Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 767042, 10 pages
http://dx.doi.org/10.1155/2013/767042

*Research Article*

# Parallel RFSAI-BFGS Preconditioners for Large Symmetric Eigenproblems

**L. Bergamaschi**[1] **and A. Martínez**[2]

[1] *Department of Civil, Environmental, and Architectural Engineering, University of Padua, Via Trieste 63, 35100 Padova, Italy*
[2] *Department of Mathematics, University of Padua, Via Trieste 63, 35100 Padova, Italy*

Correspondence should be addressed to L. Bergamaschi; berga@dmsa.unipd.it

Received 9 April 2013; Revised 18 July 2013; Accepted 1 August 2013

Academic Editor: D. R. Sahu

We propose a parallel preconditioner for the Newton method in the computation of the leftmost eigenpairs of large and sparse symmetric positive definite matrices. A sequence of preconditioners starting from an enhanced approximate inverse RFSAI (Bergamaschi and Martínez, 2012) and enriched by a BFGS-like update formula is proposed to accelerate the preconditioned conjugate gradient solution of the linearized Newton system to solve $A\mathbf{u} = q(\mathbf{u})\mathbf{u}$, $q(\mathbf{u})$ being the Rayleigh quotient. In a previous work (Bergamaschi and Martínez, 2013) the sequence of preconditioned Jacobians is proven to remain close to the identity matrix if the initial preconditioned Jacobian is so. Numerical results onto matrices arising from various realistic problems with size up to 1.5 million unknowns account for the efficiency and the scalability of the proposed low rank update of the RFSAI preconditioner. The overall RFSAI-BFGS preconditioned Newton algorithm has shown comparable efficiencies with a well-established eigenvalue solver on all the test problems.

## 1. Introduction

The computation of the $p \ll n$ leftmost eigenpairs of a symmetric positive definite (SPD) matrix $A$ is a common task in many scientific applications. Typical examples are offered by the vibrational analysis of mechanical structures [1]: the spectral superposition approach for the solution of large sets of 1st order linear differential equations [2] and the approximation of the generalized inverse of the graph Laplacian [3], to mention a few.

We denote

$$\lambda_1 < \lambda_2 < \cdots < \lambda_p < \cdots < \lambda_n \tag{1}$$

as the eigenvalues of $A$ and

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p, \ldots, \mathbf{v}_n \tag{2}$$

as the corresponding (normalized) eigenvectors.

In this paper, we propose to use the Newton method in the unit sphere [4, 5] or Newton-Grassman method, which constructs a sequence of vectors $\{\mathbf{u}_k\}$ by solving the linear systems

$$\left(I - \mathbf{u}_k\mathbf{u}_k^\top\right)\left(A - \theta_k I\right)\left(I - \mathbf{u}_k\mathbf{u}_k^\top\right)\mathbf{s} = -\left(A\mathbf{u}_k - \theta_k\mathbf{u}_k\right), \qquad \theta_k = \frac{\mathbf{u}_k^\top A\mathbf{u}_k}{\mathbf{u}_k^\top\mathbf{u}_k} \tag{3}$$

ensuring that the correction $\mathbf{s}$ be orthogonal to $\mathbf{u}_k$. Then the next approximation is set as $\mathbf{u}_{k+1} = \mathbf{t}\|\mathbf{t}\|^{-1}$, where $\mathbf{t} = \mathbf{u}_k + \mathbf{s}$. Linear system (3) is shown to be better conditioned than the one with $A - \theta_k I$. For SPD matrices, this scheme is shown to converge cubically to the wanted eigenvector, provided that the system (3) is solved exactly. The same linear system represents the *correction equation* in the well-known Jacobi-Davidson method [6], which in its turn can be viewed as an accelerated inexact Newton method [7]. When $A$ is SPD and the leftmost eigenpairs are being sought, it has been proved in [8] that the preconditioned conjugate gradient (PCG) method can be employed in the solution of the correction equation.

Starting from the findings in [9–11] the main contribution of this paper is the development of a sequence of preconditioners $\{P_k\}$ for the PCG solution of the Newton correction equation (3), based on the BFGS update of a given parallel initial preconditioner for the coefficient matrix $A$. A similar approach has been used in [12] where a rank-two modification of a given preconditioner is used to accelerated MINRES in the framework of the inexact Rayleigh quotient iteration. Updating (cheaply) a given preconditioner to obtain a new preconditioner for a slightly changed linear system is also common in optimization problems. See for example, [13–15].

The initial Newton vector is obtained after a small number of iterations of a conjugate gradient procedure for the minimization of the Rayleigh quotient (DACG, [16]). As the initial preconditioner we choose RFSAI [17], which is built by recursively applying the (factorized sparse approximate inverse) FSAI preconditioner developed in [18]. We elected a factorized approximate inverse preconditioner (AIP) since it is more naturally parallelizable than preconditioners based on ILU factorizations. Moreover, factorized variants of AIP provide better approximations to $A^{-1}$ for the same amount of storage than nonfactorized ones, because they can express denser matrices than the total number of nonzeros in their factors [19]. The RFSAI formulation enhances this characteristic, since the final preconditioner is implicitly built by a product of four or more triangular factors.

The combined DACG-Newton algorithm is used in the approximation of $p = 10$ eigenpairs of a number of matrices arising from various realistic applications of size up to $1.5 \times 10^6$ and number of nonzeros up to $6 \times 10^7$. Numerical results show that, in the solution of the correction equation, the PCG method preconditioned by BFGS displays much faster convergence than the same method when the preconditioner is kept fixed during the Newton process, in every test case. Moreover, the proposed approach is shown to outperform the pure DACG method.

## 2. BFGS Sequence of Preconditioners

*2.1. Choice of the Initial Preconditioner.* Following the developments in [17], we propose an implicit enlargement of the sparsity pattern using a banded target matrix $B$; the lower factor of the FSAI preconditioner is obtained by minimizing $\|B - GL\|_F$ over the set of matrices $G$ having a fixed sparsity pattern. Denoting with $G_{\text{out}}$ the result of this minimization, we compute explicitly the preconditioned matrix $S = G_{\text{out}} A G_{\text{out}}^T$ and then evaluate a second FSAI factor $G_{\text{in}}$ for $S$. Thus, the final preconditioner can be written as the product $G_{\text{out}}^T G_{\text{in}}^T G_{\text{in}} G_{\text{out}}$. This RFSAI—recursive FSAI—procedure can be iterated a number of times to yield a preconditioner written as a product of several triangular factors. This preconditioner has shown its efficiency in the acceleration of the PCG onto realistic geomechanical problems of very large size [17].

We denote FSAIout as the procedure which computes the $G_{\text{out}}$ factor by minimizing $\|B - GL\|$, where $B$ is an arbitrary banded matrix. FSAIout depends on the `nband` parameter in addition to the usual FSAI parameters: $\delta_{\text{out}}$ prefiltration threshold, $d_{\text{out}}$ power of $\widetilde{A}$ defining the sparsity pattern, and $\varepsilon_{\text{out}}$ postfiltration parameter.

