Journal of Computational and Applied Mathematics 235 (2011) 2647–2661

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Two sides tangential filtering decomposition

Laura Grigori [a], Frédéric Nataf [b], Qiang Niu [c,*]

[a] INRIA Saclay-Ile de France, Bat 490, Universite Paris-Sud 11, 91405, Orsay, France
[b] Laboratoire J.L. Lions, CNRS UMR7598, Universite Paris 6, France
[c] Mathematics and Physics Teaching Centre, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, PR China

ARTICLE INFO

Article history:
Received 25 September 2009
Received in revised form 15 March 2010

Keywords:
Preconditioner
Linear systems of equations
Frequency filtering decomposition

ABSTRACT

In this paper we study a class of preconditioners that satisfy the so-called left and/or right filtering conditions. For practical applications, we use a multiplicative combination of filtering based preconditioners with the classical *ILU*(0) preconditioner, which is known to be efficient. Although the left filtering condition has a more sound theoretical motivation than the right one, extensive tests on convection–diffusion equations with heterogeneous and anisotropic diffusion tensors reveal that satisfying left or right filtering conditions lead to comparable results. On the filtering vector, these numerical tests reveal that $\mathbf{e} = [1, \ldots, 1]^T$ is a reasonable choice, which is effective and can avoid the preprocessing needed in other methods to build the filtering vector. Numerical tests show that the composite preconditioners are rather robust and efficient for these problems with strongly varying coefficients.

1. Introduction

Large sparse linear systems of equations

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

with

$$\mathbf{A} = \begin{bmatrix} D_1 & U_1 & & \\ L_1 & D_2 & \ddots & \\ & \ddots & \ddots & U_{n_x-1} \\ & & L_{n_x-1} & D_{n_x} \end{bmatrix} \in \mathcal{R}^{n\times n}, \quad \mathbf{b} \in \mathcal{R}^n$$

arise in many applications. In present work, we consider the preconditioning techniques for linear systems of form (1) generated from the discretization of the following convection–diffusion problem by a finite volume method on a structured grid in two and three dimensions

$$\begin{aligned} &div(\mathbf{a}(x)u) - div(\kappa(x)\nabla u) = f \quad \text{in } \Omega \\ &u = 0 \quad \text{on } \partial\Omega_D \\ &\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega_N \end{aligned} \tag{2}$$

* Corresponding author. Tel.: +86 7563620656.
E-mail addresses: niuq962@sina.com, kangniu@gmail.com (Q. Niu).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2010.11.016

where $\Omega = [0, 1]^n$ ($n = 2$, or 3), $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$. The vector field $\mathbf{a}$, and the tensor κ are the given coefficients of the partial differential operator. In the 2D case, we have $\partial\Omega_D = [0, 1] \times \{0, 1\}$, and in the 3D case, we have $\partial\Omega_D = [0, 1] \times \{0, 1\} \times [0, 1]$. Due to the discontinuous coefficients in the **PDE** problems and the size of $\mathbf{A}$, preconditioning plays an important role in improving the efficiency of iterative solvers. It is generally recognized that pointwise incomplete factorization preconditioners, e.g. $ILU(0)$ and *MILU* [1,2] are not efficient for such kind of problems. Algebraic multigrid methods [3,4] have been proved successful for a wide class of problems of form (1). Another type of popular preconditioning technique is based on the block incomplete factorization of the coefficient matrix $\mathbf{A}$, and it has been discussed in [5–13].

In this paper we study a class of preconditioners that satisfy the so-called left and/or right filtering conditions. The formal definitions are given as follows.

Definition 1. A preconditioner $\mathbf{M}$ satisfies the right filtering property if

$$\mathbf{Ag} = \mathbf{Mg} \tag{3}$$

where $\mathbf{g}$ is a filtering vector.

Definition 2. A preconditioner $\mathbf{M}$ satisfies the left filtering property if

$$\mathbf{g}^T\mathbf{A} = \mathbf{g}^T\mathbf{M} \tag{4}$$

where $\mathbf{g}$ is a filtering vector.

Right filtering conditions are used to design preconditioners in [13,12,14–17] for tridiagonal block matrices. Left filtering is used in *MILU* [6] and in a very popular preconditioner in the oil industry called nested factorization [18,19]. With an appropriate choice of the starting vector, the Krylov subspace methods preconditioned by a left filtering preconditioner are able to make the residual vector orthogonal with the left filtering vector throughout the iterations. This property is important in oil industry. Several filtering vectors have been considered in the literature: a vector of all ones in *MILU* and nested factorization, sine functions in [13], eigenvectors associated with certain generalized eigenvalue problems in [12,14], adaptive test vectors in [15], and Ritz vectors in [20].

We introduce a new preconditioner that satisfies both left and right filtering conditions. It is based on the tangential filtering preconditioner considered earlier in [20]. The preconditioner constructed by this approach is referred to as the two sides tangential filtering decomposition preconditioner and is denoted as $\mathbf{M}_{lr}$. This difference makes sense only for nonsymmetric problems.

Our main goal is to study the behavior of the preconditioners based on left and/or right filtering. Our study is based on extensive experimental tests that address several issues of interest:

- left and/or right filtering conditions,
- composite preconditioner based on an $ILU(0)$ and filtering preconditioners,
- choice of the filtering vectors.

Our study is based on composite preconditioners since it has been observed in [20] that the convergence of Krylov subspace methods is improved when a filtering preconditioner is combined with an $ILU(0)$ preconditioner, denoted by $\mathbf{M}_{ilu}$ in this paper. This is because $ILU(0)$ preconditioner can be very efficient to make most of the eigenvalues of the preconditioned matrix around 1, whereas this preconditioner has difficulties in removing the eigenvalues that are close to 0 [21,22]. We consider the composite preconditioners implicitly defined by

$$\mathbf{M}_c^{(r)} = (\mathbf{M}^{-1} + \mathbf{M}_{ilu}^{-1} - \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{M}^{-1})^{-1} \tag{5}$$

and

$$\mathbf{M}_c^{(l)} = (\mathbf{M}^{-1} + \mathbf{M}_{ilu}^{-1} - \mathbf{M}^{-1}\mathbf{A}\mathbf{M}_{ilu}^{-1})^{-1}. \tag{6}$$

If $\mathbf{M}$ has the right filtering property, it is shown later in the paper that this property is inherited by $\mathbf{M}_c^{(r)}$. Also, if $\mathbf{M}$ has the left filtering property, then it is also inherited by $\mathbf{M}_c^{(l)}$. Here the subscript c refers to the composite preconditioner, while the superscript $^{(r)}$ ($^{(l)}$) refers to the fact that the corresponding composite preconditioner inherits the right (left) filtering property.

The choice of the filtering vector is an important issue. In our tests we compare the use of the Ritz vector (corresponding to the smallest eigenvalue in modulus) with the use of the vector $\mathbf{e} = [1, 1, \ldots, 1]^T$, and their combination on both left and right filtering conditions.

Our findings are as follows:

- The numerical experiments reveal that on our test problems, there is little difference between using the right combination approach (5) and the left combination approach (6).
- The composite preconditioner based on a combination of $\mathbf{M}_{lr}$ and $\mathbf{M}_{ilu}$ is robust, and converges much faster than a single preconditioner. This is in accordance with the results presented in [20]. Spectrum analysis shows that the composite preconditioners benefit from each of the preconditioners, and can make the spectrum of the preconditioned matrix well clustered at one. Several examples are given to illustrate the spectrum distribution of the preconditioned matrices.

- The numerical results show that for our test problems $\mathbf{e} = [1, 1, \ldots, 1]^T$ is an appropriate choice for the filtering vector. The preconditioner based on this choice is as efficient as the preconditioners obtained when other vectors are used. However other vector choices need a preprocessing phase to construct the filtering preconditioner, and hence an overall more number of iterations. This represents hence an improvement over the preconditioner used for example in [20].

The paper is organized as follows. In Section 2 we briefly review right and left filtering based preconditioners and we introduce the two sides filtering decomposition. The properties of the two sides tangential filtering preconditioner and the composite preconditioners are analyzed in Section 3. Numerical tests are described in Section 4. Finally, we conclude the paper in Section 5, and present some representative spectrum plots of the preconditioned matrices in the Appendix.

2. Tangential filtering decomposition

In this section, we give the definitions of the left and right filtering conditions, and then we introduce the two sides tangential filtering preconditioner that satisfies both the left and right filtering conditions.

We refer to the diagonal matrix constructed from the vector v as $Diag(v)$, and to the block diagonal matrix constructed from the blocks $A_0, A_1, \ldots, A_n$ as $B\,Diag(A_1, A_2, \ldots, A_n)$. The elementwise vector division is denoted by $./$.

2.1. Filtering conditions

Right filtering conditions are used in the design of several preconditioners, as for example in [20,16,17,12,14,15,13]. We describe here more in detail the low frequency tangential filtering preconditioner introduced in [20], which is used later to develop the two sides tangential filtering preconditioner. In this paper, we refer to this preconditioner with the right tangential filtering property (3) as $\mathbf{M}_r$.

