] \in \mathbb{C}^{m \times k}$, the k targeted harmonic Ritz vectors used to construct the spectral preconditioners, and we select $W = V_k$ in the definition of the coarse-grid correction M_c in the preconditioners M_{sbru} and in M_{mul} , and for the filtering operator in M_{add} . Although the computation of the coarse-grid spectral updates require extra costs for the eigen-

Algorithm 4 The SPID-BGMRES method.

```

1: for  $i = 1, \dots, m$  do
2:    $MR = M_i(B - AX_0)$ 
3:   Perform an inner iteration of the preconditioned BGMRES method (see Algorithm 5)
4:   Compute the  $k$  targeted eigenpairs of  $(\theta_i, g_i)$  (The  $\theta_i, (i = 1, \dots, k)$  are the harmonic Ritz vectors) of the matrix  $H_m^H H_m + E_m H_{m+1,m}^H H_{m+1,m} E_m^H$ . Then store the  $g_i$  into the matrix  $G_k$ 
5:   Compute  $V_k = \mathcal{V}_m G_k$  and  $A_c = (V_k)^H A V_k$ 
6:   Let  $M$  be the spectral two-level preconditioners function
7:   Define  $M_{i+1} = M(M_i, V_k, A_c)$ 
8: end for

```

computation to update the initial preconditioner M , these costs can be amortized by the faster convergence as demonstrated in our numerical results presented in Section 5.

4. Initial deflation strategy

In this section the spectrally preconditioned BGMRES method with eigenspace recycling presented earlier is combined with an initial deflation strategy to handle the case of the approximate linear dependence of the p right-hand sides. For brevity, we refer to the initially deflated variant of the conventional BGMRES method as BGMRES_D. Following an approach similar to the technique proposed in [7], the deflation process performs a block size reduction based on the SVD of the upper triangular factor arising from the QR-decomposition of the preconditioned block true residual MR_0 . That is,

$$MR_0 = QT,$$

$$T = S \Sigma V^H,$$

Algorithm 5 The preconditioned initially deflated BGMRESD method.

Input: $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times p}$, initial guesses $X_0 \in \mathbb{C}^{n \times p}$, a convergence threshold tol , a deflation threshold ϵ_d , the size of the restart m , the maximum number of iteration $itermax$, and the first-level preconditioner M_1

Output: $X \in \mathbb{C}^{n \times p}$

- 1: Compute the initial block residuals $R_0 \in \mathbb{C}^{n \times p}$ as $R_0 = B - AX_0$
- 2: **for** $i = 1, \dots, itermax$ **do**
- 3: Compute the reduced QR decomposition of $MR_0 = QT$ with $Q \in \mathbb{C}^{n \times p}$ and $T \in \mathbb{C}^{p \times p}$
- 4: Compute the SVD of T as $T = S\Sigma V^H$.
- 5: Select p_d singular values of T such that $\sigma_l(T) > \epsilon_d tol$ for all l such that $1 \leq l \leq p_d$
- 6: Define $V_1 \in \mathbb{C}^{n \times p_d}$ as $V_1 = QS(:, 1 : p_d)$
- 7: Let $B_k = \begin{bmatrix} I_{p_d} \\ 0_{kp_d \times p_d} \end{bmatrix}$, $1 \leq k \leq m$
- 8: **for** $j = 1, \dots, m$ **do**
- 9: Apply preconditioned block Arnoldi process. It generates $\mathcal{V}_{j+1}$, $\tilde{H}_j$ such that: $MA\mathcal{V}_j = \mathcal{V}_{j+1}\tilde{H}_j$ with $\mathcal{V}_{j+1}^H \mathcal{V}_{j+1} = I_{(j+1)p_d}$
- 10: Solve the minimization problem $Y_j = \underset{Y \in \mathbb{C}^{j p_d \times p_d}}{\operatorname{argmin}} \|B_j - \tilde{H}_j Y\|_F$
- 11: Compute $X_j = X_0 + \mathcal{V}_j Y_j \Sigma(1 : p_d, 1 : p_d) V(1 : p, 1 : p_d)^H$ and $R_j = B - AX_j$
- 12: if $\|R_j(:, s)\|_2 / \|B(:, s)\|_2 \leq tol$, $s = 1, \dots, p$, then stop
- 13: **end for**
- 14: $X_m = X_0 + \mathcal{V}_m Y_m \Sigma(1 : p_d, 1 : p_d) V(1 : p, 1 : p_d)^H$
- 15: $R_m = B - AX_m$
- 16: Check convergence, and restart if convergence is not achieved, set $R_0 = R_m$ and $X_0 = X_m$
- 17: Construct the spectral preconditioners M by Algorithm 4 and go to step 3
- 18: **end for**

where $Q \in \mathbb{C}^{n \times p}$ is a matrix with orthonormal columns, $T \in \mathbb{C}^{p \times p}$ is an upper triangular full-rank matrix, $S \in \mathbb{C}^{p \times p}$, $V \in \mathbb{C}^{p \times p}$ are unitary and $\Sigma \in \mathbb{C}^{p \times p}$ is diagonal. Small singular values would signal an approximate linear dependency of the block residuals. We filter the largest p_d singular values based on the following requirement

$$\sigma_l(T) > \epsilon_d tol, \quad l = 1, 2, \dots, p_d, \quad (15)$$

where $0 < \epsilon_d \leq 1$ and tol is the convergence threshold used in the stopping criterion of the BGMRESD method, arguing that the more accurate the iterative solution, the more right-hand sides we should solve for. By decomposing Σ as

$$\Sigma = \begin{bmatrix} \Sigma_+ & 0 \\ 0 & \Sigma_- \end{bmatrix},$$

with $\Sigma_+ \in \mathbb{C}^{p_d \times p_d}$ and $\Sigma_- \in \mathbb{C}^{(p-p_d) \times (p-p_d)}$, the block true residual can be written as follows

$$MR_0 = QS_+ \Sigma_+ V_+^H + QS_- \Sigma_- V_-^H, \quad (16)$$

with $S_+ \in \mathbb{C}^{p \times p_d}$, $S_- \in \mathbb{C}^{p \times (p-p_d)}$, $V_+ \in \mathbb{C}^{p \times p_d}$ and $V_- \in \mathbb{C}^{p \times (p-p_d)}$. Upon the block residuals deflation, only the linear systems corresponding to the p_d most linearly independent columns of MR_0 are solved simultaneously.