The second preconditioner factor, $G_{\text{in}}$, is the result of the FSAI procedure applied to the whole product matrix $G_{\text{out}} A G_{\text{out}}^T$, with parameters $\delta_{\text{in}}$, $d_{\text{in}}$, and $\varepsilon_{\text{in}}$. The steps to obtain the final RFSAI preconditioner are summarized in Algorithm 1.

Following the idea described in [10], we propose a sequence of preconditioners for the Newton systems using the BFGS rank-two update. To precondition the initial Newton system as follows:

$$J_0 \mathbf{s}_0 = -\mathbf{r}, \tag{4}$$

where

$$\begin{gathered} J_0 = \left(I - \mathbf{u}_0\mathbf{u}_0^\top\right)\left(A - \theta_0 I\right)\left(I - \mathbf{u}_0\mathbf{u}_0^\top\right), \\ \mathbf{r} = -\left(A\mathbf{u}_0 - \theta_0\mathbf{u}_0\right), \\ \theta_0 = \mathbf{u}_0^\top A \mathbf{u}_0. \end{gathered} \tag{5}$$

We choose to use a projected RFSAI preconditioner, $P_0 = (I - \mathbf{u}_0\mathbf{u}_0^\top)\widehat{P}_0(I - \mathbf{u}_0\mathbf{u}_0^\top)$ with $\widehat{P}_0 = W^T W$ and $W = G_{\text{in}} G_{\text{out}}$.

*2.2. Update of Initial Preconditioner by BFGS-Like Rank-Two Corrections.* A sequence of projected preconditioners for the subsequent linear systems $J_{k+1}\mathbf{s}_{k+1} = -\mathbf{r}_{k+1}$ may be defined by

$$\widehat{P}_{k+1} = -\frac{\mathbf{s}\mathbf{s}^\top}{\mathbf{s}^\top\mathbf{r}} + \left(I - \frac{\mathbf{s}\mathbf{r}^\top}{\mathbf{s}^\top\mathbf{r}}\right)\widehat{P}_k\left(I - \frac{\mathbf{r}\mathbf{s}^\top}{\mathbf{s}^\top\mathbf{r}}\right), \tag{6}$$

where $\mathbf{s} \equiv \mathbf{s}_k$ the solution of the previous correction equation and $\mathbf{r} \equiv \mathbf{r}_k$. In view of the cubic convergence of the Newton process, we used the residual $-\mathbf{r}_k$ instead of $\mathbf{y}_k = \mathbf{r}_{k+1} - \mathbf{r}_k$.

Theorem 3 of Section 3 will state that the preconditioner defined in (6) is SPD if $\widehat{P}_k$ is so.

## 3. Theoretical Analysis of the Preconditioner

*3.1. Finding the Smallest Eigenpair.* We recall in this section the main theoretical results regarding the sequence of preconditioners previously defined. Differently from the classical papers which study BFGS convergence properties, here our Jacobian matrix $J(\mathbf{u}) = (I - \mathbf{u}\mathbf{u}^\top)(A - q(\mathbf{u})I)(I - \mathbf{u}\mathbf{u}^\top)$ is singular whatever $\mathbf{u}$, in particular it is singular when $\mathbf{u}$ is equal to the exact eigenvector. The theoretical analysis of the "goodness" of the preconditioner will be, therefore, completely different, though obtaining similar results, than that proposed, for example, in [10, 20]. In the following developments we will indicate as $\mathbf{v}_1$ the exact eigenvector corresponding to the smallest exact eigenvalue $\lambda_1$. The error vector at step $k$ is denoted by $\mathbf{e}_k = \mathbf{u}_k - \mathbf{v}_1$, while the error in the eigenvalue approximation is $\varepsilon_k = \theta_k - \lambda_1 (>0)$. It is easily proved that there is a constant $M$ independent of $k$ such that

$$\varepsilon_k \le M\|\mathbf{e}_k\|^2. \tag{7}$$

*Remark 1.* At first sight the Jacobian matrix in the correction equation is singular, but this does not matter since the PCG

```
INPUT: nband, $\delta_{\text{out}}$, $d_{\text{out}}$, $\varepsilon_{\text{out}}$, $\delta_{\text{in}}$, $d_{\text{in}}$, $\varepsilon_{\text{in}}$
Compute the first lower triangular factor: $G_{\text{out}} = \texttt{FSAIout}\,(A, \texttt{nband}, \delta_{\text{out}}, d_{\text{out}}, \varepsilon_{\text{out}})$
Compute the product: $A^{(1)} = G_{\text{out}} A G_{\text{out}}^T$
Compute the second lower triangular factor: $G_{\text{in}} = \text{FSAI}\left(A^{(1)}, \delta_{\text{in}}, d_{\text{in}}, \varepsilon_{\text{in}}\right)$
```

Algorithm 1: RFSAI computation.

algorithm is run within the subspace of vectors orthogonal to $\mathbf{u}_k$ (in fact also $\mathbf{r}^\top \mathbf{u}_k = 0$). Thus, notion of positive definiteness, eigenvalue distribution, condition number, norms, and so forth, apply as usual but with respect to matrices restricted to this subspace.

The following lemma will bound the extremal eigenvalues of $J_k$ in the subspace orthogonal to $\mathbf{u}_k$.

**Lemma 2.** *There is a positive number $\delta$ such that if $\|\mathbf{e}_k\| < \delta$ then*

$$J_k = \left(I - \mathbf{u}_k \mathbf{u}_k^\top\right)\left(A - \theta_k I\right)\left(I - \mathbf{u}_k \mathbf{u}_k^\top\right) \tag{8}$$

*is SPD in the subspace orthogonal to $\mathbf{u}_k$. Moreover the following bounds hold:*

$$\frac{\lambda_2 - \lambda_1}{2} < \mathbf{z}^\top J_k \mathbf{z} < \lambda_n \tag{9}$$

*for every unit norm vector $\mathbf{z}$ orthogonal to $\mathbf{u}_k$.*

*Proof.* See [21]. □

The previous lemma allows us to prove that the preconditioner defined in (6) is SPD, as stated in the following theorem.