Definition 3. The right tangential filtering preconditioner $\mathbf{M}_r$ of $\mathbf{A}$ from (1) is defined by an incomplete block factorization [20]

$$\mathbf{M_r} = \begin{bmatrix} \tilde{T}_1 & & & \\ L_1 & \tilde{T}_2 & & \\ & \ddots & \ddots & \\ & & L_{n_x-1} & \tilde{T}_{n_x} \end{bmatrix} \begin{bmatrix} \tilde{T}_1^{-1} & & & \\ & \tilde{T}_2^{-1} & & \\ & & \ddots & \\ & & & \tilde{T}_{n_x}^{-1} \end{bmatrix} \begin{bmatrix} \tilde{T}_1 & U_1 & & \\ & \tilde{T}_2 & \ddots & \\ & & \ddots & U_{n_x-1} \\ & & & \tilde{T}_{n_x} \end{bmatrix}, \tag{7}$$

and a filtering vector $\mathbf{g} = [g_1^T, \ldots, g_{n_x}^T]^T$. The diagonal blocks are computed as

$$\tilde{T}_i = \begin{cases} D_1, & i = 1, \\ D_i - L_{i-1}(2\beta_{i-1} - \beta_{i-1}\tilde{T}_{i-1}\beta_{i-1})U_{i-1}, & 1 < i \le n_x. \end{cases} \tag{8}$$

where β_{i-1} is a diagonal approximation of $\tilde{T}_{i-1}^{-1}$, $i = 2, \ldots, n_x$, computed as

$$\beta_{i-1} = Diag(\tilde{T}_{i-1}^{-1}U_{i-1}g_i./(U_{i-1}g_i)). \tag{9}$$

Suppose $\mathbf{g}$ is an eigenvector of $\mathbf{A}$ associated with its smallest (in modulus) eigenvalue, and $\mathbf{M}_r$ is the preconditioner constructed by using $\mathbf{g}$ as a right filtering vector. Then

$$\mathbf{M}_r^{-1}\mathbf{A}\mathbf{g} = \mathbf{g},$$

which implies that $\mathbf{g}$ becomes an eigenvector of $\mathbf{M}_r^{-1}\mathbf{A}$ associated with eigenvalue 1. Roughly speaking, the right filtering preconditioner is able to move the “unwanted” eigenvalues to 1, such that the convergence of the preconditioned iterative method can be efficiently accelerated.

An interesting observation is that with an appropriate choice of β_{i-1} in Definition 3, a preconditioner that satisfies the left filtering property can be obtained. The following definition introduces this preconditioner, that we refer to as left tangential filtering preconditioner $\mathbf{M}_l$.

Definition 4. The left tangential filtering preconditioner $\mathbf{M}_l$ of $\mathbf{A}$ from (1) is defined by an incomplete block factorization

$$\mathbf{M}_l = \begin{bmatrix} \tilde{T}_1 & & & \\ L_1 & \tilde{T}_2 & & \\ & \ddots & \ddots & \\ & & L_{n_x-1} & \tilde{T}_{n_x} \end{bmatrix} \begin{bmatrix} \tilde{T}_1^{-1} & & & \\ & \tilde{T}_2^{-1} & & \\ & & \ddots & \\ & & & \tilde{T}_{n_x}^{-1} \end{bmatrix} \begin{bmatrix} \tilde{T}_1 & U_1 & & \\ & \tilde{T}_2 & \ddots & \\ & & \ddots & U_{n_x-1} \\ & & & \tilde{T}_{n_x} \end{bmatrix}, \tag{10}$$

and a filtering vector $\mathbf{g} = [g_1^T, \ldots, g_{n_x}^T]^T$. The diagonal blocks are computed as

$$\tilde{T}_i = \begin{cases} D_1, & i = 1, \\ D_i - L_{i-1}(2\beta_{i-1} - \beta_{i-1}\tilde{T}_{i-1}\beta_{i-1})U_{i-1}, & 1 < i \le n_x. \end{cases} \tag{11}$$

where β_{i-1} is a diagonal approximation to $\tilde{T}_{i-1}^{-1}$, $i = 2, \ldots, n_x$, computed as

$$\beta_{i-1} = Diag(\tilde{T}_{i-1}^{-T} L_{i-1}^T g_i ./ L_{i-1}^T g_i). \tag{12}$$

The left filtering condition is also used implicitly in the *MILU* preconditioner [6] and in a very popular preconditioner in the oil industry called nested factorization [18]. Both preconditioners satisfy the left filtering condition with a special filtering vector **e** whose elements are all 1.

The left filtering condition has a more sound theoretical motivation than the right one. It can be viewed as the constrained residual acceleration method, which is an old technique that has been used in petroleum reservoir simulation since 1970s [23,24]. In contrast to the right filtering preconditioner, the residual constraint is able to eliminate the eigenspace corresponding to the smallest eigenvalues of the iteration matrix. This property has been analyzed in [23]. We remark that the left filtering preconditioner inherits this nice property for well chosen left filtering vectors.

By choosing the starting vector appropriately, Krylov subspace methods preconditioned by a left filtering preconditioner are able to make the residual vector orthogonal with respect to the left filtering vector throughout the iterations, this property has been mentioned in [18,23]. For completeness, we give the following theorem for preconditioned Krylov subspace methods.

Theorem 1. *Taking $\mathbf{M}_l$ with left filtering property (4) as a preconditioner, and using $\mathbf{M}_l^{-1}\mathbf{r}_0$ as a starting vector to construct a Krylov subspace, then we have*

$$\mathbf{g}^T \mathbf{r}_k = 0, \tag{13}$$

where $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$, with $\mathbf{x}_k$ is the kth approximate solution computed by a Krylov subspace method and $x_0 = \mathbf{M}_l^{-1}\mathbf{b}$.

Proof. Suppose left preconditioning is used, then the kth approximate solution $\mathbf{x}_k$ is derived from the combined subspace

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}(\hat{\mathbf{r}}_0, \mathbf{M}_l^{-1}\mathbf{A}\hat{\mathbf{r}}_0, \ldots, (\mathbf{M}_l^{-1}\mathbf{A})^{k-1}\hat{\mathbf{r}}_0) \quad \text{with } \hat{\mathbf{r}}_0 = \mathbf{M}_l^{-1}\mathbf{r}_0.$$

Thus, $\mathbf{x}_k$ takes the form of

$$\mathbf{x}_k = \mathbf{x}_0 + \mathcal{P}_{k-1}(\mathbf{M}_l^{-1}\mathbf{A})\mathbf{M}_l^{-1}\mathbf{r}_0,$$

where $\mathcal{P}_{k-1}(\lambda)$ is a polynomial of degree no more than $k - 1$. Therefore we have

$$\begin{aligned} \mathbf{r}_k &= \mathbf{r}_0 - \mathbf{A}\mathcal{P}_{k-1}(\mathbf{M}_l^{-1}\mathbf{A})\mathbf{M}_l^{-1}\mathbf{r}_0 \\ &= \mathbf{r}_0 - \mathcal{P}_k(\mathbf{A}\mathbf{M}_l^{-1})\mathbf{r}_0. \end{aligned}$$

As

$$\begin{aligned} \mathbf{g}^T\mathbf{r}_0 &= \mathbf{g}^T(\mathbf{b} - \mathbf{A}x_0) \\ &= \mathbf{g}^T(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}b) \\ &= \mathbf{g}^T(\mathbf{M} - \mathbf{A})\mathbf{M}^{-1}b \\ &= 0. \end{aligned}$$

So suppose $\mathcal{P}_k(\lambda) = \sum_{i=1}^k \alpha_i\lambda^i$, then

$$\begin{aligned} \mathbf{g}^T\mathbf{r}_k &= \mathbf{g}^T\mathbf{r}_0 - \mathbf{g}^T\mathcal{P}_k(\mathbf{A}\mathbf{M}_l^{-1})\mathbf{r}_0 \\ &= \sum_{i=1}^k \alpha_i \mathbf{g}^T(I - (\mathbf{A}\mathbf{M}_l^{-1})^i)\mathbf{r}_0 \\ &= \sum_{i=1}^k \alpha_i \mathbf{g}^T(I - \mathbf{A}\mathbf{M}_l^{-1})\mathcal{Q}_{i-1}(\mathbf{A}\mathbf{M}_l^{-1})\mathbf{r}_0 \\ &= \sum_{i=1}^k \alpha_i \mathbf{g}^T(\mathbf{M}_l - \mathbf{A})\mathbf{M}_l^{-1}\mathcal{Q}_{i-1}(\mathbf{A}\mathbf{M}_l^{-1})\mathbf{r}_0 \\ &= 0, \end{aligned}$$

where $\mathcal{Q}_{i-1}(\lambda) = \frac{1-\lambda^i}{1-\lambda}$ is a polynomial of degree $i - 1$, for each $i = 1, \ldots, k$. $\square$

If $\mathbf{g} = [1, \ldots, 1]^T$, then we have $sum(\mathbf{r}_k) = 0$ throughout the iterations, i.e., the sum of the residual components is zero. This property is very important in improving the convergence rate and the acceptability of the interim solutions in reservoir simulations [25,18].

