

## An Efficient Algorithm to Construct an Orthonormal Basis for the Extended Krylov Subspace

Akira Imakura\*

*Graduate School of Systems and Information Engineering, University of Tsukuba,  
1-1-1, Tennodai Tsukuba-city, Ibaraki 305-8573, Japan.*

*Received 24 April 2013; Accepted (in revised version) 2 June 2014*

*Available online 11 July 2014*

---

**Abstract.** Subspace projection methods based on the Krylov subspace using powers of a matrix  $A$  have often been standard for solving large matrix computations in many areas of application. Recently, projection methods based on the extended Krylov subspace using powers of  $A$  and  $A^{-1}$  have attracted attention, particularly for functions of a matrix times a vector and matrix equations. In this article, we propose an efficient algorithm for constructing an orthonormal basis for the extended Krylov subspace. Numerical experiments indicate that this algorithm has less computational cost and approximately the same accuracy as the traditional algorithm.

**AMS subject classifications:** 65F10, 65F25, 65F30, 15A06

**Key words:** Extended Krylov subspace, Krylov subspace, subspace projection methods, orthonormal basis, linear systems.

---

### 1. Introduction

In this article, we investigate relevant subspaces and algorithms for constructing orthonormal bases used in subspace projection methods. These methods are commonly used in large matrix computations associated with linear systems  $Ax = b$ , eigenvalue problems  $Ax = \lambda x$ , functions of a matrix times a vector  $f(A)b$  and matrix equations  $AX + XB + C = O$ . Subspace projection methods reduce large dimensional matrix computations into smaller dimensional matrix computations, in a wide variety of large scale simulations from several areas of application.

Subspace projection methods involve constructing a given type of subspace, and then computing approximate solutions using the constructed subspace. The efficiency of a subspace projection method is largely dependent on the subspace used. There are two conditions to achieve high performance: 1) the solution of the target problem must be well

---

\*Corresponding author. *Email address:* imakura@cs.tsukuba.ac.jp (A. Imakura)

approximated on a small dimensional subspace; and 2) the efficient construction of an orthonormal basis should be possible.

One of the most successful subspaces used in subspace projection methods where  $A \in \mathbb{C}^{n \times n}$ ,  $\mathbf{b} \in \mathbb{C}^n$  is the Krylov subspace, which involves powers of the matrix  $A$  as follows:

$$\mathbf{K}_m(A, \mathbf{b}) := \text{span}\{\mathbf{b}, A\mathbf{b}, \dots, A^{m-1}\mathbf{b}\}. \quad (1.1)$$

State-of-the-art algorithms for solving linear systems and eigenvalue problems are largely based on the Krylov subspace (1.1), which has also been actively studied for the more complicated matrix computations arising with functions of a matrix times a vector and matrix equations — e.g. see. Refs. [7, 12, 13].

Druskin & Knizhnerman [2] proposed an alternative to the Krylov subspace — viz. the extended Krylov subspace, involving not only powers of  $A$  but also of  $A^{-1}$  and defined by

$$\mathbf{EK}_m(A, \mathbf{b}) := \text{span}\{\mathbf{b}, A^{-1}\mathbf{b}, A\mathbf{b}, A^{-2}\mathbf{b}, \dots, A^{m-1}\mathbf{b}, A^{-m}\mathbf{b}\}, \quad (1.2)$$

where  $A$  is assumed to be nonsingular. This extended Krylov subspace has since received much attention in computations for functions of a matrix times a vector and matrix equations, because it can generally approximate solutions with higher accuracy than the Krylov subspace [5, 6, 8, 15, 16]. The extended Krylov subspace generally shows good convergence behaviour. However, multiple linear systems must be solved in order to construct its orthonormal basis, and this is the most time consuming step. Here we analyse characteristics of the linear systems that arise in the construction of the basis, and then propose a novel and more efficient algorithm for constructing an orthonormal basis of the extended Krylov subspace. The efficiency of our proposed algorithm is also evaluated in a series of numerical experiments. Throughout this article,  $\mathbf{V}$  and  $\mathbf{W}$  are subspaces and  $\mathbf{V} + \mathbf{W}$  is their sum — i.e.  $\mathbf{V} + \mathbf{W} := \{\mathbf{v} + \mathbf{w} | \mathbf{v} \in \mathbf{V}, \mathbf{w} \in \mathbf{W}\}$ . The MATLAB colon notation is used, where for example  $A(:, i)$  denotes the  $i$ -th column of a matrix  $A$ .

In Section 2, we briefly introduce the extended Krylov subspace and basic algorithm concepts used in constructing its orthonormal basis. In Section 3, we propose an efficient algorithm for constructing the basis of the extended Krylov subspace via an analysis of the characteristics of the arising linear systems. In Section 4, we evaluate the efficiency of our proposed algorithm in a series of numerical experiments. Our conclusions are summarized in Section 5.

## 2. The Extended Krylov Subspace and its Orthonormal Basis

As shown in Algorithm 2.1 below, subspace projection methods for large matrix computations consist of two steps: 1) constructing a subspace; and 2) computing an approximate solution. It is notable that the dimension of the subspace  $\tilde{k} := \dim \mathbf{L}_k$  is not always equivalent to the number of iterations  $k$ .

When the target problem has a vector solution (as is the case for linear systems, eigenvalue problems, and functions of a matrix times a vector), the approximate solutions

**Algorithm 2.1** The subspace projection method

- 
- 1: **for**  $k = 1, 2, \dots$ , until convergence **do**
  - 2:   Construct the  $k$ th subspace  $\mathbf{L}_k$  such that  $\mathbf{L}_{k-1} \subset \mathbf{L}_k$ , and its basis vectors  $V_k$ .
  - 3:   Compute the  $k$ th approximate solution based on  $V_k$ .
  - 4: **end for**
- 

$\mathbf{x}_k \in \mathbb{C}^n$  obtained from the subspace projection methods are described by linear combination of the basis vectors — i.e.  $\mathbf{x}_k = V_k \mathbf{y}_k$ , where  $V_k \in \mathbb{C}^{n \times \tilde{k}}$  is the matrix with its columns the basis vectors and  $\mathbf{y}_k \in \mathbb{C}^{\tilde{k}}$  the vector with the corresponding coefficients of the linear combination as its entries. On the other hand, when the problem has a square matrix solution (as with matrix equations), the approximate solutions  $X_k$  are generally constructed from low rank matrices — i.e.  $X_k = V_k Y_k V_k^H$  involving a matrix coefficient  $Y_k \in \mathbb{C}^{\tilde{k} \times \tilde{k}}$ . Incidentally, the vectors  $\mathbf{y}_k$  and matrices  $Y_k$  are usually determined by some kind of minimisation or Galerkin-type condition. In the following subsections, we briefly describe the extended Krylov subspace and then present the algorithms traditionally used for constructing the orthonormal basis.