**Theorem 3.** *If the correction equation is solved exactly, then any matrix $\widehat{P}_k$ defined by (6) is SPD and hence $P_k$ is SPD in the subspace orthogonal to $\mathbf{u}_k$.*

*Proof.* See [21]. □

Let us define the difference between the preconditioned Jacobian and the identity matrix as

$$E_k = I - J_k^{1/2} P_k J_k^{1/2}. \tag{10}$$

Since by definition we have $J_k \mathbf{u}_k = 0$ then $\mathbf{u}_k$ is the eigenvector of $J_k$ corresponding to the zero eigenvalue. Hence, since also $J_k^{1/2}\mathbf{u}_k = 0$ the error $E_k$ can also be defined as

$$\begin{aligned} E_k &= I - J_k^{1/2} P_k J_k^{1/2} \\ &= I - J_k^{1/2}\left(I - \mathbf{u}_k \mathbf{u}_k^\top\right) \\ &\quad \times \widehat{P}_k \left(I - \mathbf{u}_k \mathbf{u}_k^\top\right) J_k^{1/2} \\ &= I - J_k^{1/2} \widehat{P}_k J_k^{1/2}. \end{aligned} \tag{11}$$

The following technical lemma will bound the norm of $\widehat{P}_k$ in terms of that of $E_k$. Being $\widehat{P}_k$ SPD we can define its norm in the space orthogonal to $\mathbf{u}_k$ as

$$\left\|\widehat{P}_k\right\| = \sup_{\mathbf{w}\perp\mathbf{u}_k, \mathbf{w}\neq 0} \frac{\mathbf{w}^\top \widehat{P}_k \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}. \tag{12}$$

**Lemma 4.** *There is a positive number $\delta$ such that if $\|\mathbf{e}_k\| < \delta$ then*

$$\left\|\widehat{P}_k\right\| \leq \frac{2}{\lambda_2 - \lambda_1}\left(1 + \|E_k\|\right). \tag{13}$$

*Proof.* See [21]. □

The next lemma will relate the norms of the difference $\mathbf{s}$ and of the error vector $\mathbf{e}_k$.

**Lemma 5.** *There exists a positive number $\delta$ s.t. if $\|\mathbf{e}_k\| < \delta$ then*

$$\|\mathbf{s}\| \leq 3\left\|\mathbf{e}_k\right\|. \tag{14}$$

*Proof.* See [21]. □

Before stating Theorem 7 we need to state as a last preliminary result that also the difference between the square root of two consecutive Jacobians is bounded in terms of the norm of the error vector.

**Lemma 6.** *Let $S_k = J_{k+1}^{1/2} - J_k^{1/2}$. Then there is a positive number $\delta$ s.t. if $\|\mathbf{e}_k\| < \delta$ then*

$$\left\|S_k\right\| \leq c_3 \sqrt{\left\|\mathbf{e}_k\right\|}. \tag{15}$$

*for a suitable constant $c_3$.*

*Proof.* See [21]. □

The following theorem will state the main theoretical result of this Section: the so called *bounded deterioration* [22] of the preconditioner at step $k+1$ with respect to that of step $k$. In other words it can be proved that the distance of the preconditioned matrix from the identity matrix at step $k+1$ is less or equal than that at step $k$ plus a constant that may be small as desired, depending on the closeness of $\mathbf{u}_0$ to the exact eigenvector. We report also the proof of this theorem, which is taken from [21].

**Theorem 7.** *Let $\delta_0$ be such that $\|E_0\| < \delta_0$, there is a positive number $\delta$ s.t. if $\|\mathbf{e}_0\| < \delta$ then*

$$\left\|E_{k+1}\right\| \leq \left\|E_k\right\| + K\sqrt{\left\|\mathbf{e}_k\right\|} \tag{16}$$

*for a suitable constant $K$.*

*Proof.* After defining $N = S_k\widehat{P}_{k+1}J_{k+1}^{1/2} + J_{k+1}^{1/2}\widehat{P}_{k+1}S_k + S_k\widehat{P}_{k+1}S_k$, we can write

$$\begin{aligned}
E_{k+1} &= I - J_{k+1}^{1/2}\widehat{P}_{k+1}J_{k+1}^{1/2} \\
&= I - \left(J_k^{1/2} + S_k\right)\widehat{P}_{k+1}\left(J_k^{1/2} + S_k\right) \\
&= I - J_k^{1/2}\widehat{P}_{k+1}J_k^{1/2} - N \\
&= I - J_k^{1/2}\frac{\mathbf{s}\mathbf{s}^\top}{\mathbf{s}^\top J_k\mathbf{s}}J_k^{1/2} \\
&\quad - J_k^{1/2}\left(I - \frac{\mathbf{s}\mathbf{s}^\top J_k}{\mathbf{s}^\top J_k\mathbf{s}}\right)\widehat{P}_k\left(I - \frac{J_k\mathbf{s}\mathbf{s}^\top}{\mathbf{s}^\top J_k\mathbf{s}}\right) \\
&\quad \times J_k^{1/2} - N \qquad (17) \\
&= I - \frac{J_k^{1/2}\mathbf{s}\mathbf{s}^\top J_k^{1/2}}{\mathbf{s}^\top J_k\mathbf{s}} \\
&\quad - \left(I - \frac{J_k^{1/2}\mathbf{s}\mathbf{s}^\top J_k^{1/2}}{\mathbf{s}^\top J_k\mathbf{s}}\right)J_k^{1/2}\widehat{P}_kJ_k^{1/2} \\
&\quad \times\left(I - \frac{J_k^{1/2}\mathbf{s}\mathbf{s}^\top J_k^{1/2}}{\mathbf{s}^\top J_k\mathbf{s}}\right) - N \\
&= W + W\left(E_k - I\right)W + N = WE_kW - N,
\end{aligned}$$

where we set $\mathbf{w} = J_k^{1/2}\mathbf{s}/\|J_k^{1/2}\mathbf{s}\|$ and $W = I - \mathbf{w}\mathbf{w}^\top$; $W$ is an orthogonal projector since $\|\mathbf{w}\| = 1$. To bound $\|N\|$, we will use Lemmas 4 and 6 as follows:

$$\begin{aligned}
\|N\| &\le \frac{2}{\lambda_2 - \lambda_1}\left(1 + \|E_{k+1}\|\right) \\
&\quad \times\left(2c_3\sqrt{\|\mathbf{e}_k\|}\sqrt{\lambda_n} + c_3^2\sqrt{\delta}\sqrt{\|\mathbf{e}_k\|}\right) \qquad (18) \\
&= c_4\left(1 + \|E_{k+1}\|\right)\sqrt{\|\mathbf{e}_k\|}.
\end{aligned}$$

Now taking norms in (17) yields

$$\|E_{k+1}\| \le \|E_k\| + c_4\left(1 + \|E_{k+1}\|\right)\sqrt{\|\mathbf{e}_k\|}, \qquad (19)$$

which can be rewritten as

$$\|E_{k+1}\|\left(1 - c_4\sqrt{\|\mathbf{e}_k\|}\right) \le \|E_k\| + c_4\sqrt{\|\mathbf{e}_k\|}. \qquad (20)$$

From (20), we derive a bound for $\|E_k\|$. If $\sqrt{\delta} < 1/2c_4$ then

$$\begin{aligned}
\|E_k\| &\le 2\|E_{k-1}\| + 1 \\
&\le \cdots \le 2^k\|E_0\| + 2^k - 1 \qquad (21) \\
&\le 2^k\left(\delta_0 + 1\right) = c_5.
\end{aligned}$$