2.2. Two sides tangential filtering decomposition

In this subsection, we introduce the two sides tangential filtering decomposition. One of the motivations behind this preconditioner comes from the following observation. Suppose that we have two approximations β_{i-1} and γ_{i-1} of $\tilde{T}_{i-1}^{-1}$. Assume that the approximations satisfy

$$\|I - \tilde{T}_{i-1}\beta_{i-1}\| \leq \alpha < 1 \quad \text{and} \quad \|I - \tilde{T}_{i-1}\gamma_{i-1}\| \leq \alpha < 1$$

respectively. Then we can combine the two approximations as

$$M_{\beta\gamma} = \beta_{i-1} + \gamma_{i-1} - \gamma_{i-1}\tilde{T}_{i-1}\beta_{i-1}.$$

By the assumptions, it holds that

$$\|I - \tilde{T}_{i-1}M_{\beta\gamma}\| \leq \|I - \tilde{T}_{i-1}\beta_{i-1}\| \, \|I - \tilde{T}_{i-1}\gamma_{i-1}\| \leq \alpha^2.$$

Therefore, $M_{\beta\gamma}$ should be a better approximation of $\tilde{T}_{i-1}^{-1}$ than just using β_{i-1} or γ_{i-1}, if the above assumptions are satisfied.

Definition 5. The two sides tangential filtering preconditioner $\mathbf{M}_{lr}$ is defined by the incomplete block factorization

$$\mathbf{M}_{lr} = \begin{bmatrix} \tilde{T}_1 & & & \\ L_1 & \tilde{T}_2 & & \\ & \ddots & \ddots & \\ & & L_{n_x-1} & \tilde{T}_{n_x} \end{bmatrix} \begin{bmatrix} \tilde{T}_1^{-1} & & & \\ & \tilde{T}_2^{-1} & & \\ & & \ddots & \\ & & & \tilde{T}_{n_x}^{-1} \end{bmatrix} \begin{bmatrix} \tilde{T}_1 & U_1 & & \\ & \tilde{T}_2 & \ddots & \\ & & \ddots & U_{n_x-1} \\ & & & \tilde{T}_{n_x} \end{bmatrix}, \tag{14}$$

along with left filtering vector $\mathbf{f} = [f_1, f_2, \ldots, f_{n_x}]^T$ and right filtering vector $\mathbf{g} = [g_1, g_2, \ldots, g_{n_x}]^T$, where the diagonal blocks $\tilde{T}_i$ are formed by

$$\tilde{T}_i = \begin{cases} D_1, & i = 1, \\ D_i - L_{i-1}(\beta_{i-1} + \gamma_{i-1} - \gamma_{i-1}\tilde{T}_{i-1}\beta_{i-1})U_{i-1}, & 1 < i \leq n_x. \end{cases} \tag{15}$$

The matrices β_{i-1} and γ_{i-1} are diagonal approximations of $\tilde{T}_{i-1}^{-1}$, computed by

$$\beta_{i-1} = Diag(\tilde{T}_{i-1}^{-1}U_{i-1}f_i./U_{i-1}f_i), \tag{16}$$

and

$$\gamma_{i-1} = Diag(\tilde{T}_{i-1}^{-T}L_{i-1}^T g_i./L_{i-1}^T g_i). \tag{17}$$

We should point out that the notations $\tilde{T}_i$ used in (15) are the same with the ones used in formula (8). By setting $\Theta_{i,i-1} = L_{i-1}\gamma_{i-1}$ and $\Theta_{i-1,i} = \beta_{i-1}U_{i-1}$, it is not difficult to find that (15) reduces to the formula

$$\tilde{T}_i = \begin{cases} D_1, & i = 1, \\ D_i - \Theta_{i,i-1}U_{i-1} + L_{i-1}\Theta_{i-1,i} - \Theta_{i,i-1}\tilde{T}_{i-1}\Theta_{i-1,i}, & 1 < i \leq n_x, \end{cases}$$

proposed in [14] for nonsymmetric problems. In practical applications, the approach of constructing the approximations discussed in this paper is quite different from that of [14], where a symmetrization is carried out before determining the transfer matrices $\Theta_{i,j}$. Hence the filtering properties do not exist any longer.

The following lemma and theorem show that the two sides tangential filtering preconditioner $\mathbf{M}_{lr}$ satisfies both the left and the right filtering conditions.

Lemma 1. *If the matrices $(\tilde{T}_i)_{2 \leq i \leq n_x}$ are invertible, then we have*

$$\mathbf{M}_{lr} - \mathbf{A} = B\,Diag(N_1, N_2, \ldots, N_{n_x}),$$

where

$$N_i = \begin{cases} 0, & i = 1, \\ L_{i-1}(\gamma_{i-1}\tilde{T}_{i-1} - I)\tilde{T}_{i-1}^{-1}(\tilde{T}_{i-1}\beta_{i-1} - I)U_{i-1}, & 2 \leq i \leq n_x. \end{cases} \tag{18}$$

Proof. From the induction formula (15), it is easy to see that

$$N_1 = 0,$$

and

$$N_i = L_{i-1}(-\gamma_{i-1} - \beta_{i-1} + \gamma_{i-1}\tilde{T}_{i-1}\beta_{i-1} + \tilde{T}_{i-1}^{-1})U_{i-1}, \quad 2 \le i \le n_x,$$

or written in compact form

$$N_i = L_{i-1}(\gamma_{i-1}\tilde{T}_{i-1} - I)\tilde{T}_{i-1}^{-1}(\tilde{T}_{i-1}\beta_{i-1} - I).$$

Thus (18) holds. □

Theorem 2. *The two sides tangential filtering preconditioner* $\mathbf{M}_{lr}$ *as described in Definition 5 satisfies the left filtering condition on the vector* $\mathbf{g}$ *and the right filtering condition on the vector* $\mathbf{f}$*, that is*

$$\mathbf{g}^T(\mathbf{M}_{lr} - \mathbf{A}) = 0, \tag{19}$$

and

$$(\mathbf{M}_{lr} - \mathbf{A})\mathbf{f} = 0. \tag{20}$$

Proof. From Lemma 1 and the definition of diagonal matrices β_{i-1} and γ_{i-1} in Definition 5, the following two relations

$$(\tilde{T}_{i-1}\beta_{i-1} - I)U_{i-1}f_i = 0$$

and

$$g_i^T L_{i-1}(\gamma_{i-1}\tilde{T}_{i-1} - I) = 0$$

are satisfied. Hence the theorem holds. □

In the rest of this section we give several properties of the two sides tangential filtering preconditioner. We use $\mathbf{A} \succ \mathbf{B}$ ($\mathbf{A} \succeq \mathbf{B}$) to denote that $\mathbf{A} - \mathbf{B}$ is symmetric positive definite (semidefinite). Consider the two sides tangential filtering preconditioner $\mathbf{M}_{lr}$ given in Definition 5. In the following discussions, we always assume $\mathbf{g} = \mathbf{f}$ is used in the symmetric case, then it is obvious that the approximations $\beta_i = \gamma_i$. The following lemma has been established in [20].

Lemma 2. *If* $\mathbf{A} \succ \mathbf{0}$*, then matrices* $\tilde{T}_i \succeq T_i$, $1 \le i \le n_x - 1$*. Moreover,* $\mathbf{M}_{lr} \succ \mathbf{0}$ *and* $\mathbf{M}_{lr} - \mathbf{A} \succeq \mathbf{0}$ *hold.*

Based on Lemma 2 and Refs. [26,3,27], we have the following theorem.

Theorem 3. *If* $\mathbf{A} \succ \mathbf{0}$*, and* $\mathbf{M}_{lr}$ *is the two sides tangential filtering decomposition preconditioner then*

$$\mathbf{A} = \mathbf{M}_{lr} - \mathbf{N}_{lr}, \tag{21}$$

is a P-regular splitting, therefore $\rho(\mathbf{M}_{lr}^{-1}\mathbf{N}_{lr}) < 1$.

2.3. On the choice of the filtering vectors

The choice of the filtering vector is an important issue, and is widely studied in [20,16,17,28,12,14,15]. Generally, the filtering vector should enable the preconditioner to effectively damp the error components in different frequencies. It has been suggested in [12,14,13] that several preconditioners should be constructed by using different types of filtering vectors. Particularly, the sine function

$$(f^j)_k = \sin(\pi\omega_j hk)$$

is considered in [13], where h is the grid size, ω_j is the frequency. The filtering vectors are generalized to eigenvectors associated with a generalized eigenvalue problem in [12,14]. The number of filtering vectors is suggested to be proportional to $\log_2(n)$. Then the final preconditioning process is equivalent to implementing a single preconditioner that is formed by combining these different preconditioners in a multiplicative way. For a special class of model problems, the convergence rate is proven to be independent of the number of unknowns. However, there are some difficult cases on which the preconditioned iterative solver is not efficient. As an improvement, an adaptive filtering approach is considered in [15]. The method uses a sequence of filtering vectors (error approximations) that can be computed adaptively. There are also some inexact filtering decompositions, for example the tangential decomposition [16] and two-frequency decomposition [17], which have the average filtering condition, not the exact one. The methods of using a sequence of filtering preconditioners are appealing, but considerable setup time and memory are needed [29].