The preconditioned block Arnoldi procedure can be obtained by modifying Step 3 in Algorithm 1. It is started with the matrix $V_1 = QS_+$, and computes at step j the block basis $\mathcal{V}_{j+1} \in \mathbb{C}^{n \times (j+1)p_d}$ and the block Hessenberg matrix $\tilde{H}_j \in \mathbb{C}^{(j+1)p_d \times j p_d}$ that satisfy

$$MA\mathcal{V}_j = \mathcal{V}_{j+1} \tilde{H}_j. \quad (17)$$

At this stage, the approximate solution to system (6) is written as $X_m = X_0 + \mathcal{V}_m Y_m \Sigma_+ V_+^H$, where Y_m is the minimizer

$$\underset{Y}{\operatorname{argmin}} \|B - A(X_0 + \mathcal{V}_j Y \Sigma_+ V_+^H)\|_F \quad (18)$$

Fig. 1. Convergence history of the BGMRES, BGMRESD, and SPID-BGMRES methods with different preconditioners on Example Π_1 and $ILU(5 \cdot 10^{-2})$. We used $\mu_1 = 1$, $\mu_2 = 2$ in M_{mul} and $\mu = 3$ in M_{add} .

computed over the space $X_0 + \operatorname{range}(\mathcal{V}_j Y \Sigma_+ V_+^H)$ which is similar in [7]. The difference between the deflation strategy implemented in the SPID-BGMRES method and the one proposed by Calandra et.al. is that in this paper we compute the block true residual norm to detect the convergence rather than using the block quasi-residual, so that the stopping criterion is independent of the preconditioner. The complete pseudocode of the SPID-BGMRES method is presented in Algorithm 4, where the adaptive spectral preconditioners are given in implicit form.

5. Numerical experiments

We illustrate the numerical behaviour of the SPID-BGMRES method for solving multiple right-hand sides linear systems (1) arising in two different applications. In our experiments, the harmonic Ritz vectors were recovered directly from the block Arnoldi process. They were computed using the MATLAB function `eigs` that calls ARPACK [24]. The iterative solution was started from the initial guess $0 \in \mathbb{C}^{n \times p}$ and was terminated when

$$\frac{\|B_m(:, s) - AX(:, s)\|_2}{\|B(:, s)\|_2} \leq tol, \quad s = 1, \dots, p,$$

or the number of iterations exceeded the value $maxiter$. Note that this stopping criterion is independent of the preconditioner used. We explicitly computed the true unpreconditioned residual at each iteration even though this quantity might be available as a by-product of the Krylov solver. The experiments were run under Windows 10 and MATLAB version R2017b, on a PC with an Inter(R) Core(TM) i5-7200U CPU @2.50 GHz and 8GB of memory. All the timings in the tables are expressed in seconds.

5.1. Quantum chromodynamics problems

In this section we illustrate the convergence performance of the SPID-BGMRES method for solving a set of quantum chromodynamic matrices available from the University of Florida Sparse Matrix Library [15]. We chose a collection of four $49,152 \times 49,152$ complex matrices, i.e. `conf5_4-8x8-05`, `conf6_0-8x8-20`, `conf6_0-8x8-30`, `conf6_0-8x8-80` denoted in the tables as problems Π_i , $i = 1, \dots, 4$, respectively. Each matrix Π_i exists with a critical value κ_c with $\frac{1}{\kappa_c} < \frac{1}{\kappa} < \infty$ such that $A = I - \kappa \Pi_i$ is real-positive. We set the number of right-hand sides equal to $p = 10$, the maximum

Table 1

Number of iterations and solution time in seconds for the SPID–BGMRES method with the M_{sbru} preconditioner varying the dimension of the low-rank update on the four problems Π_i .

Problem	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
Π_1	iters	1351	1290	1233	1222	1183	1149	1111	1134	1033	1033	1111
	cpu	130.80	129.57	122.71	121.60	119.34	118.67	112.56	113.76	103.30	107.95	122.25
Π_2	iters	741	648	592	560	541	542	531	531	531	521	569
	cpu	74.73	71.43	62.38	57.02	55.07	57.74	54.08	54.36	54.21	53.79	59.03
Π_3	iters	759	694	561	561	601	611	551	572	531	551	541
	cpu	75.40	72.16	64.61	56.86	60.83	61.54	59.68	67.88	57.64	56.39	54.94
Π_4	iters	791	651	596	581	561	561	541	525	511	521	531
	cpu	75.84	63.79	58.25	57.20	58.09	57.36	54.86	52.11	51.06	52.01	53.47

Table 2

Number of iterations and solution time in seconds for the SPID–BGMRES method with the M_{mul} and the M_{add} preconditioners, varying the number of smoothing steps and the dimension of the low-rank update on Example Π_1 .