Again from (20) and using the bound (21) we finally have

$$\begin{aligned}
\|E_{k-1}\| &\le \frac{\|E_k\| + c_4\sqrt{\|\mathbf{e}_k\|}}{1 - c_4\sqrt{\|\mathbf{e}_k\|}} \\
&\le \left(\|E_k\| + c_4\sqrt{\|\mathbf{e}_k\|}\right)\cdot\left(1 + 2c_4\sqrt{\|\mathbf{e}_k\|}\right) \qquad (22) \\
&\le \|E_k\| + 2c_4\sqrt{\|\mathbf{e}_k\|}c_5 + c_4\left(1 + 2c_4\delta\right)\sqrt{\|\mathbf{e}_k\|} \\
&= \|E_k\| + c_4\left(2c_5 + 1 + 2c_4\delta\right)\sqrt{\|\mathbf{e}_k\|}.
\end{aligned}$$

Setting $K = c_4(2c_5 + 1 + 2c_4\delta)$ completes the proof. □

*3.2. Computing Several Eigenpairs.* When seeking an eigenvalue different from $\lambda_1$, say $\lambda_j$, the Jacobian matrix changes as

$$J_k = \left(I - QQ^\top\right)\left(A - \theta_kI\right)\left(I - QQ^\top\right), \qquad (23)$$

where $Q = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_j & \mathbf{u}_k\end{bmatrix}$ is the matrix whose first $j$ columns are the previously computed eigenvectors. Analogously, also the preconditioner must be chosen orthogonal to $Q$ as

$$P_{k+1} = \left(I - QQ^\top\right)\widehat{P}_{k+1}\left(I - QQ^\top\right). \qquad (24)$$

The theoretical analysis developed in Section 3.1 applies with small technical variants also in this case since it is readily proved that $J_{k+1}^{1/2}P_{k+1}J_{k+1}^{1/2} = J_{k+1}^{1/2}\widehat{P}_{k+1}J_{k+1}^{1/2}$. The most significant changes regard the definition of $\varepsilon_k = \theta_k - \lambda_j$, $\mathbf{e}_k = \mathbf{u}_k - \mathbf{v}_j$, and the statement of Lemma 2 (and the proof of Lemma 4 that uses its results); namely, the bound for the smallest eigenvalue of $J_k$ which in the general case becomes

$$\mathbf{z}^\top J_k\mathbf{z} > \frac{\lambda_{j+1} - \lambda_j}{2} \qquad (25)$$

for every unit norm vector $\mathbf{z}$ such that $Q^\top\mathbf{z} = 0$.

## 4. Implementation

*4.1. Choosing an Initial Eigenvector Guess.* As mentioned in Section 1, another important issue in the efficiency of the Newton approach for eigenvalue computation is represented by the appropriate choice of the initial guess. We propose here to perform some preliminary iterations using the DACG eigensolver [16, 23, 24], which is based on the preconditioned conjugate gradient (nonlinear) minimization of the Rayleigh quotient. This method has proven very robust, and not particularly sensitive to the initial vector, in the computation of a few eigenpairs of large SPD matrices. Recently in [25], the DACG method has been compared with the Jacobi-Davidson method in parallel environments, showing similar performances when seeking a small number of eigenpairs $(1 \div 5)$.

```
INPUT: Vector $\mathbf{g}_l$, scalar products $\alpha_s = \mathbf{s}_s^\top \mathbf{r}_s, s = 0, \ldots, k-1$.
$\mathbf{w} = \mathbf{g}_l$
for $s := k-1$ to 0
    (1) $a_s := \mathbf{s}_s^\top \mathbf{w}/\alpha_s$
    (2) $\mathbf{w} := \mathbf{w} - a_s \mathbf{r}_s$
end for
$c = \widehat{P}_0 \mathbf{w}$
for $s := 0$ to $k-1$
    (1) $b := \mathbf{r}_s^\top \mathbf{c}/\alpha_s$
    (2) $\mathbf{c} := \mathbf{c} - (a_s + b)\, \mathbf{s}_s$
end for
$\mathbf{z} := Q^\top \mathbf{c}$
$\mathbf{c} := \mathbf{c} - Q\mathbf{z}$
```

Algorithm 2: Computation of $\mathbf{c} = P_k \mathbf{g}_l$ for the BFGS preconditioner.

*4.2. Implementation of the BFGS Preconditioner Update.* At a certain nonlinear iteration level, $k$, we need to compute $\mathbf{c} = P_k \mathbf{g}_l$, where $\mathbf{g}_l$ is the residual of the linear system at iteration $l$. Let us suppose we compute an initial preconditioner $P_0$. Then, at the initial nonlinear iteration $k = 0$, we simply have $\mathbf{c} = P_0 \mathbf{z}_l$. At step $k+1$ the preconditioner $\widehat{P}_{k+1}$ is defined recursively by (6) while $P_{k+1}$ using (24) can be written as

$$\begin{aligned} P_{k+1} &= (I - QQ^\top)\, \widehat{P}_{k+1}\, (I - QQ^\top) \\ &= (I - QQ^\top) \left\{ \left( I - \frac{\mathbf{s}\mathbf{r}^\top}{\mathbf{s}^\top \mathbf{r}} \right) \widehat{P}_k \right. \\ &\qquad \left. \times \left( I - \frac{\mathbf{r}\mathbf{s}^\top}{\mathbf{s}^\top \mathbf{r}} \right) - \frac{\mathbf{s}\mathbf{s}^\top}{\mathbf{s}^\top \mathbf{r}} \right\} \\ &\quad \times (I - QQ^\top). \end{aligned} \tag{26}$$

To compute vector $\mathbf{c}$ first we observe that $\mathbf{g}_l$ is orthogonal to $Q$ so there is no need to apply matrix $I - QQ^\top$ on the right of (26). Application of preconditioner $\widehat{P}_{k+1}$ to the vector $\mathbf{g}_l$ can be performed at the price of $2k$ dot products and $2k$ daxpys as described in Algorithm 2. The scalar products $\alpha_k = \mathbf{s}_k^\top \mathbf{r}_k$, which appear at the denominator of $\widehat{P}_{k+1}$, can be computed once and for all before starting the solution of the $(k+1)$th linear system. Last, the obtained vector $\mathbf{c}$ must be orthogonalized against the columns of $Q$ by a classical Gram-Schimdt procedure.