In [20], the authors propose a low frequency tangential filtering decomposition, which forms preconditioners with right filtering property. By combining the filtering preconditioner with the classical $ILU(0)$ preconditioner in a multiplicative way, a composite preconditioner is analyzed. The filtering vector is chosen as the Ritz vector corresponding to the lowest eigenvalue of the preconditioned matrix (by *ILU*(0)). The approach has the merit of efficiently smoothing both the high and the low frequency error components. However, a preprocessing is still needed to generate the filtering vector, which causes extra computation time.

In this paper, we recommend to use $\mathbf{e} = [1, 1, \dots, 1]^T$ as both the left and the right filtering vectors. As it will be illustrated by the numerical examples, using **e** as the filtering vector is robust and generally better than other vectors in terms of iterations. Moreover, this choice can save the preprocessing that is needed in other methods to form the filtering vectors. Therefore, the choice for the filtering vector can be much more efficient in terms of total computational cost and solution time.

On the left filtering vector, we believe that using **e** as the filtering vector is especially important. According to the analysis in [25,18,28,23], such kind of left filtering is equivalent to imposing a zero sum constraint on the residual vectors computed by the preconditioned iterative solver. By setting an appropriate initial approximate solution, this constraint ensures the mass conservation property, which is very important for solving linear systems arising from reservoir simulations [25,18,23].

On the right filtering vectors, we have tried other choices, e.g. Ritz vectors. However, this strategy is not as efficient from the view point of computational cost. To further exploit the potential power of the tangential frequency filtering preconditioner, we also test and compare different combination approaches of the left and right filtering vectors, like using **e** as the left filtering vector, and the Ritz vector as right filtering vector, and so on (see the numerical examples in Section 4). It is possible to explore other better choices of the right filtering vectors. However, we believe that the preconditioner using **e** as the filtering vector is well suited for our problems arising from the discretization of structured grids.

3. Analysis of composite preconditioning techniques

It is well known that the *ILU*(0) preconditioner $\mathbf{M}_{ilu}$ is quite efficient in damping the high frequency error components of the coefficient matrix, whereas the low frequency errors are difficult to damp. Therefore, the asymptotic convergence rate of iterative methods are dominated by the low frequency error components such that the asymptotic behavior of the preconditioned matrix with *ILU*(0) is generally not better than that of the original matrix [21,22]. It is proposed in [20] to combine the *ILU*(0) preconditioner with the tangential filtering preconditioner so as to circumvent the inefficiency of a single preconditioner. In this paper, we consider to combine the newly built two sides tangential filtering preconditioner with the *ILU*(0) preconditioner.

Suppose

$$\mathbf{A} = \mathbf{M}_{ilu} - \mathbf{N}_{ilu}$$

is the splitting associated with the *ILU*(0) preconditioner $\mathbf{M}_{ilu}$.

There are two multiplicative approaches to combine the preconditioners $\mathbf{M}_{lr}$ and $\mathbf{M}_{ilu}$,

$$\mathbf{M}_c^{(r)} = (\mathbf{M}_{lr}^{-1} + \mathbf{M}_{ilu}^{-1} - \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{M}_{lr}^{-1})^{-1} \tag{22}$$

and

$$\mathbf{M}_c^{(l)} = (\mathbf{M}_{lr}^{-1} + \mathbf{M}_{ilu}^{-1} - \mathbf{M}_{lr}^{-1}\mathbf{A}\mathbf{M}_{ilu}^{-1})^{-1}. \tag{23}$$

Here the subscript c refers to the composite preconditioner, where the superscript $^{(r)}$ ($^{(l)}$) implies that the corresponding preconditioner has the right (left) filtering property, as is illustrated by the following theorems.

Theorem 4. *The composite preconditioner $\mathbf{M}_c^{(r)}$ inherits the right filtering property (20), that is, if $(\mathbf{M}_{lr} - \mathbf{A})\mathbf{f} = 0$, then*

$$(\mathbf{M}_c^{(r)} - \mathbf{A})\mathbf{f} = 0. \tag{24}$$

Proof. From (20) and (22) we have

$$\begin{aligned} \mathbf{M}_c^{(r)^{-1}}\mathbf{A}\mathbf{f} &= \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{f} + \mathbf{M}_{lr}^{-1}\mathbf{A}\mathbf{f} - \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{M}_{lr}^{-1}\mathbf{A}\mathbf{f} \\ &= \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{f} + \mathbf{f} - \mathbf{M}_{ilu}^{-1}\mathbf{A}\mathbf{f} \\ &= \mathbf{f}, \end{aligned}$$

which is equivalent to $(\mathbf{M}_c^{(r)} - \mathbf{A})\mathbf{f} = 0$. □

Theorem 5. *The composite preconditioner $\mathbf{M}_c^{(l)}$ inherits the left filtering property (19), that is, if $\mathbf{g}^T(\mathbf{M}_{lr} - \mathbf{A}) = 0$, then*

$$\mathbf{g}^T(\mathbf{M}_c^{(l)} - \mathbf{A}) = 0. \tag{25}$$

The proof proceeds similar to that of Theorem 4.

Table 1
Comparison of two combination approaches.

$h = 1/100$	Advection–diffusion	Non-homogeneous	Skyscraper	Convective skyscraper	Anisotropic
$\mathbf{M}_c^{(l)}$	27	26	26	19	18
$\mathbf{M}_c^{(r)}$	27	26	25	19	17

Remarks. (1) When using the tangential filtering preconditioner $\mathbf{M}_r$ proposed in [20] to combine with $ILU(0)$, the composite preconditioner possesses the right filtering property if combination approach (22) is used. However, there is no filtering property if combination approach (23) is used.

(2) For $\mathbf{M}_c^{(l)}$ preconditioned Krylov subspace methods, if the starting vector $\mathbf{x}_0$ is chosen as $\mathbf{x}_0 = (\mathbf{M}_c^{(l)})^{-1}\mathbf{b}$, then the sum of the residual vector $\mathbf{r}_k$ is equal to zero, i.e.,

$$\mathbf{e}^T\mathbf{r}_k = 0.$$

Now we regard composite preconditioners $\mathbf{M}_c^{(r)}$ and $\mathbf{M}_c^{(l)}$ as they are derived from the following splittings of $\mathbf{A}$, respectively.

$$\mathbf{A} = \mathbf{M}_c^{(r)} - \mathbf{N}_c^{(r)}, \qquad \mathbf{A} = \mathbf{M}_c^{(l)} - \mathbf{N}_c^{(l)}. \tag{26}$$

For the corresponding fixed point iteration

$$\mathbf{x}_{k+1} = \mathbf{M}_c^{-1}\mathbf{N}_c\mathbf{x}_k + \mathbf{M}_c^{-1}\mathbf{b}, \tag{27}$$

with $\mathbf{M}_c$ chosen as $\mathbf{M}_c^{(r)}$ or $\mathbf{M}_c^{(l)}$, we have the following proposition which has been analyzed in [3].

Proposition 1. *For the fixed point iteration (27), the usage of $\mathbf{M}_c^{(r)}$ and $\mathbf{M}_c^{(l)}$ as preconditioner leads to the same convergence rate.*

For the fixed point iteration, from Proposition 1 we can see that there is no difference in convergence rate between using the preconditioner $\mathbf{M}_c^{(r)}$ or $\mathbf{M}_c^{(l)}$. For preconditioned Krylov subspace methods, we can also expect that the two combination approaches will produce nearly the same results. This is exactly what is observed in the numerical tests. In Table 1, the **GMRES** method preconditioned by two different combinative preconditioners (22) and (23) are compared by some representative examples generated from the discretization of (2) with five different boundary conditions (see next section of this paper). The tested matrices are two dimensional with mesh size $h = \frac{1}{100}$. From this table we can see that there is at most a difference of one step in the number of iterations.