μ	Results	M_{mul}										
		Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_1 = 0, 1$	iters	1351	1285	1181	1166	1133	1050	1105	1029	1015	922	978
	cpu	131.99	138.15	131.31	129.21	128.78	122.56	127.49	120.01	116.58	107.85	114.56
$\mu_1 = 1, 0$	iters	1351	1241	1166	1187	1163	1149	1121	1091	1081	1071	1009
	cpu	128.87	134.26	124.65	132.20	128.21	131.47	125.80	122.39	123.42	121.72	117.40
$\mu_1 = 2, 0$	iters	1351	659	599	572	609	541	521	501	491	501	480
	cpu	131.02	78.49	71.36	68.17	72.87	67.66	63.25	61.46	60.42	63.31	60.41
$\mu_1 = 1, 1$	iters	1351	677	616	559	593	494	491	511	471	461	451
	cpu	133.15	83.65	75.87	67.90	71.95	60.60	60.72	64.78	59.53	58.12	59.35
$\mu_1 = 0, 2$	iters	1351	663	628	633	559	473	500	445	446	441	432
	cpu	128.32	78.93	74.77	76.07	66.86	56.73	61.32	53.71	53.65	53.41	52.22
$\mu_1 = 2, 1$	iters	1351	455	388	351	372	341	332	321	302	311	301
	cpu	128.61	59.70	50.59	46.54	51.14	44.63	43.92	42.16	40.00	41.49	40.19
$\mu_1 = 1, 2$	iters	1351	458	401	347	350	331	314	321	321	311	296
	cpu	130.45	59.88	53.11	45.39	46.23	43.20	41.17	42.15	42.12	41.46	39.00
$\mu_1 = 2, 2$	iters	1351	366	302	311	311	301	311	321	321	301	358
	cpu	129.87	51.97	43.08	44.18	45.09	43.20	44.49	45.81	45.80	43.59	51.98

μ	Results	M_{add}										
		Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	iters	1351	1285	1221	1188	1151	1175	1077	1111	1081	1046	1025
	cpu	139.81	160.69	135.52	134.62	132.21	134.61	125.65	128.66	127.75	128.09	118.52
$\mu = 2$	iters	1351	657	631	601	551	531	501	472	531	497	500
	cpu	129.13	77.57	74.38	72.24	66.99	63.72	62.94	57.06	64.48	61.11	61.04
$\mu = 3$	iters	1351	419	371	371	360	353	351	351	351	371	351
	cpu	128.08	53.99	47.82	48.09	49.26	46.40	46.33	46.68	46.70	49.50	46.96
$\mu = 4$	iters	1351	363	311	340	365	359	421	449	455	455	456
	cpu	128.16	50.96	43.66	49.91	51.65	51.60	60.38	64.03	76.62	76.94	85.75

number of iterations $maxiter$ equal to 5000, the convergence tolerance tol equal to 10^{-6} and the maximum dimension m of the Krylov search space equal to 50. In our runs, the first-level preconditioner M was an incomplete LU factorization $ILU(t)$ with a dropping strategy based on threshold, computed in MATLAB using the built-in function `ilut` and the value $t = 5 \cdot 10^{-2}$ for the threshold. The first-level preconditioner is required to cluster most of the eigenvalues close to one so that the invariant subspace associated with the smallest eigenvalues has a small dimension. Finally, the deflation threshold was set equal to $\epsilon_d = 0.1$ in the experiments.

In Table 1, we present the number of iterations (*iters*) and the elapsed times (*cpu*) required by the SPID–BGMRES method with the M_{sbru} preconditioner, varying the dimension of the low-rank update in the range between 1 to 10. Dimension 0 means that M is not updated, so that the preconditioner reduces to $ILU(t)$ and the SPID–BGMRES method reduces to the preconditioned BGMRES method. In our runs, the SPID–BGMRES method outperformed the preconditioned BGMRES method in terms of both *iters* and *cpu* time. Generally the larger the dimension of the coarse space used, the faster the convergence although the decrease was not always monotonic. In Tables 2–5 we report on the number of

iterations of the SPID–BGMRES method with the M_{mul} and M_{add} preconditioners varying the total number of smoothing steps from 1 to 4. For all the test examples, increasing the number of smoother iterations improved the convergence. The general trend was that the larger the size of coarse space, the fewer iterations were required by the Krylov solver.

Fig. 1 compares the convergence histories of the conventional BGMRES, of the initially deflated BGMRES and of the SPID–BGMRES methods on Example Π_1 . Different preconditioners were tested with the latter solver. The vertical axis displays the $\log_{10}$ of the maximum true residual norms, and the horizontal axis displays the number of iterations. The figure shows that the deflation strategy can enhance the robustness of the BGMRES and the SPID–BGMRES methods significantly when the block residuals are approximately linearly dependent. They converged when BGMRES stagnated. The spectral updates incorporated in the SPID–BGMRES further helped to converge faster. Moreover, the multiplicative and additive preconditioner convergence curves were similar. The results were obtained using $\mu_1 = 1$, $\mu_2 = 2$ in M_{mul} and $\mu = 3$ in M_{add} .

In order to further compare the different spectral preconditioning algorithms available in the SPID–BGMRES solver, we varied the restart parameter m equal to 50, 60, 70, 80. In Fig. 2 we display the number

Table 3

Number of iterations and solution time in seconds for the SPID-BGMRES method with the M_{mul} and the M_{add} preconditioners, varying the number of smoothing steps and the dimension of the low-rank update on Example Π_2 .

M_{mul}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_i = 0, 1$	iters	741	659	610	537	531	541	511	511	511	511	496
	cpu	74.38	71.68	68.72	58.51	57.70	59.52	56.47	56.22	56.32	56.52	56.64
$\mu_i = 1, 0$	iters	741	651	601	531	532	511	501	501	511	501	491
	cpu	80.12	74.10	64.88	56.83	57.31	55.76	54.61	54.88	55.79	56.72	55.95
$\mu_i = 2, 0$	iters	741	335	321	301	298	291	281	281	272	271	271
	cpu	72.80	39.69	38.14	36.16	35.35	35.15	33.68	33.41	32.37	32.36	32.36
$\mu_i = 1, 1$	iters	741	337	321	321	292	291	285	274	271	271	263
	cpu	71.89	42.43	38.33	38.39	34.90	34.97	35.04	33.12	32.84	32.92	32.27
$\mu_i = 0, 2$	iters	741	331	321	311	292	291	281	281	272	271	271
	cpu	72.53	39.30	39.73	38.38	34.68	34.62	33.36	33.48	32.56	32.54	32.61
$\mu_i = 2, 1$	iters	741	221	219	207	202	202	202	201	202	201	201
	cpu	71.69	28.36	28.27	26.90	26.39	26.42	27.61	26.83	26.64	26.65	26.71
$\mu_i = 1, 2$	iters	741	221	221	211	203	206	202	202	201	201	201
	cpu	72.47	28.37	28.32	27.19	26.49	26.83	26.48	26.48	26.88	26.48	26.50
$\mu_i = 2, 2$	iters	741	172	171	162	161	161	161	163	162	164	171
	cpu	75.31	23.48	23.42	22.30	29.24	22.75	22.28	22.55	22.44	22.75	23.63
M_{add}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	iters	741	651	610	531	521	560	531	532	521	531	511
	cpu	74.28	71.16	66.58	58.52	57.22	64.22	58.72	60.11	58.66	59.92	58.11
$\mu = 2$	iters	741	331	321	321	301	301	302	301	311	311	311
	cpu	93.32	48.77	47.78	48.48	46.52	46.07	46.32	45.44	46.90	47.25	47.49
$\mu = 3$	iters	741	231	221	241	241	246	241	238	241	241	259
	cpu	77.56	29.69	28.36	31.39	31.22	31.73	31.05	30.72	31.10	31.37	34.14
$\mu = 4$	iters	741	172	171	201	251	251	277	289	229	234	278
	cpu	71.91	23.10	22.97	27.97	37.70	36.19	38.70	40.94	32.62	32.80	39.77