*4.3. PCG Solution of the Correction Equation.* As a Krylov subspace solver for the correction equation, we chose the preconditioned conjugate gradient (PCG) method since the Jacobian $J_k$ has been shown to be SPD in the subspace orthogonal to $\mathbf{u}_k$. Regarding the implementation of PCG, we mainly refer to the work [8], where the author shows that it is possible to solve the linear system in the subspace orthogonal to $\mathbf{u}_k$ and hence, the projection step needed in the application of $J_k$ can be skipped. Moreover, we adopted the exit strategy for the linear system solution described in the previous paper, which allows for stopping the PCG iteration, in addition to the classical exit test based on a tolerance on the relative residual and on the maximum number of iterations, whenever the current solution $\mathbf{x}_l$ satisfies

$$\left\| \mathbf{r}_{k,l} \right\| = \left\| A\mathbf{x}_l - \frac{\mathbf{x}_l^\top A\mathbf{x}_l}{\mathbf{x}_l^\top \mathbf{x}_l} \mathbf{x}_l \right\| < \tau \left( \mathbf{x}_l^\top A\mathbf{x}_l \right), \tag{27}$$

or when the decrease of $\|\mathbf{r}_{k,l}\|$ is slower than the decrease of $\|\mathbf{g}_l\|$, because in this case further iterating does not improve the accuracy of the eigenvector. Note that this dynamic exit strategy implicitly defines an inexact Newton method since the correction equation is not solved "exactly", that is, up to machine precision.

We have implemented the PCG method as described in Algorithm 5.1 of [8] with the obvious difference in the application of the preconditioner which in this case is accomplished as described in Algorithm 2.

*4.4. Implementation of the DACG-Newton Method.* The BFGS preconditioner defined in Algorithm 2 suffers from two main drawbacks, namely, increasing costs of memory for storing $\mathbf{s}$ and $\mathbf{r}$ and the increasing cost of preconditioner application with the iteration index $k$. Note that these drawbacks are common to many iterative schemes, such as sparse (Limited Memory) Broyden implementations [26], GMRES [27], and Arnoldi method for eigenvalue problems [28]. If the number of nonlinear iterations is high the application of BFGS update may be too costly as compared with the expected reduction in the iteration number. To this aim we define $k_{\max}$ the maximum number of rank-two corrections we allow. When the nonlinear iteration counter $k$ is larger than $k_{\max}$, the vectors $\mathbf{s}_i$, $\mathbf{r}_i$, $i = k - k_{\max}$ are substituted with the last computed $\mathbf{s}_k$, $\mathbf{r}_k$. Vectors $\{\mathbf{s}_i, \mathbf{r}_i\}$ are stored in a matrix $V$ with $n$ rows and $2 \times k_{\max}$ columns.

The implementation of our DACG-Newton method for computing the leftmost eigenpairs of large SPD matrices is described in Algorithm 3.

*4.5. Parallel Implementation.* We have developed a parallel code which implements the construction and application inside parallel PCG, of the RFSAI algorithm along with the BFGS updates. The resulting program is written in Fortran 90

```
INPUT: Matrix $A$;
        number of sought eigenpairs $n_{\mathrm{eig}}$;
        tolerance and maximum number of its for the outer iteration: $\tau$, ITMAX;
        tolerance for the initial eigenvector guess $\tau_{\mathrm{DACG}}$;
        tolerance and maximum number of its for the inner iteration: $\tau_{\mathrm{PCG}}$.
ITMAX$_{\mathrm{PCG}}$;
        parameters for the RFSAI preconditioner,
                $\delta_{\mathrm{out}}$, $d_{\mathrm{out}}$, $\varepsilon_{\mathrm{out}}$, nband, for the 1st FSAI factor and,
                $\delta_{\mathrm{in}}$, $d_{\mathrm{in}}$, $\varepsilon_{\mathrm{in}}$ for the 2nd factor;
        maximum allowed rank-two update in the BFGS preconditioner: $k_{\max}$.
$\widetilde{Q} := [\ \ ]$.
Compute $\widehat{P}_0$, an RFSAI preconditioner for $A$,
for $j := 1$ to $n_{\mathrm{eig}}$
  (1) Choose $\mathbf{x}_0$ such that $\widetilde{Q}^\top \mathbf{x}_0 = 0$;
  (2) Compute $\mathbf{u}_0$, an approximation to $\mathbf{v}_j$ by the DACG procedure with initial
        vector $\mathbf{x}_0$, preconditioner $\widehat{P}_0$ and tolerance $\tau_{\mathrm{DACG}}$;
  (3) $k := 0$, $\theta_k := \mathbf{u}_k^\top A \mathbf{u}_k$;
  (4) while $\|A\mathbf{u}_k - \theta_k \mathbf{u}_k\| > \tau\theta_k$ and $k <$ IMAX do
        (1) $Q := [\widetilde{Q}\ \ \mathbf{u}_k]$.
        (2) Solve $J_k \mathbf{s}_k = -\mathbf{r}_k$ for $\mathbf{s}_k \perp Q$ by the PCG method with preconditioner $P_k$ and tolerance $\tau_{\mathrm{PCG}}$.
        (3) $\mathbf{u}_{k+1} := (\mathbf{u}_k + \mathbf{s}_k)/\|\mathbf{u}_k + \mathbf{s}_k\|$, $\theta_{k+1} = \mathbf{u}_{k+1}^\top A \mathbf{u}_{k+1}$.
        (4) $k_1 = k$ MOD $k_{\max}$; $V(*, 2k_1 + 1) := s_k$, $V(*, 2k_1 + 2) := \mathbf{r}_k$.
        (5) $k := k + 1$
  (6) end while
  (7) Assume $\mathbf{v}_j = \mathbf{u}_k$ and $\lambda_j = \theta_k$. Set $\widetilde{Q} := [\widetilde{Q}\ \ \mathbf{v}_j]$
end for
```

ALGORITHM 3: DACG-Newton algorithm.

and exploits the MPI library for exchanging data among the processors. We use a block row distribution of all matrices, that is, with complete rows assigned to different processors. All the matrices are stored in static data structures in CSR format.

Parallelization of the FSAI preconditioner, which is the basis of the parallel RFSAI construction, has been performed and tested for example, in [29–31] where prefiltration and postfiltration have been implemented together with a priori sparsity pattern based on nonzeros of $A^d$ with $d \leq 4$.

The code makes also use of an optimized parallel matrix-vector product which has been developed in [32] showing its effectiveness up to 1024 processors.

## 5. Numerical Results

*5.1. Test Problems.* We report the results of our experiments with the RFSAI preconditioner in the solution of a set of problems of large size.

The test matrices are realistic examples arising from 3D FE discretization of fluid flow and geomechanical models. Described in detail as follows:

(1) `FLOW3D-663` arises from tetrahedral FE discretization of an elliptic PDE describing fluid flow in porous media.

(2) `EMILIA-923` arises from the regional geomechanical model of a deep hydrocarbon reservoir. It is obtained discretizing the structural problem with tetrahedral finite elements. Due to the complex geometry of the geological formation, it was not possible to obtain a computational grid characterized by regularly shaped elements.

(3) `LONG-1103` is the structural SPD block obtained from a 3D coupled consolidation problem of a geological formation, discretized with tetrahedral finite elements on a higly irregular computational grid.