For a special class of matrices which often arise from discretization of elliptic and parabolic differential equations, the following theorem reveals that the fixed point iteration (27) associated with the composite preconditioner is convergent, and converges faster than just using the $ILU(0)$ preconditioner or the two sides tangential filtering preconditioner $\mathbf{M}_{lr}$. We first recall a useful result which will be used in our proof. It has been established in [30] in a more general operator setting:

Lemma 3 (*Ashby et al. [30]*). *If $\mathbf{A}$ is symmetric positive definite and $\mathbf{G}$ is $\mathbf{A}$-self-adjoint in the sense that $(\mathbf{Gu}, \mathbf{v})_\mathbf{A} = (\mathbf{u}, \mathbf{Gv})_\mathbf{A}$, then*

$$\|\mathbf{G}\|_\mathbf{A} = \rho(\mathbf{G}).$$

Theorem 6. *Assume $\mathbf{A}$ is a symmetric M-matrix, then the fixed point iteration (27) associated with one of the composite preconditioners is convergent, i.e.*

$$\rho(\mathbf{M}_c^{-1}\mathbf{N}_c) \le \rho(\mathbf{M}_{ilu}^{-1}\mathbf{N}_{ilu}) \cdot \rho(\mathbf{M}_{lr}^{-1}\mathbf{N}_{lr}) < 1.$$

Proof. Firstly, for a symmetric M-matrix $\mathbf{A}$, the splitting associated with the $\mathbf{M}_{ilu}$ preconditioner is regular splitting and thus convergent [1,31], i.e. $\rho(\mathbf{M}_{ilu}^{-1}\mathbf{N}_{ilu}) < 1$. Secondly, from the definition of an M-matrix, we have that $\mathbf{A}$ is symmetric positive definite. Therefore, from Theorem 3 we also have $\rho(\mathbf{M}_{lr}^{-1}\mathbf{N}_{lr}) < 1$. Thirdly, as

$$((\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A})\mathbf{u}, \mathbf{v})_\mathbf{A} = (\mathbf{u}, (\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A})\mathbf{v})_\mathbf{A}$$

and

$$((\mathbf{I} - \mathbf{M}_{ilu}^{-1}\mathbf{A})\mathbf{u}, \mathbf{v})_\mathbf{A} = (\mathbf{u}, (\mathbf{I} - \mathbf{M}_{ilu}^{-1}\mathbf{A})\mathbf{v})_\mathbf{A},$$

where $\mathbf{u}, \mathbf{v} \in \mathcal{R}^n$ and $(\mathbf{u}, \mathbf{v})_{\mathbf{A}}$ is the inner product induced by the SPD matrix $\mathbf{A}$. So both $\mathbf{I}-\mathbf{M}_{lr}^{-1}\mathbf{A}$ and $\mathbf{I}-\mathbf{M}_{ilu}^{-1}\mathbf{A}$ are self-adjoint (or symmetric) with respect to the inner product induced by matrix $\mathbf{A}$. Then based on Lemma 3, we have

$$\|\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A}\|_{\mathbf{A}} = \rho(\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A})$$

and

$$\|\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A}\|_{\mathbf{A}} = \rho(\mathbf{I} - \mathbf{M}_{ilu}^{-1}A).$$

Therefore

$$\begin{aligned}
\rho(\mathbf{M}_c^{-1}\mathbf{N}_c) &= \rho(\mathbf{I} - \mathbf{M}_c^{-1}\mathbf{A}) \\
&\le \|\mathbf{I} - \mathbf{M}_c^{-1}\mathbf{A}\|_{\mathbf{A}} \\
&\le \|\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A}\|_{\mathbf{A}} \cdot \|\mathbf{I} - \mathbf{M}_{ilu}^{-1}\mathbf{A}\|_{\mathbf{A}} \\
&= \rho(\mathbf{I} - \mathbf{M}_{lr}^{-1}\mathbf{A}) \cdot \rho(\mathbf{I} - \mathbf{M}_{ilu}^{-1}\mathbf{A}) \\
&= \rho(\mathbf{M}_{ilu}^{-1}\mathbf{N}_{ilu}) \cdot \rho(\mathbf{M}_{lr}^{-1}\mathbf{N}_{lr}) \\
&< 1.
\end{aligned}$$

The proof is complete. □

For a symmetric M-matrix $\mathbf{A}$, both $\lambda(\mathbf{M}_{ilu}^{-1}\mathbf{A})$ and $\lambda(\mathbf{M}_{lr}^{-1}\mathbf{A})$ are in interval $(0, 1]$. Therefore, the following result is an immediate corollary of Theorem 6.

Corollary. *Assume* $\mathbf{A}$ *is a symmetric M-matrix, then*

$$\rho(\mathbf{M}_c^{-1}\mathbf{N}_c) \le |1 - \lambda_{min}(\mathbf{M}_{lr}^{-1}\mathbf{A})| \cdot |1 - \lambda_{min}(\mathbf{M}_{ilu}^{-1}\mathbf{A})|.$$

From the spectrum distribution plots (in the Appendix), it is easy to see that even though sometimes $\lambda_{min}(\mathbf{M}_{lr}^{-1}\mathbf{A})$ and $\lambda_{min}(\mathbf{M}_{ilu}^{-1}\mathbf{A})$ are close to zero, $\lambda_{min}(\mathbf{M}_c^{-1}\mathbf{A})$ can be well separated from zero. This implies that the fixed point iteration associated with the composite preconditioner should be much faster than that of $\mathbf{M}_{ilu}$ or $\mathbf{M}_{lr}$.

4. Numerical tests

In this section, we present numerical results that compare the performance of the preconditioners discussed in this paper. The performance of composite preconditioners is compared with $\mathbf{M}_{ilu}$ and the two sides tangential filtering decomposition preconditioner $\mathbf{M}_{lr}$. Several different approaches of constructing the filtering preconditioner $\mathbf{M}_{lr}$ are considered, the meaning of the notation is described below. As we have illustrated, there is only a small difference between using two composite preconditioners. So the combination approach (5) is always used in our test, if two preconditioners are combined.

$\mathbf{M}_{lr}$: Two sides tangential filtering decomposition preconditioner.
$\mathbf{M}_{ilu}$: the $ILU(0)$ preconditioner.
$\mathbf{M}_c$: Combination of $\mathbf{M}_{ilu}$ with $\mathbf{M}_{lr}$, where $\mathbf{e}$ is used as both the left and the right filtering vectors in constructing $\mathbf{M}_{lr}$.
$\mathbf{M}_{cr1}$: Combination of $\mathbf{M}_{ilu}$ with $\mathbf{M}_r$, where $\mathbf{e}$ is chosen as a filtering vector in constructing $\mathbf{M}_r$.
$\mathbf{M}_{cl1}$: Combination of $\mathbf{M}_{ilu}$ with $\mathbf{M}_l$, where $\mathbf{e}$ is chosen as the filtering vector in constructing $\mathbf{M}_l$.
$\mathbf{M}_{crr}$: Combination of $\mathbf{M}_{ilu}$ with $\mathbf{M}_r$, where a Ritz vector is chosen as the filtering vector in constructing $\mathbf{M}_r$.
$\mathbf{M}_{c1r}$: Combination of $\mathbf{M}_{ilu}$ with $\mathbf{M}_{lr}$, where $\mathbf{e}$ and a Ritz vector are chosen as the left and the right filtering vectors respectively in constructing $\mathbf{M}_{lr}$.

For symmetric problems, the preconditioners $\mathbf{M}_{cr1}$, $\mathbf{M}_{cl1}$ and $\mathbf{M}_c$ are equivalent when the same filtering vector is used. In this case, just $\mathbf{M}_c$ is displayed. The linear systems are solved by the **GMRES** algorithm preconditioned by the previously outlined composite preconditioners. The algorithm is unrestarted and the maximum Krylov subspace is set to be 200. The algorithm is stopped whenever the relative norm $\frac{\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|}{\|\mathbf{b}\|}$ is less than 10^{-12}. The exact solution is generated randomly. Unless special explanations, the initial approximate solution is always chosen such that the sum of the residual vectors be zero throughout all the iterations. In the following tables, *iter* denotes the number of iterations, *error* denotes the infinite norm of the difference between the final approximate solution and the exact solution, We use "—" to denote that the method fails to converge within 200 iterations, and *cpu* to denote the time it takes to construct the preconditioner and to solve the linear systems. We have used 25 steps of GMRES preconditioned by $\mathbf{M}_{ilu}$ to generate the Ritz vector as a filtering vector. All experiments were performed on a windows XP system with Intel Core 2 Quad CPU 2.66 GHz and main memory 2GB using MATLAB 7.0.4 [32].

The considered boundary value problem (2) is discretized on a regular Cartesian grid with a cell-centered finite volume scheme. Full up-winding is used for the convective term in the partial differential equation. The following five different cases are considered.

Table 2
Results for Case I, advection–diffusion problem in two dimensions; nonsymmetric.

$1/h$	$\mathbf{M}_{ilu}$		$\mathbf{M}_{c}$			$\mathbf{M}_{c1r}$			$\mathbf{M}_{crr}$			$\mathbf{M}_{cl1}$			$\mathbf{M}_{cr1}$		
	Iter	*Error*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*
100	110	4.2e−9	27	1.6e−10	1.9	25	1.7e−10	2.2	25	9.9e−11	2.2	26	1.4e−10	1.8	26	1.5e−10	1.8
200	198	1.7e−9	38	3.7e−10	24.4	36	3.1e−10	25.3	35	2.4e−10	24.1	37	3.9e−10	23.3	37	3.7e−10	23.3
300	—	—	45	1.0e−9	109.2	43	9.2e−10	109.9	42	4.9e−10	106.3	45	9.2e−10	106.2	45	8.1e−10	107.1
400	—	—	53	1.1e−9	411.9	51	1.6e−9	406.3	49	6.7e−10	389.5	52	1.0e−9	397.1	52	1.5e−9	395.7

Table 3
Results for Case II, non-homogeneous problems in two dimensions; symmetric.