### 2.1. Brief introduction to the extended Krylov subspace

The extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  defined by (1.2) can be regarded as the sum of the two Krylov subspaces involving  $A$  and  $A^{-1}$ , respectively — i.e.

$$\mathbf{EK}_m(A, \mathbf{b}) = \mathbf{K}_m(A, \mathbf{b}) + \mathbf{K}_m(A^{-1}, A^{-1}\mathbf{b}).$$

The extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  satisfies

$$\mathbf{EK}_{m-1}(A, \mathbf{b}) \subset \mathbf{EK}_m(A, \mathbf{b})$$

and  $\dim \mathbf{EK}_m(A, \mathbf{b}) \leq 2m$ . The enhanced efficiency of the extended Krylov subspace for various matrix computations has been widely reported [2, 5, 6, 8, 15, 16], and detailed convergence analyses can be found in Refs. [2, 8, 9]. Since the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  consists of powers of both  $A$  and  $A^{-1}$ , it can well approximate both the inner and outer spectrum, which implicitly leads to the better convergence in the extended Krylov subspace than for the traditional Krylov subspaces  $\mathbf{K}_m(A, \mathbf{b})$  or  $\mathbf{K}_m(A^{-1}, A^{-1}\mathbf{b})$  when solving large matrix computations.

### 2.2. Traditional algorithms to construct the orthonormal basis

Druskin & Knizhnerman [2] introduced an algorithm to construct the orthonormal basis of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  in the case of Hermitian matrices  $A$ . Simoncini [15] extended and improved this algorithm, to provide what is now regarded as the standard algorithm for general matrices.

**Algorithm 2.2** The Arnoldi/Lanczos procedure**Input:**  $A \in \mathbb{C}^{n \times n}$  and  $\mathbf{b} \in \mathbb{C}^n$ **Output:** The orthonormal basis  $V_m \in \mathbb{C}^{n \times m}$  of  $\mathbf{K}_m(A, \mathbf{b})$  s.t.  $V_m^H V_m = I$ 

- 1: Set  $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|_2, V_1 = \mathbf{v}_1$
- 2: **for**  $j = 1, 2, \dots, m-1$  **do**
- 3:    $\mathbf{w}_j = A\mathbf{v}_j$
- 4:    $\widehat{\mathbf{w}}_j \leftarrow$  orthogonalise  $\mathbf{w}_j$  w.r.t.  $V_j$
- 5:    $\mathbf{v}_{j+1} = \widehat{\mathbf{w}}_j/\|\widehat{\mathbf{w}}_j\|_2$
- 6:    $V_{j+1} = [V_j, \mathbf{v}_{j+1}]$
- 7: **end for**

**Algorithm 2.3** Simoncini's algorithm [15]**Input:**  $A \in \mathbb{C}^{n \times n}$  and  $\mathbf{b} \in \mathbb{C}^n$ **Output:** The orthonormal basis  $\mathcal{V}_m \in \mathbb{C}^{n \times 2m}$  of  $\mathbf{EK}_m(A, \mathbf{b})$  s.t.  $\mathcal{V}_m^H \mathcal{V}_m = I$ 

- 1: Set  $W_0(:, 1) = \mathbf{b}, W_0(:, 2) = A^{-1}\mathbf{b}$ .
- 2: Set  $V_1 = \text{orth}(W_0), \mathcal{V}_1 = V_1$
- 3: **for**  $j = 1, 2, \dots, m-1$  **do**
- 4:   Compute  $W_j(:, 1) = AV_j(:, 1)$  and  $W_j(:, 2) = A^{-1}V_j(:, 2)$
- 5:    $\widehat{W}_j \leftarrow$  orthogonalise  $W_j$  w.r.t.  $\mathcal{V}_j$
- 6:    $V_{j+1} = \text{orth}(\widehat{W}_j)$
- 7:    $\mathcal{V}_{j+1} = [\mathcal{V}_j, V_{j+1}]$
- 8: **end for**

The matrix  $V_m \in \mathbb{C}^{n \times m}$ , with columns the orthonormal basis of the Krylov subspace  $\mathbf{K}_m(A, \mathbf{b})$ , can be constructed by the Arnoldi/Lanczos procedure (Algorithm 2.2). The orthogonalisation in Step 4 of Algorithm 2.2 is undertaken by short-term or long-term recurrence for the Hermitian or non-Hermitian matrix  $A$ , and generally based on the Gram-Schmidt method — e.g. see Ref. [12]. On the other hand, if  $\mathcal{V}_m \in \mathbb{C}^{n \times 2m}$  denotes the matrix with columns the orthonormal basis of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$ , then Simoncini's algorithm constructs  $\mathcal{V}_m$  by the block version of the Arnoldi/Lanczos type procedure with two initial vectors  $\mathbf{b}$  and  $A^{-1}\mathbf{b}$  (Algorithm 2.3). Here the function “orth” in Algorithm 2.3 denotes the orthogonalisation of the matrix.

The computation of  $W_0(:, 2) = A^{-1}\mathbf{b}$  and  $W_j(:, 2) = A^{-1}V_j(:, 2)$  in Algorithm 2.3 can be undertaken by solving linear systems of form

$$AW_0(:, 2) = \mathbf{b} \quad \text{and} \quad AW_j(:, 2) = V_j(:, 2), \quad j = 1, 2, \dots, m-1, \quad (2.1)$$

usually by a direct method for a small and dense  $A$  and by a (preconditioned) Krylov subspace method for a large and sparse  $A$  [15].

### 3. An Efficient Algorithm to Construct the Orthonormal Basis

As mentioned in Section 2, Simoncini's algorithm (Algorithm 2.3) requires that we solve  $m$  linear systems (2.1) sequentially in order to construct the orthonormal basis  $\mathcal{V}_m$  of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$ . When the matrix  $A$  is large and sparse, we must construct  $m$  different Krylov subspaces and their bases in order to solve  $m$  different linear systems (2.1), the most time consuming aspect. In this section, we first analyse the characteristics of the linear systems (2.1) that arise when constructing the basis of the extended Krylov subspace. A novel algorithm is then proposed to construct the orthonormal basis, by projecting the approximate solutions of all linear systems (2.1) onto the same subspace.

#### 3.1. Analysis of characteristics of the extended Krylov subspace

A basic property of the Krylov subspace is given in the following proposition [12, 17].

**Proposition 3.1.** *Let  $d := d(A, \mathbf{b})$  be the grade of  $\mathbf{b}$  with respect to  $A$  such that*

$$d := d(A, \mathbf{b}) := \min\{k | P_k(A)\mathbf{b} = \mathbf{0}, P_k(\lambda) \in \mathbb{P}_k, P_k(0) = 1\},$$

where  $\mathbb{P}_k$  denotes set of  $k$ -degree polynomials. Then  $\mathbf{K}_d(A, \mathbf{b})$  is a (minimum)  $A$ -invariant subspace, and

$$A\mathbf{K}_d(A, \mathbf{b}) = \mathbf{K}_d(A, \mathbf{b}) = A^{-1}\mathbf{K}_d(A, \mathbf{b}) \quad (3.1)$$

is satisfied. Also, if  $V_d \in \mathbb{C}^{n \times d}$  is the matrix with its columns the orthonormal basis of  $\mathbf{K}_d(A, \mathbf{b})$ , then

$$V_d^H A V_d = H_d, \quad A V_d = V_d H_d,$$

is the matrix form of the Arnoldi procedure.