Table 4

Number of iterations and solution time in seconds for the SPID-BGMRES method with the M_{mul} and the M_{add} preconditioners, varying the number of smoothing steps and the dimension of the low-rank update on Example Π_3 .

M_{mul}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_i = 0, 1$	iters	759	672	581	591	569	541	551	541	510	521	519
	cpu	74.99	72.66	62.94	67.38	61.94	59.10	60.72	59.31	56.16	57.65	57.30
$\mu_i = 1, 0$	iters	759	649	561	596	609	567	531	531	501	541	531
	cpu	92.42	75.72	63.65	63.94	65.13	60.71	56.94	57.31	54.25	58.44	59.71
$\mu_i = 2, 0$	iters	759	351	311	321	301	301	281	271	271	262	263
	cpu	72.46	41.47	36.58	37.70	35.86	36.29	33.82	32.04	31.95	31.19	31.28
$\mu_i = 1, 1$	iters	759	351	331	321	301	301	279	272	271	271	271
	cpu	73.94	42.08	38.99	37.99	36.20	36.17	33.65	32.23	32.19	32.67	32.48
$\mu_i = 0, 2$	iters	759	351	319	301	292	291	271	271	281	272	274
	cpu	72.54	43.85	37.52	35.87	34.61	34.41	32.08	32.16	33.38	32.34	32.85
$\mu_i = 2, 1$	iters	759	221	221	201	201	201	201	201	201	201	200
	cpu	72.94	28.16	28.20	27.44	26.44	26.22	26.17	26.24	26.51	26.48	25.69
$\mu_i = 1, 2$	iters	759	221	221	201	201	201	201	201	201	201	200
	cpu	73.74	28.42	28.42	26.21	26.19	26.30	26.30	26.42	26.77	26.40	25.67
$\mu_i = 2, 2$	iters	759	162	161	161	153	153	153	161	161	154	153
	cpu	72.55	21.92	21.79	21.95	20.98	21.01	21.00	21.96	22.11	21.24	21.64
M_{add}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	iters	759	679	571	561	560	575	576	531	565	511	551
	cpu	74.88	73.84	61.63	61.18	63.89	63.08	63.62	59.04	63.29	57.18	62.48
$\mu = 2$	iters	759	341	312	321	301	311	301	298	301	293	301
	cpu	72.56	39.80	36.75	37.82	36.01	37.61	37.01	37.28	36.36	35.41	36.77
$\mu = 3$	iters	759	222	211	225	221	224	241	237	232	241	276
	cpu	72.85	28.09	26.92	28.53	28.15	28.77	31.43	30.72	29.86	31.10	36.21
$\mu = 4$	iters	759	171	171	189	201	258	251	234	250	271	271
	cpu	76.30	23.64	23.23	25.55	28.07	36.11	35.91	32.45	34.93	38.08	38.95

Table 5

Number of iterations and solution time in seconds for the SPID-BGMRES method with the M_{mul} and the M_{add} preconditioners, varying the number of smoothing steps and the dimension of the low-rank update on Example Π_4 .

		M_{mul}										
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_i = 0, 1$	iters	791	667	601	571	550	541	521	521	521	491	501
	cpu	76.95	75.58	65.46	62.71	60.17	59.33	57.09	57.46	58.03	56.21	59.98
$\mu_i = 1, 0$	iters	791	699	601	571	561	521	511	502	498	501	512
	cpu	75.78	74.10	64.16	61.53	60.62	58.11	55.16	54.70	54.76	55.25	55.70
$\mu_i = 2, 0$	iters	791	321	295	301	292	282	282	286	281	281	281
	cpu	76.13	37.58	34.50	35.86	36.61	33.68	33.71	33.79	33.18	33.17	33.46
$\mu_i = 1, 1$	iters	791	311	300	301	293	291	291	291	301	291	281
	cpu	75.68	37.00	35.51	36.06	34.64	34.52	34.99	36.84	36.44	34.85	33.86
$\mu_i = 0, 2$	iters	791	311	301	301	301	301	291	281	281	271	271
	cpu	75.62	36.63	35.78	35.90	36.06	36.40	35.17	33.27	33.38	33.04	33.53
$\mu_i = 2, 1$	iters	791	221	221	203	211	201	201	201	201	201	201
	cpu	75.56	28.16	28.24	26.23	27.13	26.17	27.72	26.67	26.66	26.36	26.37
$\mu_i = 1, 2$	iters	791	213	211	211	211	203	201	211	201	201	201
	cpu	75.88	27.22	26.99	27.10	27.11	26.32	26.19	27.27	26.65	26.26	26.32
$\mu_i = 2, 2$	iters	791	162	162	162	162	163	162	162	161	161	161
	cpu	75.66	22.02	22.10	22.15	22.12	22.32	22.27	22.50	22.16	22.61	22.25
		M_{add}										
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	iters	791	652	611	581	551	532	521	545	521	512	511
	cpu	79.80	85.66	81.05	71.94	66.59	63.58	60.66	66.59	60.09	59.73	58.69
$\mu = 2$	iters	791	321	301	301	301	291	291	291	284	301	301
	cpu	76.47	37.89	37.04	37.88	36.86	35.32	39.48	37.12	35.16	36.91	37.07
$\mu = 3$	iters	791	214	211	221	211	221	222	221	232	251	285
	cpu	76.52	27.38	27.06	28.35	27.35	28.49	29.44	30.36	30.35	33.55	37.62
$\mu = 4$	iters	791	174	171	193	265	201	275	276	296	233	288
	cpu	78.20	23.76	23.74	29.11	38.41	28.43	38.78	38.96	41.82	32.76	41.01