$1/h$	$\mathbf{M}_{ilu}$		$\mathbf{M}_{c}$			$\mathbf{M}_{c1r}$			$\mathbf{M}_{crr}$		
	Iter	*Error*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*
100	108	3.1e−9	26	1.1e−10	1.8	25	1.6e−10	2.2	25	1.2e−10	2.2
200	190	8.5e−9	37	7.1e−10	24.2	36	3.5e−10	25.3	34	2.9e−10	23.9
300	—	—	46	4.8e−10	109.1	44	7.0e−10	112.2	42	3.1e−10	106.3
400	—	—	52	1.1e−9	406.7	52	1.2e−9	412.8	49	9.1e−10	387.6

Table 4
Results for Case III, skyscrapers problems in two (top) and three (bottom) dimensions; symmetric.

$1/h$	$\mathbf{M}_{ilu}$		$\mathbf{M}_{c}$			$\mathbf{M}_{c1r}$			$\mathbf{M}_{crr}$		
	Iter	*Error*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*
100	–	–	25	3.8e−7	2.0	26	3.0e−7	2.3	38	7.6e−6	3.2
200	–	–	40	1.1e−6	25.1	43	1.1e−6	28.9	49	5.9e−7	32.9
300	–	–	47	4.5e−6	115.5	75	2.4e−6	179.4	55	6.8e−7	133.2
400	–	–	59	6.7e−6	448.3	119	1.8e−6	930.1	193	3.0e−6	1677.5
20	128	4.6e−8	11	1.0e−8	5.2	11	3.5e−9	5.5	13	5.2e−9	5.4
30	199	6.5e−7	14	6.0e−8	68.6	14	6.4e−8	69.7	28	2.2e−10	68.7
40	–	–	15	7.0e−8	477.2	17	5.1e−8	481.3	27	2.3e−9	436.2

Case I. *The advection–diffusion problem with a rotating velocity in two dimensions:*

The tensor κ is the identity, and the velocity is $\mathbf{a} = (2\pi(x_2 - 0.5), 2\pi(x_1 - 0.5))^T$. The uniform grid with $n \times n$ nodes, $n = 100, 200, 300, 400$ are tested respectively. Table 2 displays the results obtained by using different preconditioners.

Case II. *Non-homogeneous problems with large jumps in the coefficients in two dimensions:*

The values of $\mathbf{a}$ are zero. The tensor κ is isotropic and discontinuous. It jumps from the constant value 10^3 in the ring $\frac{1}{2\sqrt{2}} \le |x - c| \le \frac{1}{2}$, $c = (\frac{1}{2}, \frac{1}{2})^T$, to 1 outside. Table 3 displays the results obtained by using different preconditioners.

Case III. *Skyscraper problems:*

The tensor κ is isotropic and discontinuous. The domain contains many zones of high permeability which are isolated from each other. Let $[x]$ denote the integer value of x. In 2D, we have

$$\kappa(x) = \begin{cases} 10^3 * ([10 * x_2] + 1), & \text{if } [10 * x_i] = 0 \bmod (2),\ i = 1, 2, \\ 1, & \text{otherwise,} \end{cases}$$

and in 3D

$$\kappa(x) = \begin{cases} 10^3 * ([10 * x_2] + 1), & \text{if } [10 * x_i] = 0 \bmod (2),\ i = 1, 2, 3, \\ 1, & \text{otherwise.} \end{cases}$$

The parameter $\mathbf{a}$ is zero vector. Table 4 displays the results obtained by using different preconditioners for both the 2D and 3D problems.

Case IV. *Convective skyscraper problems:*

The same with the Skyscraper problems except that the velocity field is changed to be $\mathbf{a} = (1000, 1000, 1000)^T$. The tested results are displayed in Table 5.

Case V. *Anisotropic layers:*

The domain is made of 10 anisotropic layers with jumps of up to four orders of magnitude and an anisotropy ratio of up to 10^3 in each layer. For the 3D problem, the cube is divided into 10 layers parallel to $z = 0$, of size 0.1, in which the coefficients are constant. The coefficient κ_x in the ith layer is given by $v(i)$, the latter being the ith component of the vector

Table 5
Results for Case IV, convective skyscrapers in two (top) and three (bottom) dimensions; nonsymmetric.

$1/h$	$\mathbf{M}_{ilu}$		$\mathbf{M}_c$			$\mathbf{M}_{c1r}$			$\mathbf{M}_{crr}$			$\mathbf{M}_{cl1}$			$\mathbf{M}_{cr1}$		
	Iter	*Error*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*
100	185	3.6e−8	19	2.3e−10	1.4	20	6.0e−10	1.7	26	1.3e−8	2.4	19	4.2e−10	1.4	21	6.7e−9	1.6
200	–	–	26	1.4e−8	18.5	55	4.5e−9	35.8	42	9.9e−8	29.0	26	2.6e−8	18.1	30	8.0e−8	20.5
300	–	–	28	1.2e−7	77.6	54	7.1e−8	135.1	52	6.9e−8	193.2	28	4.5e−8	74.7	37	2.7e−7	198.6
400	–	–	39	1.3e−7	320	95	1.3e−7	4785.6	110	2.3e−7	3652.6	39	1.0e−7	311.2	54	2.5e−7	1077.2
20	66	2.6e−10	6	1.4e−9	4.6	10	4.0e−10	5.4	16	3.4e−10	5.7	9	5.3e−10	4.4	9	4.3e−10	4.4
30	110	2.1e−9	12	5.2e−11	67.7	27	2.3e−10	76.4	37	4.1e−10	73.5	31	1.6e−10	68.9	15	2.3e−10	60.6
40	114	3.5e−9	10	6.2e−11	469.1	14	2.9e−11	477.2	32	1.9e−9	442.6	12	2.1e−11	412.2	13	8.2e−10	413.5

Table 6
Results for Case V, anisotropic layers in two (top) and three (bottom) dimensions; symmetric.

$1/h$	$\mathbf{M}_{ilu}$		$\mathbf{M}_c$			$\mathbf{M}_{c1r}$			$\mathbf{M}_{crr}$		
	Iter	*Error*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*	*Iter*	*Error*	*Cpu*
100	188	5.2e−7	17	2.2e−7	1.4	19	1.8e−7	1.9	21	1.1e−7	2.1
200	–	–	29	1.3e−6	20.0	63	5.0e−6	40.3	35	3.2e−6	25.1
300	–	–	40	9.8e−8	97.9	71	4.4e−6	171.7	41	5.6e−6	106.2
400	–	–	50	7.6e−8	389.1	162	1.6e−5	1351.2	55	9.4e−7	434.8
20	25	1.5e−8	11	1.9e−8	6.0	11	3.7e−8	7.2	14	5.3e−7	6.4
30	33	2.5e−7	12	7.8e−8	72.4	18	1.1e−7	89.6	18	1.0e−7	81.2
40	40	1.8e−7	12	2.1e−7	572.3	33	4.1e−6	636.1	22	1.3e−7	560.3

$v = [\alpha, \beta, \alpha, \beta, \alpha, \beta, \gamma, \alpha, \alpha]$, where $\alpha = 1$, $\beta = 10^2$ and $\gamma = 10^4$. We have $\kappa_y = 10\kappa_x$ and $\kappa_z = 1000\kappa_x$. The velocity field is zero. Numerical results are shown in Table 6.

From Tables 3–6 we can see that $\mathbf{M}_c$, $\mathbf{M}_{cl1}$ and $\mathbf{M}_{cr1}$ lead to similar results for 2D problems. For the 3D problems, $\mathbf{M}_c$ is faster than $\mathbf{M}_{cl1}$ and $\mathbf{M}_{cr1}$ in terms of iteration numbers. For the symmetric problems, $\mathbf{M}_c$, $\mathbf{M}_{cr1}$ and $\mathbf{M}_{cl1}$ are equivalent, and they are more robust than $\mathbf{M}_{crr}$ and $\mathbf{M}_{c1r}$. In conclusion, the numerical result illustrated that:

- Using **e** as a filtering vector is more robust than using a Ritz vector as the filtering vector.
- For the nonsymmetric cases, the left filtering preconditioner leads to comparable results to that of right filtering preconditioner for the convection–diffusion equations with heterogeneous and anisotropic diffusion tensors.
- For 3D nonsymmetric problems, the two sides filtering preconditioner leads to the smallest iteration numbers. However, the two sides filtering preconditioner is not the fastest in terms of cpu time. This is due to the fact that two local approximate matrices (16) and (17) have to be generated during the construction of the preconditioner, and this requires more computation than constructing the one side filtering preconditioner.