From Proposition 3.1, characteristics of the linear systems (2.1) that arise when constructing the basis of the extended Krylov subspace are identified as follows.

**Proposition 3.2.** *If  $d := d(A, \mathbf{b})$  be the grade of  $\mathbf{b}$  with respect to  $A$ , then*

$$\mathbf{EK}_m(A, \mathbf{b}) \subseteq \mathbf{K}_d(A, \mathbf{b})$$

is satisfied.

*Proof.* From (3.1) in Proposition 3.1, we have

$$A^k \mathbf{b} \in A^k \mathbf{K}_d(A, \mathbf{b}) = \mathbf{K}_d(A, \mathbf{b}), \quad \forall k \in \mathbb{Z}.$$

hence we have the result from the definition of the extended Krylov subspace.  $\square$

From Proposition 3.2, all solutions  $W_j(:, 2), j = 0, 1, \dots, m - 1$  for the linear systems (2.1) can be obtained mathematically by projection onto the same Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$ . Moreover, based on Proposition 3.2 we also have the following theorem for the relationship between the orthonormal basis  $\mathcal{V}_m$  of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  and the orthonormal basis  $V_d$  of the Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$ .

**Theorem 3.1.** Let  $d := d(A, \mathbf{b})$  be the grade of  $\mathbf{b}$  with respect to  $A$ , and let  $V_d \in \mathbb{C}^{n \times d}$  be the matrix with its columns the orthonormal basis of the Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$ . Then

$$\mathbf{EK}_m(A, \mathbf{b}) = V_d \mathbf{EK}_m(H_d, \beta \mathbf{e}_1),$$

where

$$H_d = V_d^H A V_d \in \mathbb{C}^{d \times d}, \quad \mathbf{e}_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^d, \quad \beta = \|\mathbf{b}\|_2.$$

*Proof.* From Proposition 3.2,  $A^k \mathbf{b}$  for  $k = -m, -(m-1), \dots, m-1$  can be written as the linear combination of the orthonormal basis of the Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$ :

$$A^k \mathbf{b} = V_d \mathbf{y}_d^{(k)}, \quad k = -m, -(m-1), \dots, m-1,$$

where  $\mathbf{y}_d^{(k)} \in \mathbb{C}^d$ . Now from  $\mathbf{b} = V_d(\beta \mathbf{e}_1)$  and  $AV_d = V_d H_d$  the vector  $\mathbf{y}_d^{(k)}$  can be rewritten as

$$\mathbf{y}_d^{(k)} = V_d^H A^k V_d(\beta \mathbf{e}_1) = H_d^k(\beta \mathbf{e}_1).$$

Consequently,

$$\begin{aligned} \mathbf{EK}_m(A, \mathbf{b}) &= \text{span} \{A^{-m} \mathbf{b}, A^{-(m-1)} \mathbf{b}, \dots, A^{m-1} \mathbf{b}\} \\ &= \text{span} \{V_d H_d^{-m}(\beta \mathbf{e}_1), V_d H_d^{-(m-1)}(\beta \mathbf{e}_1), \dots, V_d H_d^{m-1}(\beta \mathbf{e}_1)\} \\ &= V_d \text{span} \{H_d^{-m}(\beta \mathbf{e}_1), H_d^{-(m-1)}(\beta \mathbf{e}_1), \dots, H_d^{m-1}(\beta \mathbf{e}_1)\} \\ &= V_d \mathbf{EK}_m(H_d, (\beta \mathbf{e}_1)). \quad \square \end{aligned}$$

### 3.2. Proposal for an efficient algorithm

From Theorem 3.1 in the previous subsection, the orthonormal basis  $\mathcal{V}_m$  of the extended Krylov subspace can be constructed by multiplying the two orthonormal bases — i.e.

$$\mathcal{V}_m = V_d \mathcal{V}_{d,m}, \quad (3.2)$$

where  $V_d$  corresponds to the Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$  and  $\mathcal{V}_{d,m}$  to the extended Krylov subspace  $\mathbf{EK}_m(H_d, \beta \mathbf{e}_1)$  with the Hessenberg matrix  $H_d := V_d^H A V_d$  and  $\beta \mathbf{e}_1$ . Based on Eq. (3.2), we now propose the novel algorithm to construct the orthonormal basis.

The  $\mathcal{V}_m$  can be constructed via the Krylov subspace  $\mathbf{K}_d(A, \mathbf{b})$  as in Algorithm 3.1, where the main feature is that the coefficient matrix of the linear systems is not  $A$  but the Hessenberg matrix  $H_d$ . Simoncini's algorithm (Algorithm 2.3) requires that we solve  $m$  linear systems (2.1) with  $A$  for constructing  $\mathcal{V}_m$ , whereas on using Algorithm 3.1 we can construct  $\mathcal{V}_m$  by solving  $m$  linear systems with  $H_d$ . Since  $H_d$  is the upper Hessenberg matrix (or the tridiagonal matrix for a Hermitian  $A$ ), linear systems with  $H_d$  can be solved efficiently by a direct method (such as the Givens-rotation method).

**Algorithm 3.1** A theoretical version of a proposed algorithm**Input:**  $A \in \mathbb{C}^{n \times n}$  and  $\mathbf{b} \in \mathbb{C}^n$ **Output:** The orthonormal basis  $\mathcal{V}_m \in \mathbb{C}^{n \times 2m}$  of  $\mathbf{EK}_m(A, \mathbf{b})$  s.t.  $\mathcal{V}_m^H \mathcal{V}_m = I$ 

- 1: Run the Arnoldi/Lanczos procedure (Algorithm 2.2) for  $\mathbf{K}_d(A, \mathbf{b})$ ,  
and get the orthonormal basis  $V_d$  and  $H_d = V_d^H A V_d$ ,  $\beta = \|\mathbf{b}\|_2$
- 2: Run the Simoncini's algorithm (Algorithm 2.3) for  $\mathbf{EK}_m(H_d, \beta \mathbf{e}_1)$ ,  
and get the orthonormal basis  $\mathcal{V}_{d,m}$
- 3: Compute  $\mathcal{V}_m = V_d \mathcal{V}_{d,m}$

**Algorithm 3.2** A practical version of the proposed algorithm**Input:**  $A \in \mathbb{C}^{n \times n}$  and  $\mathbf{b} \in \mathbb{C}^n$ **Output:** The orthonormal basis  $\mathcal{V}_m \in \mathbb{C}^{n \times 2m}$  of  $\mathbf{EK}_m(A, \mathbf{b})$  s.t.  $\mathcal{V}_m^H \mathcal{V}_m = I$ 