Fig. 2. Sensitivity to the restart parameter of the BGMRESD method with different preconditioners on Example Π_1 (Left: M_{slru} , Middle: M_{mul} , Right: M_{add} . Here $\mu = \mu_1 + \mu_2 = 3$).

of iterations versus the number of shifted eigenvalues for solving Example Π_1 . The two-grid preconditioners M_{mul} and M_{add} exhibited similar performance at different restart parameters, while they outperformed the low-rank spectrally updated preconditioner M_{slru} .

5.2. Boundary element matrices

The second set of linear systems arises from the boundary integral discretization of scattering problems in computational electromagnetics. In variational form, the problem can be formulated as the following integral equation model:

find the surface current $\vec{j}$ such that for all tangential test functions $\vec{j}^t$, we have

$$\int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{j}(x) \cdot \vec{j}^t(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{j}(x) \cdot \operatorname{div}_{\Gamma} \vec{j}^t(y) \right) dx dy = \frac{i}{k Z_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{j}^t(x) dx. \quad (19)$$

Eqn. (19) is called the Electric Field Integral Equation (EFIE); we denote by $G(|y-x|) = \frac{e^{ik|y-x|}}{4\pi|y-x|}$ the Green's function, i is the imaginary unit $\sqrt{-1}$, Γ is the boundary of the object, $\vec{E}_{inc}(x)$ denotes the incident radiation, k is the wave number and $Z_0 = \sqrt{\mu_0/\epsilon_0}$ the characteristic impedance of vacuum (ϵ is the electric permittivity and μ the magnetic permeability). The EFIE formulation can be applied to arbitrary geometries including those with cavities, disconnected parts, breaks on the surface, hence its wide popularity in industry. Upon the Galerkin discretization over a mesh containing n edges, the surface current $\vec{j}$ is expanded into the set of so-called Rao-Wilton-Glisson basis functions $\{\vec{\varphi}_i\}_{1 \leq i \leq n}$ [29]. Then Eqn. (19) is applied to each basis function $\vec{\varphi}_i$. Finally, we are led to solving the following linear system

$$\sum_{1 \leq \ell \leq n} \lambda_{\ell} \left[\int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{\varphi}_{\ell}(x) \cdot \vec{\varphi}_j(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{\varphi}_{\ell}(x) \cdot \operatorname{div}_{\Gamma} \vec{\varphi}_j(y) \right) dx dy \right] = \frac{i}{k Z_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(x) dx, \quad (20)$$

(a) Cube (124 MHz, n=1800)

(b) Cylinder (112 MHz, n=1266)

(c) Satellite (57 MHz, n=1701)

(d) Sphere (223 MHz, n=2430)

Fig. 3. Meshes used in the boundary integral discretization problems. For each geometry, we report in parenthesis the frequency of the illuminating wave and the dimension of the pertinent linear system.

in the unknowns λ_ℓ for $1 \leq \ell \leq n$, where each λ_ℓ is associated with the vectorial flux across an edge in the mesh. The set of equations (20) can be recast in matrix form as

$$A\lambda = b, \quad (21)$$

where $A = [A_{j\ell}]$ and $b = [b_j]$ have elements

$$A_{j\ell} = \int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{\varphi}_\ell(x) \cdot \vec{\varphi}_j(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{\varphi}_\ell(x) \cdot \operatorname{div}_{\Gamma} \vec{\varphi}_j(y) \right) dx dy,$$

$$b_j = \frac{i}{kZ_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(y) dx.$$

The coefficient matrix A is dense complex symmetric. The right-hand side varies with the frequency and the direction of the illuminating wave.

We chose a collection of four boundary element matrices corresponding to the four geometries described in Fig. 3. As in Section 5.1, we set the number of right-hand sides equal to $p = 10$, the maximum number of iterations *maxiter* equal to 5000, the convergence tolerance *tol* to 10^{-6} , the maximum dimension m of the Krylov search space equal to 50, and the deflation threshold equal to $\epsilon_d = 0.1$. The first-level preconditioner M was a sparse approximation of A^{-1} . Sparse approximate inverse methods have shown to be robust preconditioners for this problem class, as they can capture the rapid decay of the Green's function G very effectively (see e.g. [10,12]). In our experiments, we used physical right-hand sides in Eqn. (21). We took as incident field a plane wave of general form in spherical coordinates

$$\vec{E}_{inc}(x, \varphi, p_\theta, p_\varphi) = (p_\theta) \hat{u}_\theta e^{ikx \cdot \hat{u}_r(\varphi)} + (p_\varphi) \hat{u}_\varphi e^{ikx \cdot \hat{u}_r(\varphi)},$$

where (p_θ, p_φ) are two complex numbers and $\hat{u}_r, \hat{u}_\theta, \hat{u}_\varphi$ are the unitary vectors

$$\hat{u}_r = \begin{pmatrix} \cos \varphi \cos \theta \\ \sin \varphi \cos \theta \\ \sin \theta \end{pmatrix}, \hat{u}_\theta = \begin{pmatrix} -\cos \varphi \sin \theta \\ -\sin \varphi \sin \theta \\ \cos \theta \end{pmatrix}, \hat{u}_\varphi = \begin{pmatrix} -\sin \varphi \cos \theta \\ -\cos \varphi \cos \theta \\ \sin \theta \end{pmatrix}.$$