Performance of the restarted **GMRES** method [2] and the **BiCGStab** method [33] is examined on four represent matrices from our previous examples. The maximum subspace dimension for the **GMRES** method is set to be 20, and the algorithm is stopped whenever the relative norm $\frac{\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|}{\|\mathbf{b}\|}$ is less than 10^{-12}. In Figs. 1–2, we depict the convergence curves by the preconditioned **GMRES**(**20**) and **BiCGStab**. From these figures we can see that both methods converge very fast preconditioned with the $\mathbf{M}_c$ preconditioner, compared with the **GMRES**(**20**) method, the **BiCGStab** method has better convergence behavior in terms of iteration numbers.

5. Conclusions

In this paper, we have discussed the left, right, and two sides tangential filtering decompositions. The filtering preconditioner constructed by the introduced decomposition is combined with the classical $ILU(0)$ preconditioner in multiplicative ways. The composite preconditioners are very efficient in damping the high and low frequency modes, and thus perform very well for the block tridiagonal linear systems arising from the discretization of **PDE** problems on Cartesian grids. On the filtering vector, we adopt **e** as the filtering vector in this paper. There are several advantages of this choice. Firstly, it is as efficient as other vector choices, and the preprocessing that is needed to construct the filtering preconditioner can be saved, secondly, using **e** as the left filtering vector is able to enable the zero material balance error all throughout the iterations, which is important to improve the convergence.

Acknowledgements

L. Grigori and F. Nataf are supported by French National Research Fund Agency (ANR) through COSINUS program (project PETAL no ANR-08-COSI-009). Q. Niu is supported by XJTLU research fund and part of his work was performed during his visit at INRIA Sacley funded by China Scholarship Council.

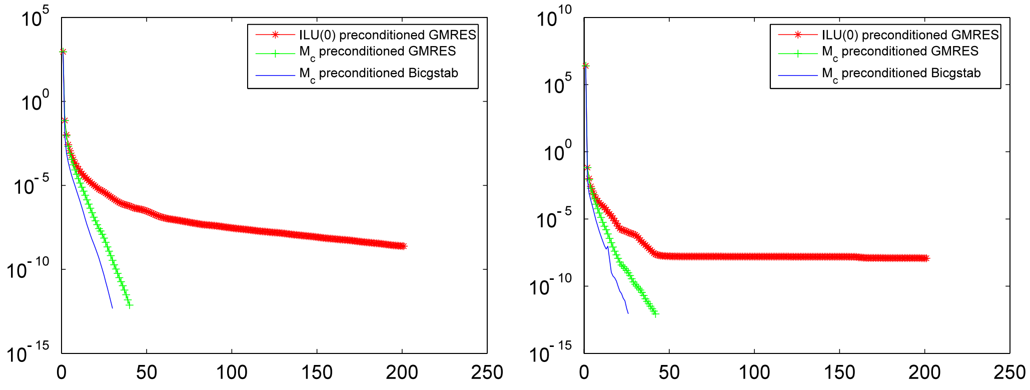

Fig. 1. Convergence curves of preconditioned **GMRES**(**20**) and **BiCGStab** by $\mathbf{M}_c$. Left: Non-homogeneous 200 × 200, right: Skyscraper 200 × 200.

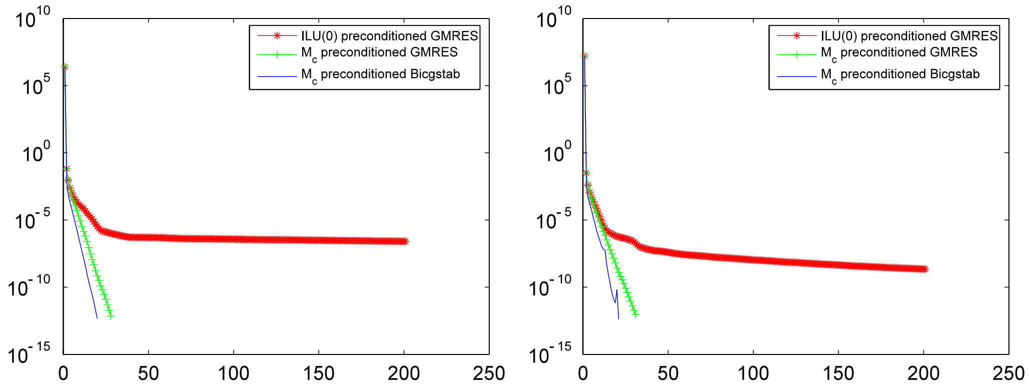

Fig. 2. Convergence curves of preconditioned **GMRES**(**20**) and **BiCGStab** by $\mathbf{M}_c$. Left: Convective Skyscraper 200 × 200, right: Anisotropic layers 200 × 200.

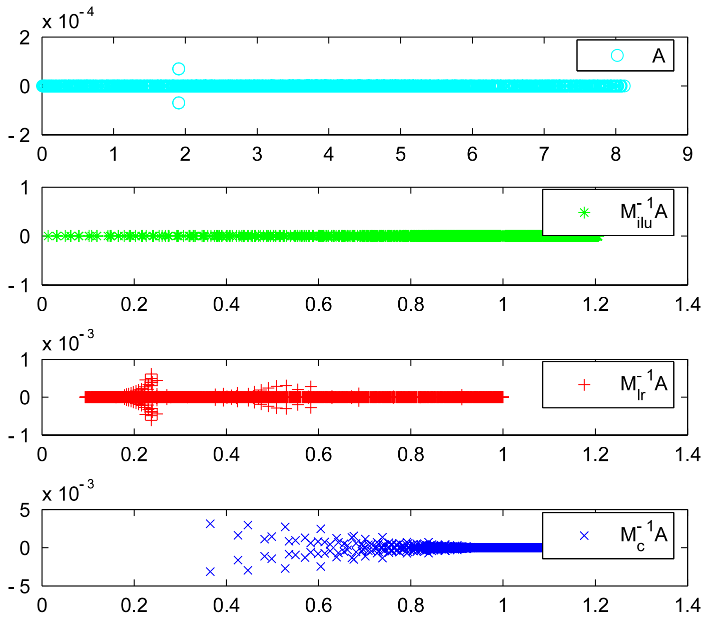

Fig. 3. Spectrum distribution of the preconditioned advection–diffusion 50 × 50 matrix.

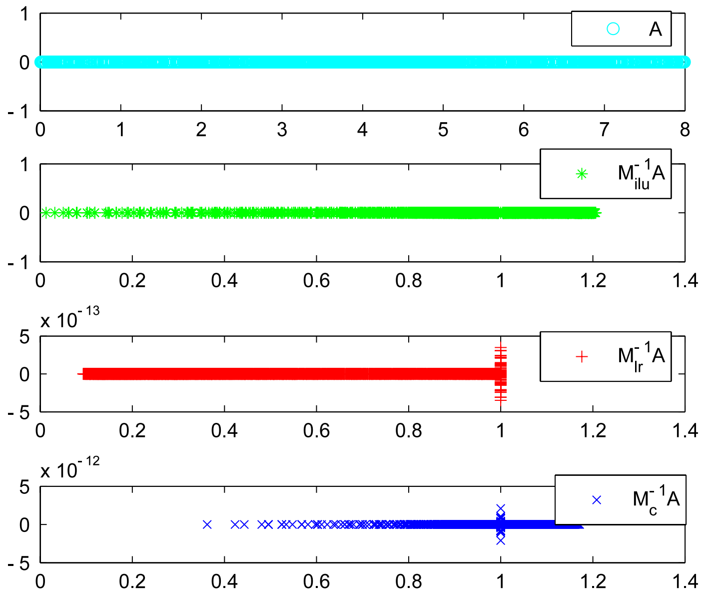

Fig. 4. Spectrum distribution of the preconditioned non-homogeneous 50 × 50 matrix.

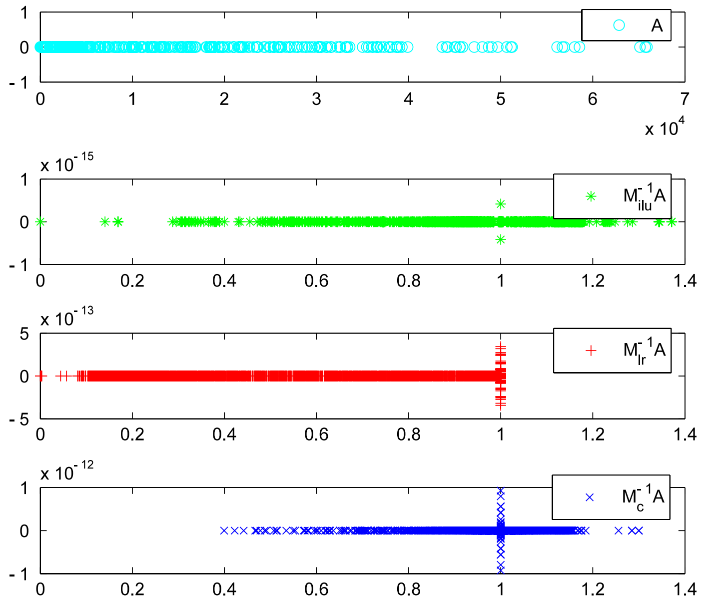

Fig. 5. Spectrum distribution of the preconditioned skyscraper 50 × 50 matrix.