- 1: Run the Arnoldi/Lanczos procedure for  $\mathcal{K}_k(A, \mathbf{b})$  until  $A^{-1} \mathbf{b}$  is converged,  
and get the orthonormal basis  $V_k$  and  $H_k = V_k^H A V_k$ ,  $\beta = \|\mathbf{b}\|_2$
- 2: Run Simoncini's algorithm (Algorithm 2.3) for  $\mathbf{EK}_m(H_k, \beta \mathbf{e}_1)$ ,  
and get the orthonormal basis  $\mathcal{V}_{k,m}$
- 3: Compute  $\mathcal{V}_m = V_k \mathcal{V}_{k,m}$ , if required

Unfortunately, it is difficult to know the grade  $d$  and to compute the orthonormal basis  $V_d$  for general matrices  $A$ , so we propose a more practical version (Algorithm 3.2). Thus instead of Eq. (3.2), Algorithm 3.2 is based on the approximation

$$\mathcal{V}_m \approx V_k \mathcal{V}_{k,m}, \quad (3.3)$$

where  $V_k$  is the matrix with its columns the orthonormal basis of  $\mathbf{K}_k(A, \mathbf{b})$ , and  $\mathcal{V}_{k,m}$  is a matrix with its columns the orthonormal basis of  $\mathbf{EK}_m(H_k, \beta \mathbf{e}_1)$  and  $H_k := V_k^H A V_k$ .

Let us consider the validity of the approximation (3.3) for Algorithm 3.2. From the definition of the extended Krylov subspace and Theorem 3.1, the orthonormal basis  $\mathcal{V}_m$  can be written

$$\mathcal{V}_m = F(A, \mathbf{b}) := [f_1(A)\mathbf{b}, f_2(A)\mathbf{b}, \dots, f_{2m}(A)\mathbf{b}], \quad (3.4)$$

using functions of a matrix times a vector. Here, it is known that functions of a matrix times a vector  $f(A)\mathbf{b}$  can be approximated in the Krylov subspace  $\mathbf{K}_k(A, \mathbf{b})$ :

$$f(A)\mathbf{b} \approx V_k f(H_k) V_k^H \mathbf{b} = V_k f(H_k) (\beta \mathbf{e}_1) \quad (3.5)$$

— cf. Ref [7] and references therein. From Eqs. (3.4) and (3.5), a matrix  $\mathcal{V}_m$  can therefore be well approximated in the Krylov subspace  $\mathbf{K}_k(A, \mathbf{b})$  as follows:

$$\begin{aligned} \mathcal{V}_m &\approx [V_k f_1(H_k) \beta \mathbf{e}_1, V_k f_2(H_k) \beta \mathbf{e}_1, \dots, V_k f_{2m}(H_k) \beta \mathbf{e}_1] \\ &= V_k F(H_k, (\beta \mathbf{e}_1)) \\ &= V_k \mathcal{V}_{k,m}, \end{aligned}$$

Table 1: The main computational costs and storage requirements of Simoncini's algorithm and the proposed algorithm for a non-Hermitian  $A$ .

(a) explicit computation of $\mathcal{V}_m$				
	linear systems		misc.	main storage
	for $A$ [times]	for $H_k$ [#flops]	[#flops]	
Algorithm 2.3	$m$	0	$8nm^2$	$2nm$
Algorithm 3.2	1	$\frac{1}{2}k^2m$	$8km^2 + 4nkm$	$2nm + kn + 2km$

(b) approximation of a vector on $\mathbf{EK}_m(A, \mathbf{b})$ : $\mathbf{x}_m = \mathcal{V}_m \mathbf{y}_m$				
	linear systems		misc.	main storage
	for $A$ [times]	for $H_k$ [#flops]	[#flops]	
Algorithm 2.3	$m$	0	$8nm^2 + 4nm$	$2nm$
Algorithm 3.2	1	$\frac{1}{2}k^2m$	$8km^2 + 2nk + 4km$	$kn + 2km$

supporting the validity of (3.3) in terms of approximating functions of a matrix times a vector in the Krylov subspace.

Various stopping criteria for the Arnoldi/Lanczos procedure could be used in Step 1 of Algorithm 3.2. For the computations described in this article, the accuracy of approximation for the linear system  $AW_0(:, 2) = \mathbf{b}$  was adopted, reflecting the fact that Simoncini's algorithm approximately solves the linear system for a large and sparse  $A$ . It is notable that the iterative solver for  $AW_0(:, 2) = \mathbf{b}$  is limited to the FOM method [11] or the GMRES method [14] without preconditioners (or the D-Lanczos method [12, Algorithm 6.17] for a Hermitian  $A$ ), given the necessity for explicit computation of  $H_k$ .

From a practical standpoint, we do not always require that  $\mathcal{V}_m$  be computed explicitly in Step 3 of the algorithm. Thus when used to approximate a vector on  $\mathbf{EK}_m(A, \mathbf{b})$  such as  $\mathbf{x}_m = \mathcal{V}_m \mathbf{y}_m$ , we can compute  $\mathbf{x}_m$  as

$$\mathbf{x}_m = \mathcal{V}_m \mathbf{y}_m = V_k(\mathcal{V}_{k,m} \mathbf{y}_m), \quad (3.6)$$

without explicit computation of  $\mathcal{V}_m$ . Avoiding explicit computation of  $\mathcal{V}_m$  often leads to a significant reduction in computational cost for solving several matrix computations.

Table 1 shows the main computational costs and storage requirements without the coefficient matrix  $A$  of Simoncini's algorithm and the proposed algorithm. The figures in Table 1(a) are for the explicit computation of  $\mathcal{V}_m$ , and the figures in Table 1(b) are for the approximation of a vector without computation of  $\mathbf{y}_m$ , where  $\mathcal{V}_m$  is not explicitly computed — cf. Eq. (3.6). The storage requirements of the proposed algorithm can be reduced by a recalculation of  $V_k$  in Step 3, especially for a Hermitian  $A$ , in return for increasing the number of linear systems for  $A$  from 1 to 2.

## 4. Numerical Experiments and Results

The relative performance of the proposed algorithm (Algorithm 3.2) and Simoncini's algorithm (Algorithm 2.3) was compared as follows. In Section 4.1, the accuracy of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  constructed by both algorithms is considered, for symmetric and nonsymmetric model problems. In Section 4.2, we apply both algorithms to matrices obtained from the University of Florida Sparse Matrix Collection [1], and then compare their computation times required for constructing  $\mathcal{V}_m$ . In Section 4.3, we evaluate the efficiency of both algorithms to functions of a matrix times a vector, in the case of symmetric matrices.