(4) `GEO-1438` arises from a regional geomechanical model of the sedimentary basin underlying the Venice lagoon discretized by a linear FE with randomly heterogeneous properties. The computational domain is a box with an areal extent of 50 × 50 km and 10 km deep consisting of regularly shaped tetrahedral finite elements.

Matrices 2 to 4 are publicly available in the University of Florida Sparse Matrix Collection at http://www.cise.ufl.edu/research/sparse/matrices/. The size and number of nonzero elements for each matrix is provided in Table 1.

All tests are performed on the IBM BlueGene\Q cluster at the CINECA Centre for HPC, equipped with IBM PowerA2 processors at 1.6 GHz with 10240 nodes, 163 840 computing cores, and 164 Tbytes of internal network RAM. The Fortran 90 code is compiled with the native IBM `xlf` compiler using the `-O3 -qarch=qp -qtune=qp` options.

Table 1: Size $n$ and number of nonzeros (nnz) of the test matrices.

| Name | $n$ | nnz |
|---|---|---|
| FLOW3D-663 | 663 390 | 9 762 648 |
| EMILIA-923 | 923 136 | 41 005 206 |
| LONG-1103 | 1 102 614 | 48 987 558 |
| GEO-1438 | 1 437 960 | 63 156 690 |

Table 2: Default values of parameters.

| | |
|---|---|
| Number of eigenpairs to compute | $n_{\text{eig}} = 10$ |
| Parameters for the outer iteration | $\tau = 10^{-8}$, ITMAX = 50 |
| Tolerance for the initial eigenvector guess | $\tau_{\text{DACG}} = 10^{-2}$ |
| Parameters for the PCG iteration | $\tau_{\text{PCG}} = 10^{-2}$, $\text{ITMAX}_{\text{PCG}} = 50$ |

*5.2. Results on a Fixed Number of Processors.* We now compare the performance of the DACG-Newton algorithm for various $k_{\max}$ values. We tested the proposed algorithm in the computation of the 10 smallest eigenpairs of the afore mentioned large matrices arising from various realistic applications. In all the runs, unless differently specified, we selected the values of the parameters as reported in Table 2.

We will also compute the fill-in $\rho$ of the initial preconditioner defined as

$$\rho = \frac{\text{nnz}\left(G_{\text{out}}\right) + \text{nnz}\left(G_{\text{in}}\right)}{\text{nnz}\left(\text{lower triangular part of } A\right)}. \qquad (28)$$

In Table 3, we report the parameters for the RFSAI preconditioner which we selected for the four-test problems.

We first analyze the influence of parameter $k_{\max}$ in the convergence of the Newton-DACG method. To this aim we provide, in Table 4 the values of outer iteration number, total matrix-vector products (MVP) and CPU time in evaluating 10 eigenpairs of matrix EMILIA-923 for different values of $k_{\max}$. From the table, we notice how iteration number and CPU time monotonically decreases with increasing $k_{\max}$. Moreover, for $k_{\max} \le 2$, the iterative process reached the maximum number of iterations before fulfilling the exit test on the relative residual. Another evidence of the efficiency of the BFGS update is given in Figure 1, where we plot the relative residual norm $\|\mathbf{r}_{k,l}\|/(\mathbf{x}_l^T A \mathbf{x}_l)$ versus number of linear iterations in converging to eigenpair number 3 of matrix FLOW3D-663. Note that, with $k_{\max} = 0$, that is, using the constant RFSAI preconditioner, the Newton phase could not reach the $10^{-8}$ accuracy within the prescribed ITMAX = 50 maximum number of outer iterations.

In Table 5, we report the number of matrix vector products and CPU times for Newton-DACG as compared with "pure" DACG, that is, with DACG run with a final tolerance of $10^{-8}$. In every test problem DACG-Newton reveals superior to DACG.

In particular, for problem FLOW3D-663, the improvement provided by DACG-Newton is impressive. The DACG method, if run until the final tolerance $\tau = 10^{-8}$, is very slow due to the very small relative separation between consecutive eigenvalues $\xi_j = (\lambda_{j+1} - \lambda_j)/\lambda_j$, which drives convergence of

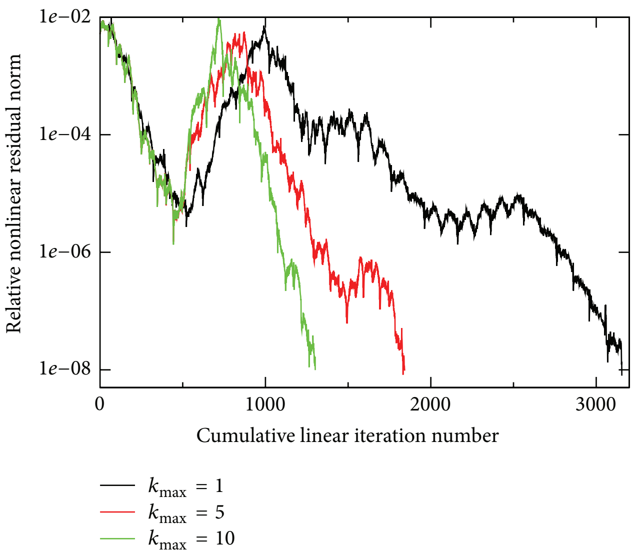

Figure 1: Convergence profile of the relative residual norm versus cumulative inner iterations for eigenvalue number 3, matrix FLOW3D-663.

this PCG-like solver [16, 25]. In fact the ratio $\xi_j$ also influences the convergence of Newton's method being related to the condition number of the Jacobian. However, the (RFSAI-BFGS preconditioned) DACG-Newton algorithm seems less sensitive to this ill-conditioning.

*5.3. Parallel Results and Scalability.* We now analyze the parallel efficiency of the previously described eigensolver preconditioned by RFSAI-BFGS. We will use a strong scaling measure to see how CPU times vary with the number of processors for a fixed total problem size. Denoting with $T_p$ the total CPU elapsed times expressed in seconds on $p$ processors, we define relative measures of the parallel efficiency and speedup of our code. We define $S_p^{(\overline{p})}$ as the pseudo speedup computed with respect to the smallest number of processors $(\overline{p})$ used to solve a given problem and $E_p^{(\overline{p})}$ the corresponding efficiency:

$$S_p^{(\overline{p})} = \frac{T_{\overline{p}}\overline{p}}{T_p}, \qquad E_p^{(\overline{p})} = \frac{S_p^{(\overline{p})}}{p} = \frac{T_{\overline{p}}\overline{p}}{T_p p}. \qquad (29)$$

The results obtained with the four test matrices are presented in Tables 6, 7, 8, and 9. As a general comment, we may observe that the overall code scales well for the three largest matrices, up to 256 processors (512 for problem GEO-1438). The parallel pseudo efficiency, as expected, decreases with growing number of processors but it is roughly 50% for the largest number of processors employed.