Appendix

The eigenvalue distribution of the preconditioned matrix is plotted from Figs. 3–7. The notations used in the figures are as follows:

A: the coefficient matrix.

$\mathbf{M}_{ilu}^{-1}\mathbf{A}$: the preconditioned matrix by $ILU(0)$ preconditioner.

$\mathbf{M}_{lr}^{-1}\mathbf{A}$: the preconditioned matrix by two sides filtering preconditioner proposed in this paper.

$\mathbf{M}_{c}^{-1}\mathbf{A}$: the preconditioned matrix by the combination preconditioner (23) (the same as using (22)).

From these figures we can see that the composite preconditioners tend to make the spectrum clustered at 1. In the symmetric case, complex eigenvalues appear due to the nonsymmetric composite preconditioner, but their imaginary parts are usually very small.

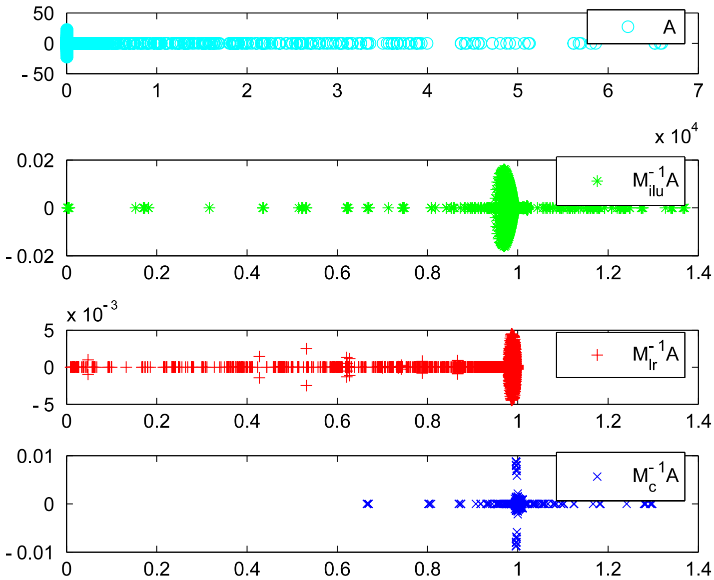

Fig. 6. Spectrum distribution of the preconditioned convective skyscraper 50 × 50 matrix.

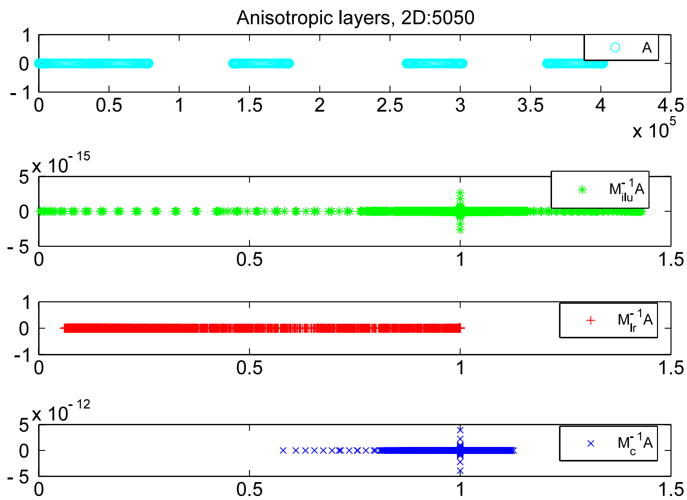

Fig. 7. Spectrum distribution of the preconditioned anisotropic layers 50 × 50 matrix.

References

[1] J.A. Meijerink, H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is symmetric *M*-matrix, Math. Comput. 137 (1977) 148–162.
[2] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, MA, 1996.
[3] W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, Springer, New York, 1994.
[4] J.W. Ruge, K. Stüben, Algebraic Multigrid (AMG), in: Multigrid Methods, Frontiers in Applied Mathematics, vol. 3, SIAM, Philadephia, PA, 1987, pp. 73–130.
[5] O. Axelsson, Iterative Solution Methods, Cambridge University Press, New York, 1994.
[6] O. Axelsson, L. Kolotilina, Diagonally compensated reduction and related preconditioning methods, Numer. Linear Algebra Appl. 1 (1994) 155–177.
[7] O. Axelsson, H. Lu, On the eigenvalue estimates for block incomplete factorization methods, SIAM. J. Matrix Anal. Appl. 16 (1995) 1074–1085.
[8] O. Axelsson, G. Lindskog, On the eigenvalue distribution of a class of preconditioning methods, Numer. Math. 48 (1986) 479–498.
[9] Z.-Z. Bai, G.-Q. Li, L.-Z. Lu, Combinative preconditioners of modified incomplete Cholesky factorization and Sherman–Morrison–Woodbury update for self-adjoint elliptic Dirichlet-periodic boundary value problems, J. Comput. Math. 22 (2004) 833–856.
[10] T.F. Chan, P.S. Vassilevski, A framework for block ILU factorizations using block-size reduction, Math. Comput. 64 (1995) 129–156.
[11] P.S. Vassilevski, Multilevel Block Factorization Preconditioners, Matrix-based Analysis and Algorithms for Solving Finite Element Equations, Springer, New York, 2008.
[12] C. Wagner, Tangential frequency filtering decompositions for symmetric matrices, Numer. Math. 78 (1997) 119–142.
[13] G. Wittum, Filternde Zerlegungen, in: Schnelle Löser für große Gleichungssysteme, Teubner Skripten zur Numerik, Band 1, Teubner-Verlag, Stuttgart, 1992.
[14] C. Wagner, Tangential frequency filtering decompositions for unsymmetric matrices, Numer. Math. 78 (1997) 143–163.
[15] C. Wagner, G. Wittum, Adaptive filtering, Numer. Math. 78 (1997) 305–382.

[16] A. Buzdin, Tangential decomposition, Computing 61 (1998) 257–276.
[17] A. Buzdin, G. Wittum, Two-frequency decomposition, Numer. Math. 97 (2004) 269–295.
[18] J.R. Appleyard, I.M. Cheshire, Nested Factorization, SPE 12264, presented at the Seventh SPE Symposium on Reservoir Simulation, San Francisco, 1983.
[19] P. Kumar, L. Grigori, F. Nataf, Q. Niu, Combinative Preconditioning Based on Relaxed Nested Factorization and Tangential Filtering preconditioner, INRIA TR 6955.
[20] Y. Achdou, F. Nataf, Low frequency tangential filtering decomposition, Numer. Linear Algebra Appl. 14 (2007) 129–147.
[21] T.F. Chan, Fourier analysis of relaxed incomplete factorization preconditioners, SIAM J. Sci. Comput. 12 (1991) 668–690.
[22] T.F. Chan, J.M. Donato, Fourier analysis of incomplete factorization preconditioners for three-dimensional anisotropic problems, SIAM J. Sci. Stat. Comput. 13 (1992) 319–338.
[23] J.R. Wallis, R.P. Kendall, T.E. Little, Constraint residual acceleration of conjugate residual methods, SPE 13536, presented at the SPE 1985, Reservoir Simulation Symposium held in Dallas, Texas, 1985.
[24] J.R. Wallis, A method for improving line successive overrelaxation in anisotropic problems—a theoretical analysis, Trans. Sco. Pet. Engr. 255 (1973) 105–118.
[25] J.R. Appleyard, Proof that Simple Colsum Modified ILU Factorization of Matrices Arising in Fluid Flow Problems are not Singular, by John Appleyard B.A., Ph.D., Polyhedron Software Ltd., May 2004.
[26] Z.-Z. Bai, M. Ng, Preconditioners for nonsymmetric block-toeplitz-like-plus-diagonal linear system, Numer. Math. 96 (2003) 197–220.
[27] J.M. Ortega, Numerical Analysis—A Second Course, Academic, New York, 1972.
[28] Q. Niu, L. Grigori, P. Kumar, F. Nataf, Modified tangential frequency filtering decomposition and its fourier analysis, Numer. Math. 116 (2010) 123–148.
[29] Y. Achdou, F. Nataf, An iterated tangential filtering decomposition, Numer. Linear Algebra Appl. 10 (2003) 511–539.
[30] S.F. Ashby, M.J. Holst, T.A. Manteuffel, P.E. Saylor, The role of inner product in stopping criteria for conjugate gradient iterations, BIT 41 (2001) 26–52.
[31] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[32] The MathWorks, Inc. MATLAB 7, September 2004.
[33] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi–CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.