All numerical experiments were carried out in double precision arithmetic on OS: CentOS (64bit), CPU: Intel Xeon X5550 2.67GHz (single core), Memory: 48GB, Compiler: GNU Fortran ver. 4.1.2, Compile option: -O3.

### 4.1. Numerical experiment I

The two model problems for constructing  $\mathcal{V}_m$  are as follows. The first problem is for a diagonal matrix

$$\begin{aligned} \mathcal{V}_m &\leftarrow \mathbf{EK}_m(A, \mathbf{b}), \quad \mathcal{V}_m^H \mathcal{V}_m = I, \\ A &= \text{diag}(0.01, 0.02, \dots, 1.00), \quad \mathbf{b} = [1, 1, \dots, 1]^T; \end{aligned} \quad (4.1)$$

and the second is for an upper bidiagonal matrix with 0.1 superdiagonal entries

$$\begin{aligned} \mathcal{V}_m &\leftarrow \mathbf{EK}_m(A, \mathbf{b}), \quad \mathcal{V}_m^H \mathcal{V}_m = I, \\ A &= \text{bidiag}(\mathbf{d}, 0.1), \quad \mathbf{d} = [0.02^2, 0.04^2, \dots, 1.00^2], \quad \mathbf{b} = [1, 1, \dots, 1]^T. \end{aligned} \quad (4.2)$$

The proposed algorithm (Algorithm 3.2) and Simoncini's algorithm (Algorithm 2.3), which is based on the Krylov subspace method, were applied to both of these two problems. In particular, the accuracy of the constructed subspaces for the principle angles  $\cos(\theta_i)$ ,  $i = 1, 2, \dots, 2m$  was compared with the subspace constructed by Simoncini's algorithm based on a direct method.

Let  $\mathbf{P}$  and  $\mathbf{Q}$  be subspaces in  $\mathbb{C}^n$  of dimension  $s$ , and let  $P, Q \in \mathbb{C}^{n \times s}$  be column orthogonal matrices with columns the orthonormal bases of  $\mathbf{P}$  and  $\mathbf{Q}$ , respectively. The principle (or canonical) angles  $\cos(\theta_i)$  between the subspaces  $\mathbf{P}$  and  $\mathbf{Q}$  are thus

$$\cos(\theta_i) = \sigma_i(P^H Q), \quad i = 1, 2, \dots, s,$$

where  $\sigma_i(P^H Q)$  are the singular values of  $P^H Q$ . It is notable that  $0 \leq \cos(\theta_i) \leq 1$ , and in the case of the principle angles that  $\cos(\theta_i) = 1$  for all  $i$ . The constructed subspaces are equivalent, but in the case of  $\cos(\theta_i) = 0$  for all  $i$  the subspaces are orthogonal — e.g. see Ref. [3].

For the iterative solver of Step 1 in the proposed algorithm, the D-Lanczos method was used for the model problem (4.1) with a symmetric  $A$ , and the FOM method for the

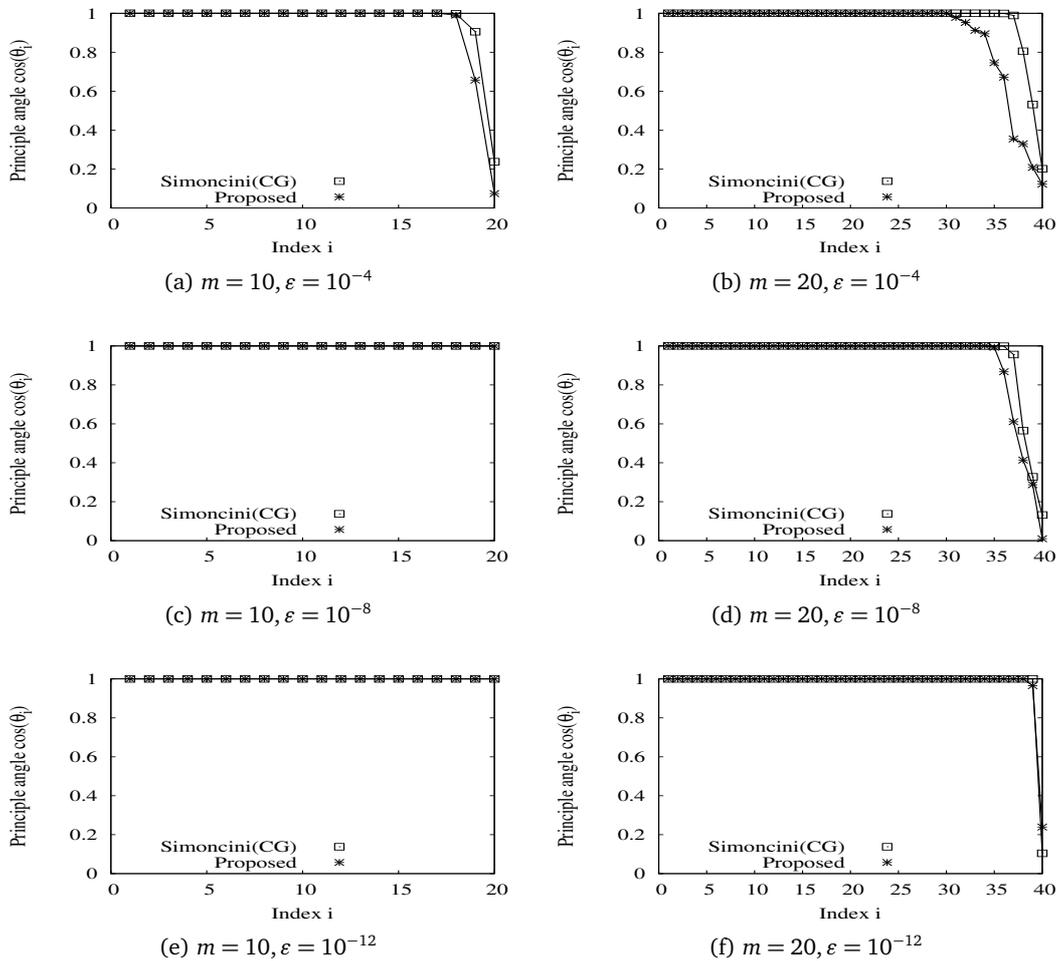


Figure 1: Graphs of the principle angles  $\cos(\theta_i), i = 1, 2, \dots, 2m$  for the model problem (4.1).

model problem (4.2) with a nonsymmetric  $A$ . The CG method [4] and the FOM method were used to solve each linear system in Simoncini’s algorithm for the model problems (4.1) and (4.2), respectively. With  $\mathbf{EK}_m(A, b)$  when  $m = 10, 20$ , for the CG method, the D-Lanczos method and the FOM method the stopping criteria were set at a relative residual 2-norm less than  $\epsilon$ , where  $\epsilon = 10^{-4}, 10^{-8}, 10^{-12}$ . The initial approximate solution of all of the linear systems was set to be the zero vector  $[0, 0, \dots, 0]^T$ .