Without loss of generality, we took $\theta = 0$ and φ variable from 0 to 2π , leading to the following expression for the incident field

$$\vec{E}_{inc}(x) = \vec{E}_{inc}(x, \varphi) = \hat{z} e^{ikx \cdot \hat{u}_r(\varphi)} = \hat{z} e^{ik(x_1 \cos \varphi + x_2 \sin \varphi)}.$$

Fig. 4 plots the convergence histories of the conventional BGMRES method versus the initially deflated BGMRES method and the SPID-BGMRES method on the four selected boundary element problems. Different preconditioners were tested with the latter solver. The benefits of the initial deflation strategy are evident. Eigenvector recycling can further help to reduce the number of iterations. On the Satellite problem, a complete RCS calculation with 360 right-hand sides took 451 (72s) BGMRES iterations without coarse-grid correction and 202 (36s) iterations using 10 spectral corrections. By its nature, the spectral update is more effective if the first level preconditioner M leaves a small cluster of eigenvalues close to zero in the spectrum of MA . The sparse approximate inverse that we used did already a good job in grouping most of the eigenvalues close to point one as illustrated in Fig. 5. Therefore, in the convergence plots shown in Fig. 4 the improvement due to the coarse-grid correction is not dramatic, although still favourable. In Fig. 6 we propose an experiment where we used a less accurate first level preconditioner M . The spectrum of MA depicted in Fig. 7 shows a larger cluster of eigenvalues close to zero. On this example, using only 10 spectral updates enabled us to converge in a Krylov search space of dimension $m = 100$, whereas both the BGMRES and the BGMRES method

Fig. 4. Convergence histories of the BGMRES, BGMRES and SPID-BGMRES iterative solutions on selected boundary element matrices. We used $\mu_1 = 2$, $\mu_2 = 2$ in M_{mul} and $\mu = 4$ in M_{add} .

Fig. 5. Spectra of the preconditioned matrix MA for the four boundary element matrices.

Table 6

Number of iterations and solution time in seconds for the SPID-BGMRES method on the Sphere problem, varying the number of smoothing steps and the dimension of the low-rank update. The eigenvectors V_k 's are computed in pre-processing by running the eigensolver ARPACK on the matrix MA .

		M_{slru}										
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
-	<i>iters</i>	194	179	177	155	155	155	155	153	149	149	137
	<i>cpu</i>	6.07	5.60	5.62	5.56	5.06	5.08	4.98	4.98	4.90	4.8	4.33
		M_{mul}										
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_i = 0, 1$	<i>iters</i>	194	179	177	155	155	155	154	151	149	149	137
	<i>cpu</i>	6.79	8.60	8.05	6.32	6.32	6.29	6.23	6.13	5.94	5.95	5.45
$\mu_i = 1, 0$	<i>iters</i>	194	179	177	155	155	155	154	151	149	149	137
	<i>cpu</i>	6.18	7.30	7.34	6.74	6.32	6.27	6.40	6.18	6.03	6.26	5.53
$\mu_i = 2, 0$	<i>iters</i>	194	109	103	103	101	101	101	101	100	99	96
	<i>cpu</i>	6.13	4.81	4.52	4.56	4.44	4.76	4.41	4.58	4.30	4.30	4.05
$\mu_i = 1, 1$	<i>iters</i>	194	109	103	103	101	101	101	101	100	99	96
	<i>cpu</i>	6.08	4.82	4.54	4.56	4.42	4.44	4.71	4.43	4.31	4.26	4.08
$\mu_i = 0, 2$	<i>iters</i>	194	109	103	103	101	101	101	101	100	99	96
	<i>cpu</i>	7.52	5.00	4.55	4.53	4.49	4.46	4.42	4.44	4.26	4.29	4.08
$\mu_i = 2, 1$	<i>iters</i>	194	101	100	96	95	92	91	91	91	90	87
	<i>cpu</i>	6.15	4.96	4.69	4.48	4.41	4.18	4.21	4.13	4.16	4.12	3.90
$\mu_i = 1, 2$	<i>iters</i>	194	101	100	96	95	92	91	91	91	90	87
	<i>cpu</i>	6.06	4.90	4.70	4.48	4.43	4.19	4.17	4.27	4.20	4.13	3.92
$\mu_i = 2, 2$	<i>iters</i>	194	94	92	90	89	86	86	86	85	85	82
	<i>cpu</i>	6.07	4.77	4.61	4.49	4.38	4.19	4.34	4.57	4.41	4.18	4.01
		M_{add}										
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	<i>iters</i>	194	179	177	155	155	155	154	151	149	149	137
	<i>cpu</i>	6.02	6.61	6.22	5.47	5.45	5.39	5.36	5.31	5.29	5.12	4.72
$\mu = 2$	<i>iters</i>	194	109	103	103	101	101	101	101	100	100	96
	<i>cpu</i>	6.12	4.29	4.06	4.07	3.99	3.98	3.98	4.00	3.93	3.86	3.67
$\mu = 3$	<i>iters</i>	194	101	101	97	97	96	96	96	94	91	91
	<i>cpu</i>	6.05	4.44	4.44	4.20	4.16	4.09	4.04	4.11	3.94	3.82	3.85
$\mu = 4$	<i>iters</i>	194	94	95	95	94	93	91	91	91	90	90
	<i>cpu</i>	6.16	4.34	4.60	4.74	4.44	5.39	4.34	4.17	4.20	4.13	4.16

ods diverged in the same space (see Fig. 6). By using 20 spectral updates, the number of iterations halved.

Obviously, the accuracy of the eigenvectors used can affect the robustness of the SPID-BGMRES method. Some results are reported in Tables 6–7. In Table 6 the eigenvectors V_k 's are recycled from the block Arnoldi process as described in Section 3, while in Table 7 they are computed in pre-processing by running the eigensolver ARPACK on the

matrix MA . In the latter case the convergence is monotonic with the coarse space, and both number of iterations and solution times are (often) smaller. Monitoring the quality of the spectral approximation and filtering the good eigenpairs is an important performance issue of our method that deserves further attention. We will analyse the problem in a future study.