Regarding the FLOW3D-663 problem, which is characterized by a relatively small dimension and high sparsity (≈15 nonzeros per row), the parallel efficiency starts to worsen with $p = 128$ processors.

Table 3: RFSAI parameters used for the eigensolution of the four test problems.

| Matrix name | $G_{\text{out}}$ | | | nband | $G_{\text{in}}$ | | | $\rho$ |
|---|---|---|---|---|---|---|---|---|
| | $\delta$ | $d$ | $\varepsilon$ | | $\delta$ | $d$ | $\varepsilon$ | |
| FLOW3D-663 | 0.01 | 4 | 0.05 | 1 | 0.01 | 2 | 0.05 | 1.16 |
| EMILIA-923 | 0.1 | 4 | 0.05 | 1 | 0.05 | 2 | 0.01 | 0.74 |
| LONG-1103 | 0.15 | 4 | 0.1 | 1 | 0.1 | 4 | 0.1 | 0.29 |
| GEO-1438 | 0.15 | 4 | 0.1 | 1 | 0.1 | 2 | 0.05 | 0.34 |

Table 4: Influence of parameter $k_{\max}$ in the number of iterations of the Newton phase. Problem EMILIA-923 with $p = 64$ processors.

| $k_{\max}$ | Outer its | MVP | CPU | Comments |
|---|---|---|---|---|
| 0 | 431 | 15035 | 421.25 | $\text{ITMAX}_{\text{PCG}} = 100$. No convergence within ITMAX = 50. Final (average) residual norm of $10^{-6}$ |
| 1 | 438 | 10899 | 336.74 | No convergence within ITMAX = 50. Final (average) residual norm of $10^{-6}$ |
| 2 | 343 | 8329 | 258.22 | No convergence within ITMAX = 50. Final (average) residual norm of $10^{-7}$ |
| 5 | 267 | 6839 | 198.19 | |
| 10 | 204 | 5629 | 178.04 | |
| 20 | 185 | 5616 | 164.18 | |

Table 5: Comparison between DACG and Newton-DACG for the four test problems on $p = 64$ processors. Results on problem LONG-1103 are obtained using $\tau_{\text{DACG}} = 0.5$.

| Matrix name | DACG | | Newton-DACG | | |
|---|---|---|---|---|---|
| | MVP | CPU time | $k_{\max}$ | MVP | CPU time |
| FLOW3D-663 | 51427 | 658.55 | 10 | 15833 | 229.33 |
| EMILIA-923 | 13404 | 410.93 | 20 | 10649 | 346.83 |
| LONG-1103 | 15222 | 390.05 | 20 | 8857 | 256.59 |
| GEO-1438 | 12667 | 354.63 | 20 | 10441 | 326.99 |

Table 6: Iterations, timings, and pseudo efficiencies for problem FLOW3D-663.

| $p$ | precond | DACG | | Newton | | Total | | $E_p^{(8)}$ |
|---|---|---|---|---|---|---|---|---|
| | CPU | CPU | MVP | CPU | MVP | CPU | MVP | |
| 8 | 42.43 | 238.34 | 4291 | 704.78 | 11112 | 985.55 | 15403 | |
| 16 | 27.00 | 131.29 | 4291 | 408.05 | 11681 | 566.33 | 15972 | 0.87 |
| 32 | 18.17 | 79.85 | 4292 | 241.79 | 11275 | 339.81 | 15567 | 0.73 |
| 64 | 89.32 | 54.57 | 4291 | 166.77 | 11542 | 230.67 | 15833 | 0.53 |
| 128 | 84.89 | 40.68 | 4292 | 119.82 | 11177 | 165.39 | 15469 | 0.37 |

Table 7: Iterations, timings, and pseudo efficiencies for problem EMILIA-923.

| $p$ | precond | DACG | | Newton | | Total | | $E_p^{(16)}$ |
|---|---|---|---|---|---|---|---|---|
| | CPU | CPU | MVP | CPU | MVP | CPU | MVP | |
| 16 | 85.91 | 478.29 | 5379 | 527.05 | 5616 | 1091.24 | 10995 | |
| 32 | 55.44 | 254.57 | 5377 | 286.06 | 5620 | 596.07 | 10997 | 0.92 |
| 64 | 35.47 | 149.71 | 5378 | 164.18 | 5616 | 349.36 | 10994 | 0.78 |
| 128 | 22.30 | 88.96 | 5378 | 98.61 | 5618 | 209.87 | 10996 | 0.65 |
| 256 | 13.53 | 58.98 | 5378 | 66.68 | 5616 | 139.19 | 10994 | 0.49 |

Table 8: Iterations, timings, and pseudo efficiencies for problem LONG-1103.

| $p$ | precond | DACG | | Newton | | Total | | $E_p^{(32)}$ |
|---|---|---|---|---|---|---|---|---|
| | CPU | CPU | MVP | CPU | MVP | CPU | MVP | |
| 32 | 24.87 | 127.47 | 2734 | 346.87 | 5972 | 499.20 | 8706 | |
| 64 | 15.16 | 68.12 | 2734 | 172.24 | 5823 | 255.52 | 8557 | 0.97 |
| 128 | 8.57 | 42.90 | 2734 | 109.31 | 5928 | 160.78 | 8662 | 0.78 |
| 256 | 5.12 | 31.10 | 2734 | 77.41 | 5806 | 113.63 | 8540 | 0.55 |

Table 9: Iterations, timings, and pseudo efficiencies for problem GEO-1438.

| $p$ | precond | DACG | | Newton | | Total | | $E_p^{(32)}$ |
|---|---|---|---|---|---|---|---|---|
| | CPU | CPU | MVP | CPU | MVP | CPU | MVP | |
| 32 | 23.87 | 289.92 | 5002 | 358.20 | 5432 | 671.99 | 10434 | |
| 64 | 13.16 | 139.61 | 5002 | 172.14 | 5439 | 324.91 | 10441 | 1.03 |
| 128 | 7.66 | 80.15 | 5002 | 97.98 | 5427 | 185.79 | 10429 | 0.90 |
| 256 | 5.38 | 49.86 | 5002 | 61.66 | 5431 | 116.90 | 10433 | 0.72 |
| 512 | 6.82 | 37.78 | 5002 | 47.31 | 5429 | 91.91 | 10431 | 0.46 |

## 6. Conclusion

We have proposed a parallel RFSAI-BFGS preconditioner for the acceleration of the PCG method in the approximate solution of the linearized Newton systems in the evaluation of a number of the leftmost eigenpairs of large SPD matrices. We have shown that updating an initial preconditioner (here RFSAI) by a low-rank correction using the BFGS formula, produces significant savings in the total number of iterations of the inner solver (the PCG method). The Newton's algorithm, preconditioner by RFSAI-BFGS, with the aid of a number of initial iterations of DACG to obtain a good initial eigenvector guess, has been completely parallelized and run on the new IBM BlueGene\Q, located at CINECA, Bologna, Italy. The scalability results are very satisfactory, as compared to the size and nonzeros of the problems selected. In particular, for the largest problem, an efficiency of 72% is obtained with 256 processors. Future research is aimed at investigating the relations between the proposed accelerated Newton method and the well-established Jacobi-Davidson method.