Numerical results are shown for the model problem (4.1) in Fig. 1, and for the model problem (4.2) in Fig. 2. For both problems, the principle angles corresponding to both algorithms are  $\cos(\theta_i) \approx 1$  for  $m = 10, \epsilon = 10^{-12}$  — cf. Fig. 1 (e) and 2 (e). Thus with  $\epsilon = 10^{-12}$  each algorithm can accurately construct the extended Krylov subspace  $\mathbf{EK}_m(A, b)$  when  $m = 10$ . From Fig. 1 and 2, it is evident that some part of the principle angles

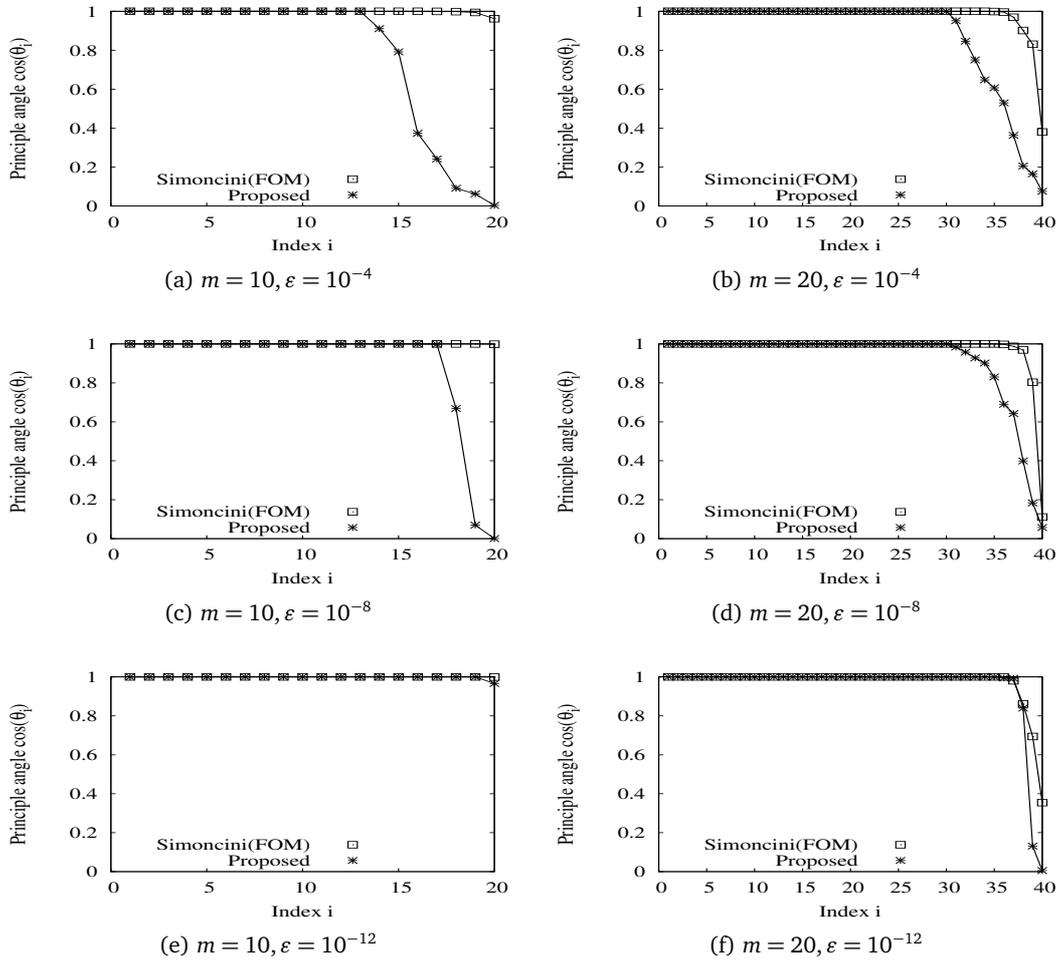


Figure 2: Graphs of the principle angles  $\cos(\theta_i), i = 1, 2, \dots, 2m$  for the model problem (4.2).

becomes  $\cos(\theta_i) \not\approx 1$  as the stopping criterion increases. Deteriorating accuracy appears more prominently when  $m = 20$ .

The proposed algorithm was then compared with Simoncini’s algorithm using the CG method, where Simoncini’s algorithm showed higher accuracy for  $\epsilon = 10^{-4}$  and  $10^{-8}$ . On the other hand, for  $\epsilon = 10^{-12}$  both algorithms constructed the extended Krylov subspace with approximately the same accuracy.

### 4.2. Numerical experiment II

In this subsection, we apply Simoncini’s algorithm and the proposed algorithm to matrices obtained from the University of Florida Sparse Matrix Collection, in constructing

Table 2: Characteristics of the test matrices for Simoncini’s algorithm and the proposed algorithm.

(Type) Matrix name	$n$	$Nnz$	$Ave.Nnz$	Application area
(S) BCSSTK27	1224	56126	45.85	Structural problem
(S) CRYSTM01	4875	105339	21.61	Materials problem
(S) KUU	7102	340200	47.90	Structural problem
(S) FV1	9604	85264	8.88	2D/3D problem
(N) ADD20	2395	17319	7.23	Circuit simulation
(N) POISSON3DA	13514	352762	26.10	Computational fluid dynamics
(N) SHERMAN4	1104	3786	3.43	Computational fluid dynamics
(N) RAEFSKY1	3242	294276	90.77	Computational fluid dynamics

orthonormal bases for the extended Krylov subspaces. Their computation times are then compared.

The characteristics of the test matrices are shown in Table 2. Here (S) and (N) respectively denote real symmetric and real nonsymmetric matrices, and  $n$ ,  $Nnz$  and  $Ave.Nnz$  respectively denote the dimension, the number of nonzero elements and the average nonzero elements per row or column.

For the iterative solver in Step 1 of the proposed algorithm, we used the D-Lanczos method for the symmetric matrices, and the FOM method for the nonsymmetric matrices. The ICCG method [10] (the CG method with the IC(0) preconditioner) and the BiCGSTAB method [18] with the ILU(0) preconditioner [10] were used to solve the linear systems in Simoncini’s algorithm for symmetric and nonsymmetric matrices, respectively. For the proposed algorithm, we explicitly constructed  $\mathcal{V}_m$  and recalculated  $V_k$  in Step 3, and set  $m$  of  $\mathbf{EK}_m(A, \mathbf{b})$  as  $m = 1, 2, \dots, 50$  and  $\mathbf{b} = [1, 1, \dots, 1]^T$ . The stopping criterion of the linear solvers was set at the relative residual 2-norm less than  $10^{-10}$ . The initial approximate solution of all of the linear systems was set to be the zero vector  $[0, 0, \dots, 0]^T$ .