Fig. 6. Convergence histories of iterative solutions on the Satellite problem using a low-accurate preconditioner M and a coarse-grid correction of size 10 on the left graph and 20 on the right graph. We used $\mu_1 = 2, \mu_2 = 2$ in M_{mul} , $\mu = 4$ in M_{add} and $restart = 100$.

Table 7

Number of iterations and solution time in seconds for the SPID-BGMRES method on the Sphere problem, varying the number of smoothing steps and the dimension of the low-rank update. The eigenvectors V_k 's are recycled from the block Arnoldi process as described in Section 3.

M_{slru}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
-	<i>iters</i>	194	181	177	162	163	163	171	162	163	157	159
	<i>cpu</i>	6.33	6.09	5.99	5.57	5.57	5.55	5.84	5.58	5.56	5.41	5.53
M_{mul}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu_i = 0, 1$	<i>iters</i>	194	183	177	161	161	159	163	163	158	155	153
	<i>cpu</i>	6.38	8.00	7.51	6.89	6.85	6.78	7.03	7.14	6.76	6.66	6.53
$\mu_i = 1, 0$	<i>iters</i>	194	181	177	161	161	162	163	163	159	158	155
	<i>cpu</i>	6.10	7.64	7.52	7.10	7.14	7.21	7.27	7.76	7.09	7.13	6.93
$\mu_i = 2, 0$	<i>iters</i>	194	107	108	109	111	110	109	109	107	103	103
	<i>cpu</i>	7.53	5.21	5.21	5.33	5.42	5.34	5.28	5.39	5.18	4.92	4.98
$\mu_i = 1, 1$	<i>iters</i>	194	107	109	109	109	109	107	109	107	102	101
	<i>cpu</i>	6.36	5.12	5.22	5.22	6.72	5.30	5.27	5.45	5.02	4.78	4.78
$\mu_i = 0, 2$	<i>iters</i>	194	107	109	105	112	107	105	111	105	105	101
	<i>cpu</i>	6.44	4.95	5.15	5.08	5.24	5.32	5.04	5.27	4.99	4.98	4.87
$\mu_i = 2, 1$	<i>iters</i>	194	100	101	101	101	101	101	101	100	99	97
	<i>cpu</i>	6.30	5.06	5.30	5.32	5.28	5.34	5.35	5.32	5.14	5.05	4.94
$\mu_i = 1, 2$	<i>iters</i>	194	101	98	100	101	101	101	101	99	97	96
	<i>cpu</i>	6.39	5.29	4.95	5.13	5.38	5.35	5.36	5.36	5.10	4.97	5.13
$\mu_i = 2, 2$	<i>iters</i>	194	93	93	93	93	93	95	94	92	90	90
	<i>cpu</i>	6.43	5.04	5.10	5.08	5.08	5.08	5.20	5.22	5.00	4.80	4.85
M_{add}												
μ	Results	Dimension of the coarse space										
		0	1	2	3	4	5	6	7	8	9	10
$\mu = 1$	<i>iters</i>	194	181	178	161	161	166	163	163	161	161	163
	<i>cpu</i>	6.40	7.26	6.84	6.34	6.10	6.27	6.11	6.07	6.00	6.02	6.07
$\mu = 2$	<i>iters</i>	194	107	109	114	112	115	116	118	122	113	110
	<i>cpu</i>	6.18	4.52	4.57	4.76	4.73	4.85	4.93	5.17	5.11	4.78	4.62
$\mu = 3$	<i>iters</i>	194	101	101	101	101	101	101	103	104	104	101
	<i>cpu</i>	6.14	4.68	4.78	4.77	4.75	4.86	4.81	4.89	4.89	4.95	4.77
$\mu = 4$	<i>iters</i>	194	95	95	96	96	96	97	101	101	101	101
	<i>cpu</i>	6.14	4.72	4.73	4.77	4.84	4.79	4.86	5.28	5.28	5.40	5.32

Fig. 7. Eigenvalues distribution of the preconditioned matrix MA for the experiments on the Satellite problem illustrated in Fig. 6.

6. Conclusions

In this paper, novel variants of the BGMRES method were presented for the simultaneous solution of sequences of linear systems with multiple right-hand sides that arise from the discretization of either partial differential equations or integral equations. The novelty of the proposed family of algorithms is to combine effectively eigenspace recycling and initial deflation in the same Krylov subspace formulation. Eigenvalue recycling enables us to compute an approximation of an invariant subspace of the preconditioned matrix MA associated with the eigenvalues nearest the origin and use it to improve the spectral properties of the linear system by adapting an existing preconditioner. The spectral preconditioning update results in shifting the small eigenvalues from close to the origin to near to one for the new preconditioner, which can be very beneficial to reduce the number of Krylov iterations. On the other hand, initial deflation monitors the possible convergence of a linear combination of the linear systems due to the approximate linear dependence of the block of right-hand sides. Numerical experiments show that the proposed spectrally preconditioned and initially deflated BGMRES algorithms can outperform the conventional BGMRES method. To the best of our knowledge, our method is the first block Krylov solver that can combine initial deflation with eigenspace recycling. We believe that it can be useful for solving problems in many application areas, such as lattice quantum chromodynamics (QCD), electromagnetics and Acoustics applications.

Acknowledgments

This research is supported by NSFC (61772003). The first author is also funded by the University of Groningen Ubbo Emmius scholarship. We would like to thank the anonymous referees for their valuable remarks and comments that greatly improved the quality of the original manuscript.