## Acknowledgments

The work of the first author has been partially supported by the Spanish Grant MTM2010-18674. The authors also acknowledge the CINECA Iscra Award SPREAD (2012) for the availability of HPC resources and support.

## References

[1] K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.

[2] G. Gambolati, "On time integration of groundwater flow equations by spectral methods," *Water Resources Research*, vol. 29, no. 4, pp. 1257–1267, 1993.

[3] E. Bozzo and M. Franceschet, "Approximations of the generalized inverse of the graph Laplacian matrix," *Internet Mathematics*, vol. 8, no. 4, pp. 456–481, 2012.

[4] V. Simoncini and L. Eldén, "Inexact Rayleigh quotient-type methods for eigenvalue computations," *BIT Numerical Mathematics*, vol. 42, no. 1, pp. 159–182, 2002.

[5] M. A. Freitag and A. Spence, "Rayleigh quotient iteration and simplified Jacobi-Davidson method with preconditioned iterative solves," *Linear Algebra and Its Applications*, vol. 428, no. 8-9, pp. 2049–2060, 2008.

[6] G. L. G. Sleijpen and H. A. Van der Vorst, "Jacobi-Davidson iteration method for linear eigenvalue problems," *SIAM Review*, vol. 42, no. 2, pp. 267–293, 2000.

[7] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, "Accelerated inexact Newton schemes for large systems of nonlinear equations," *SIAM Journal on Scientific Computing*, vol. 19, no. 2, pp. 657–674, 1998.

[8] Y. Notay, "Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem," *Numerical Linear Algebra with Applications*, vol. 9, no. 1, pp. 21–44, 2002.

[9] L. Bergamaschi, R. Bru, A. Martínez, and M. Putti, "Quasi-newton preconditioners for the inexact Newton method," *Electronic Transactions on Numerical Analysis*, vol. 23, pp. 76–87, 2006.

[10] L. Bergamaschi, R. Bru, and A. Martínez, "Low-rank update of preconditioners for the inexact Newton method with SPD Jacobian," *Mathematical and Computer Modelling*, vol. 54, no. 7-8, pp. 1863–1873, 2011.

[11] L. Bergamaschi, R. Bru, A. Martínez, and M. Putti, "Quasi-Newton acceleration of ILU preconditioners for nonlinear two-phase flow equations in porous media," *Advances in Engineering Software*, vol. 46, no. 1, pp. 63–68, 2012.

[12] F. Xue and H. C. Elman, "Convergence analysis of iterative solvers in inexact Rayleigh Quotient iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 877–899, 2009.

[13] S. Bellavia, V. De Simone, D. di Serafino, and B. Morini, "Efficient preconditioner updates for shifted linear systems," *SIAM Journal on Scientific Computing*, vol. 33, no. 4, pp. 1785–1809, 2011.

[14] S. Bellavia, D. Bertaccini, and B. Morini, "Nonsymmetric preconditioner updates in newton-krylov methods for nonlinear

systems," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2595–2619, 2011.

[15] S. Bellavia, V. De Simone, D. Di Serafino, and B. Morini, "A preconditioning framework for sequences of diagonally modified linear systems arising in optimization," *SIAM Journal on Numerical Analysis*, vol. 50, no. 6, pp. 3280–3302, 2012.

[16] L. Bergamaschi, G. Gambolati, and G. Pini, "Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem," *Numerical Linear Algebra with Applications*, vol. 4, no. 2, pp. 69–84, 1997.

[17] L. Bergamaschi and A. Martínez, "Banded target matrices and recursive FSAI for parallel preconditioning," *Numerical Algorithms*, vol. 61, no. 2, pp. 223–241, 2012.

[18] L. Yu. Kolotilina and A. Yu. Yeremin, "Factorized sparse approximate inverse preconditionings. I. Theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 1, pp. 45–58, 1993.

[19] M. Benzi and M. Tůma, "A comparative study of sparse approximate inverse preconditioners," *Applied Numerical Mathematics*, vol. 30, no. 2, pp. 305–340, 1999.

[20] L. Bergamaschi, R. Bru, A. Martínez, J. Mas, and M. Putti, "Low-rank update of preconditioners for the nonlinear Richards equation," *Mathematical and Computer Modelling*, vol. 57, no. 7-8, pp. 1933–1941, 2013.

[21] L. Bergamaschi and A. Martínez, "Efficiently preconditioned inexact newton methods for large symmetric eigenvalue problems," submitted.

[22] C. T. Kelley, *Iterative Methods for Optimization*, vol. 18 of *Frontiers in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 1999.

[23] L. Bergamaschi, G. Pini, and F. Sartoretto, "Approximate inverse preconditioning in the parallel solution of sparse eigenproblems," *Numerical Linear Algebra with Applications*, vol. 7, no. 3, pp. 99–116, 2000.

[24] L. Bergamaschi and M. Putti, "Numerical comparison of iterative eigensolvers for large sparse symmetric positive definite matrices," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 45, pp. 5233–5247, 2002.

[25] L. Bergamaschi, A. Martínez, and G. Pini, "Parallel Rayleigh quotient optimization with FSAI-based preconditioning," *Journal of Applied Mathematics*, vol. 2012, Article ID 872901, 14 pages, 2012.

[26] S. G. Nash and J. Nocedal, "A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization," *SIAM Journal on Optimization*, vol. 1, no. 3, pp. 358–372, 1991.

[27] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[28] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, 1996.

[29] L. Bergamaschi and Á. Martínez, "Parallel acceleration of Krylov solvers by factorized approximate inverse preconditioners," in *Proceedings of the 6th International Conference High Performance Computing for Computational Science (VECPAR '05)*, M. Daydè, Ed., vol. 3402 of *Lecture Notes in Computer Sciences*, pp. 623–636, Springer, Heidelberg, Germany, 2005.

[30] L. Bergamaschi, Á. Martínez, and G. Pini, "An efficient parallel MLPG method for poroelastic models," *Computer Modeling in Engineering & Sciences*, vol. 49, no. 3, pp. 191–216, 2009.

[31] L. Bergamaschi and A. Martínez, "Parallel inexact constraint preconditioners for saddle point problems," in *Proceedings of the 17th International Conference on Parallel Processing*, R. N. E. Jeannot and J. Roman, Eds., vol. 6853 of *Lecture Notes in Computer Sciences*, pp. 78–89, Springer, Bordeaux, France, 2011.

[32] A. Martínez, L. Bergamaschi, M. Caliari, and M. Vianello, "A massively parallel exponential integrator for advection-diffusion models," *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 82–91, 2009.