Fig. 3 shows the computation times for  $m = 1, 2, \dots, 50$ . When based on the direct method, Simoncini’s algorithm first decomposes the coefficient matrix  $A$  and so requires a long computation time even for a small  $m$ , but especially for large problems — cf. KUU, FV1 and POISSON3DA. Conversely, it shows a relatively small increase of computation time for different  $m$ . However, when Simoncini’s algorithm is based on the iterative method,  $\mathcal{V}_m$  can be constructed within a comparatively shorter computation time for a small  $m$ . — although the computation time increases directly with  $m$ , because the most time-consuming aspect of Simoncini’s algorithm involves solving  $m$  linear systems (2.1), as seen in Table 1.

The proposed algorithm needs to solve the linear system for  $A$  in Step 1 without preconditioners, because it must directly compute  $H_k$ . In the case of a small  $m$ , it therefore has a longer computation time than Simoncini’s algorithm using the preconditioned Krylov subspace method. However, in contrast to Simoncini’s algorithm, the main computational costs of the proposed algorithm arise from solving 1 or 2 linear systems for  $A$  independent of  $m$ , so there a very small increase of computation time with  $m$ . Consequently, in the case

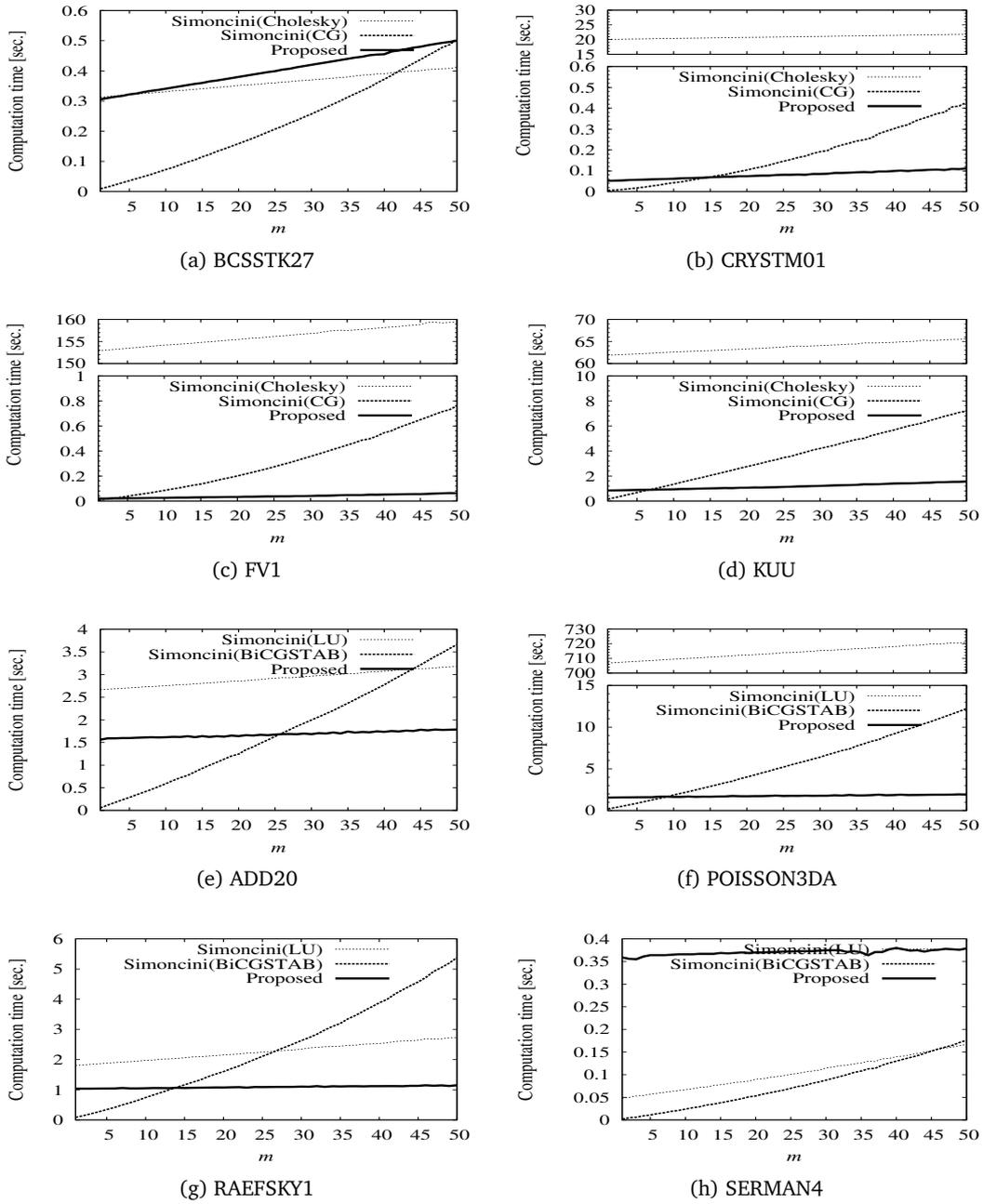


Figure 3: Computation time for  $m = 1, 2, \dots, 50$ .

of a large  $m$ , the proposed algorithm can construct  $\mathcal{V}_m$  with minimal computation time for many matrices.

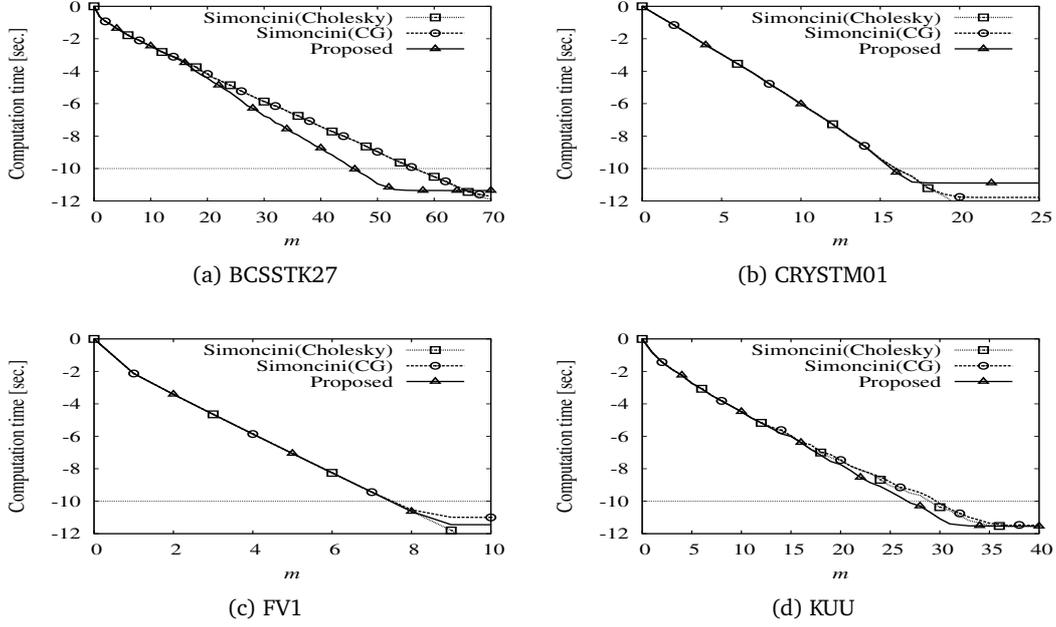


Figure 4: The relative error vs.  $m$  for solving the function of a matrix times a vector.