References

- [1] Agullo E, Giraud L, Jing YF. Block GMRES method with inexact breakdowns and deflated restarting. *SIAM J Matrix Anal Appl* 2014;35(4):1625–51.
- [2] Alléon G, Amram S, Durante N, Homs P, Pogarielloff D, Farhat C. Massively parallel processing boosts the solution of industrial electromagnetic problems: high performance out-of-core solution of complex dense systems. In: Heath M, Torczon V, Aftabk G, Björstad PE, Karp AH, Koebel CH, Kumar V, Lucas RF, Watson LT, Womble DE, editors. Proceedings of the eighth SIAM conference on parallel computing. Philadelphia: SIAM Book; 1997. Conference held in Minneapolis, Minnesota, USA.
- [3] Baglama J, Calvetti D, Golub GH, Reichel L. Adaptively preconditioned GMRES algorithms. *SIAM J Sci Comput* 1999;20(1):243–69.
- [4] Bergamaschi L, Faccia E, Martínez A, Putti M. Spectral preconditioners for the efficient numerical solution of a continuous branched transport model. *J Comput Appl Math* 2018. Published online on January 31, 2018.
- [5] Calandra H, Gratton S, Lago R, Pinel X, Vasseur X. Two-level preconditioned Krylov subspace methods for the solution of three-dimensional heterogeneous Helmholtz problems in seismics. *Numer Anal Appl* 2012a;5(2):175–81.
- [6] Calandra H, Gratton S, Lago R, Vasseur X, Carvalho LM. A modified block flexible GMRES method with deflation at each iteration for the solution of non-hermitian linear systems with multiple right-hand sides. *SIAM J Sci Comput* 2013;35(5):S345–67.
- [7] Calandra H, Gratton S, Langou J, Pinel X, Vasseur X. Flexible variants of block restarted GMRES methods with application to geophysics. *SIAM J Sci Comput* 2012b;34(2):A714–36.
- [8] Calvez CL, Molina B. Implicitly restarted and deflated GMRES. *Numer Algorithms* 1999;21:261–85.
- [9] Carpentieri B. A matrix-free two-grid preconditioner for boundary integral equations in electromagnetism. *Computing* 2006;77(3):275–96.
- [10] Carpentieri B, Duff IS, Giraud L. Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism. *Numer Linear Algebra Appl* 2000;7(7–8):667–85.
- [11] Carpentieri B, Duff IS, Giraud L. A class of spectral two-level preconditioners. *SIAM J Sci Comput* 2003;25(2):749–65.
- [12] Carpentieri B, Duff IS, Giraud L, Sylvand G. Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations. *SIAM J Sci Comput* 2005;27(3):774–92.
- [13] Carpentieri B, Giraud L, Gratton S. Additive and multiplicative two-level spectral preconditioning for general linear systems. *SIAM J Sci Comput* 2007;29(4):1593–612.
- [14] Chew WC, Wang YM. A recursive T-matrix approach for the solution of electromagnetic scattering by many spheres. *IEEE Trans Antennas Propag* 1993;41(12):1633–9.
- [15] Davis TA, Hu Y. The University of Florida sparse matrix collection. *NA Digest* 1994;92.
- [16] Erhel J, Burrage K, Pohl B. Restarted GMRES preconditioned by deflation. *J Comput Appl Math* 1996;69:303–18.
- [17] Fournier L, Lanteri S. Multiplicative and additive parallel multigrid algorithms for the acceleration of compressible flow computations on unstructured meshes. *Appl Numer Math* 2001;36:401–26.
- [18] Freund RW. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J Comput Appl Math* 2000;123(1):395–421.
- [19] Giraud L, Langou J, Sylvand G. On the parallel solution of large industrial wave propagation problems. *J Comput Acoust* 2006;14(01):83–111.
- [20] Greenbaum A. Iterative methods for solving linear systems. In: Frontiers in applied mathematics, No. 17. Philadelphia: SIAM; 1997.
- [21] Gutknecht MH. Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. In: Siddiqi AH, Duff IS, Christensen O, editors. Modern mathematical models, methods and algorithms for real world systems, pages 420–447. New Delhi: Anamaya Publishers; 2007.
- [22] Hackbusch W. Multigrid methods and applications. Berlin: Springer-Verlag; 1985.
- [23] Kharchenko SA, Yerebin AY. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numer Linear Algebra Appl* 1995;2(1):51–77.
- [24] Lehoucq RB, Sorensen DC, Yang C. ARPACK Users' guide: solution of large scale eigenvalue problems with implicitly restarted arnoldi methods. Philadelphia: SIAM; 1998.
- [25] Morgan R.B. Implicitly restarted GMRES and arnoldi methods for nonsymmetric systems of equations. *SIAM J Matrix Anal Appl* 21(4):1112–1135.
- [26] Morgan RB. A restarted GMRES method augmented with eigenvectors. *SIAM J Matrix Anal Appl* 1995;16:1154–71.
- [27] Morgan RB. GMRES with deflated restarting. *SIAM J Sci Comput* 2002;24(1):20–37.
- [28] Paige CC, Parlett BN, van der Vorst HA. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numer Linear Algebra Appl* 1995;2(2):115–33.
- [29] Rao SM, Wilton DR, Glisson AW. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans Antennas Propagat* 1982;AP-30:409–18.
- [30] Robbè M, Sadkane M. Exact and inexact breakdowns in the block GMRES method. *Linear Algebra Appl* 2006;419(1):265–85.
- [31] Saad Y. Iterative methods for sparse linear systems. 2nd edition. Philadelphia: SIAM; 2003.
- [32] Sakurai T, Tadano H, Kuramashi Y. Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD. *Comput Phys Commun* 2010;181(1):113–17.
- [33] Sun DL, Huang TZ, Jing YF, Carpentieri B. A block GMRES method with deflated restarting for solving linear systems with multiple shifts and multiple right-hand sides. *Numer Linear Algebra Appl* 2017:e2148.
- [34] Tuminaro RS. A highly parallel multigrid-like method for the solution of the Euler equations. *SIAM J Sci Stat Comput* 1992;13:88–100.
- [35] van der Vorst HA, Vuik C. The superlinear convergence behaviour of GMRES. *J Comput Appl Math* 1993;48:327–41.
- [36] Saad Y. Analysis of augmented Krylov subspace techniques. *SIAM J Sci Comput* 1993;14:461–9.