### 4.3. Numerical experiment III

In this subsection, the function of a matrix times a vector

$$\begin{aligned} \mathbf{x} &= f(A)\mathbf{b}, \quad A \in \mathbb{R}^{n \times n}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^n, \\ f(\lambda) &= 1/\sqrt{\lambda}, \quad \mathbf{b} = [1, 1, \dots, 1]^T, \end{aligned} \quad (4.3)$$

is computed by projection onto the extended Krylov subspace — thus

$$\mathbf{x} \approx \mathbf{x}_m = \mathcal{V}_m f(\mathcal{H}_m) \boldsymbol{\beta} e_1,$$

where  $\mathcal{H}_m := \mathcal{V}_m^T A \mathcal{V}_m$  can be efficiently computed using Simoncini's algorithm as in the Arnoldi procedure [15] and  $\boldsymbol{\beta} = \|\mathbf{b}\|_2$ . The computation efficiency of the proposed algorithm was compared with Simoncini's algorithm for computing (4.3). For the test matrices, we used the real symmetric matrices shown in Table 2 — i.e. BCSSTK27, CRYSTM01, FV1 and KUU.

For the iterative solver in Step 1 of the proposed algorithm, we used the D-Lanczos method. We also used the ICCG method to solve each of the linear systems of Simoncini's algorithm. For the proposed algorithm, we computed  $\mathbf{x}_m$  by Eq. (3.6) instead of explicitly constructing  $\mathcal{V}_m$ .  $V_k$  was recalculated in Step 3.

The stopping criterion of the linear solvers was set at a relative residual 2-norm less than  $10^{-10}$ . The initial approximate solution of all of the linear systems was again set to be the zero vector  $[0, 0, \dots, 0]^T$ .

Table 3: Convergence results of Simoncini's algorithm and the proposed algorithm.

Matrix name	Simoncini(Ch)		Simoncini(CG)		Proposed	
	$m$	time [sec.]	$m$	time [sec.]	$m$	time [sec.]
BCSSTK27	57	$4.33 \times 10^{-1}$	57	$6.25 \times 10^{-1}$	46	$3.14 \times 10^{-1}$
CRYSTM01	16	$2.06 \times 10^1$	16	$8.23 \times 10^{-2}$	16	$5.30 \times 10^{-2}$
FV1	8	$1.54 \times 10^2$	8	$7.50 \times 10^{-2}$	8	$1.90 \times 10^{-2}$
KUU	29	$6.42 \times 10^1$	30	$4.30 \times 10^0$	27	$8.35 \times 10^{-1}$

Fig. 4 shows the relative error  $\|\mathbf{x}^* - \mathbf{x}_m\|_2 / \|\mathbf{x}^*\|_2$  for each  $m$ , where  $\mathbf{x}^*$  is the exact solution of function (4.3) as computed from the eigenvalue decomposition of  $A$ . We also show the computation time and  $m$  satisfying  $\|\mathbf{x}^* - \mathbf{x}_m\|_2 / \|\mathbf{x}^*\|_2 \leq 10^{-10}$  in Table 3.

From Fig. 4, we see that the proposed algorithm shows approximately the same convergence behaviour as does the Simoncini's algorithm despite less accuracy, as noted in Section 4.1. For BCSSTK27 in particular, the proposed algorithm shows better convergence behaviour. As shown in Table 3, given approximately the same convergence behaviour and the smaller computational costs for constructing the basis, the proposed algorithm computes function (4.3) with a much shorter computation time than Simoncini's algorithm.

## 5. Conclusions

The characteristics of the linear systems arising from the construction of a basis for the extended Krylov subspace is analysed in this article, and an efficient algorithm for constructing an orthonormal basis of the extended Krylov subspace proposed.

From a series of numerical experiments, we conclude that the proposed algorithm can construct the orthonormal basis  $\mathcal{V}_m$  of the extended Krylov subspace  $\mathbf{EK}_m(A, \mathbf{b})$  within a shorter and less  $m$ -dependent computation time than Simoncini's algorithm. In addition, while our numerical results indicate that the accuracy of our proposed algorithm is somewhat reduced in terms of principle angles, this does not greatly influence the efficiency in solving a function of a matrix times a vector.

In future work, the relationship between the accuracy of the proposed algorithm and Simoncini's algorithm using iterative methods may be investigated, and the efficiency in solving a wide variety of large matrix computations explored.

## Acknowledgments

This work was supported in part by MEXT SPIRE, JICFuS and KEKENHI (Grant No. 25870099).

## References

- [1] T.A. Davis, *The University of Florida Sparse Matrix Collection*, <http://www.cise.ufl.edu/research/sparse/matrices/>.
- [2] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl. **19**, 755-771 (1998).
- [3] G.H. Golub and C.F. van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore (1997).
- [4] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nati. Bur. Stat. **49**, 409-436 (1952).
- [5] M. Heyouni, *Extended Arnoldi methods for large Sylvester matrix equations*, Technical report, L.M.P.A. (2008).
- [6] M. Heyouni and K. Jbilou, *An extended block method for large-scale continuous-time algebraic Riccati equations*. Technical report, L.M.P.A., Universit  du Littoral (2007).
- [7] N.J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia (2008).
- [8] L. Knizhnerman and V. Simoncini, *A new investigation of the extended Krylov subspace method for matrix function evaluations*, Numer. Linear Alg. Appl. **17**, 615-638 (2010).
- [9] L. Knizhnerman and V. Simoncini, *Convergence analysis of the extended Krylov subspace method for the Lyapunov equation*, Numer. Math. **118**, 567-586 (2011).
- [10] J.A. Meijerink and A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comput. **31**, 148-162 (1977).
- [11] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comput. **37**, 105-126 (1981).
- [12] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia (2003).
- [13] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, 2nd edition, SIAM, Philadelphia (2011).
- [14] Y. Saad and M.H. Schultz, *GMRES: A generalised minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Stat. Comput. **7**, 856-869 (1986).
- [15] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput. **29**, 1268-1288 (2007).
- [16] V. Simoncini, *Extended Krylov subspace for parameter dependent systems*, Appl. Numer. Math. **60**, 550-560 (2010).
- [17] M. Sugihara and K. Murota, *Theoretical Numerical Linear Algebra*, Iwanami, Tokyo (2009), in Japanese.
- [18] 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**, 631-644 (1992).