

# Krylov Subspace Methods for Tensor Computations

Berkant Savas and Lars Eldén

20 februari 2009

LITH-MAT-R-2009-02-SE



**Linköpings universitet**

Matematiska Institutionen  
Linköpings Universitet  
581 83 Linköping

Department of Mathematics  
Linköpings University  
581 83 Linköping

# Krylov Subspace Methods for Tensor Computations.

Berkant Savas and Lars Eldén  
Department of Mathematics  
Linköping University

February 20, 2009

## Abstract

A couple of generalizations of matrix Krylov subspace methods to tensors are presented. It is shown that a particular variant can be interpreted as a Krylov factorization of the tensor. A generalization to tensors of the Krylov-Schur method for computing matrix eigenvalues is proposed. The methods are intended for the computation of low-rank approximations of large and sparse tensors. A few numerical experiments are reported.

## 1 Introduction

Large-scale problems in engineering and science often require solution of sparse linear algebra problems, such as systems of equations, and eigenvalue problems. Since the 1950's Krylov subspace methods have become the main class of algorithms for solving iteratively large and sparse matrix problems. Given a square matrix  $A \in \mathbb{R}^{m \times m}$  and starting vector  $u_1 \in \mathbb{R}^n$  the Krylov subspace based on  $A$  and  $u_1$  is given by

$$\mathcal{K}_k(A, u_1) = \text{span}\{u_1, Au_1, A^2u_1, \dots, A^{k-1}u_1\}. \quad (1)$$

In floating point arithmetic the vectors in (1) are useless unless they are orthonormalized. Applying Gram-Schmidt orthogonalization one obtains the Arnoldi process, which, in exact arithmetic, generates an orthonormal basis for the Krylov subspace  $\mathcal{K}_k(A, u_1)$ . In addition, the Arnoldi process generates the factorization

$$AU_k = U_{k+1}\hat{H}_k, \quad (2)$$

where  $U_k = (u_1 \dots u_k)$ ,  $U_{k+1} = (U_k \ u_{k+1})$  with orthonormal columns, and  $\hat{H}_k$  is an Hessenberg matrix with the orthonormalization coefficients. Based on the factorization (2) one can compute an approximation of the solution

of a linear system or an eigenvalue problem by projecting onto the space spanned by the columns of  $U_k$ , where  $k$  is much smaller than the dimension of  $A$ ; on that subspace the operator  $A$  is represented by the small matrix  $\widehat{H}_k$ . This approach is particularly useful for large, and sparse problems, since it uses the matrix  $A$  in matrix-vector multiplications only.

Projection to a low-dimensional subspace is a common technique in many areas of information science. Often in such applications large and sparse tensors occur, see e.g. the series of papers by Bader, Kolda, et al. [1, 4, 5, 8, 13, 14, 15]. The following question arises naturally:

*Can Krylov methods be generalized to tensors, to be used for projection to low-dimensional subspaces?*

We answer this question in the affirmative, and describe several alternative ways one can implement Krylov subspace methods for tensors. Our method is inspired by Golub-Kahan bidiagonalization [10], and the Arnoldi method, see e.g. [19, p. 303]. In the bidiagonalization procedure two sequences of orthogonal vectors are generated; for a tensor of order three, our procedure generates three sequences of orthogonal vectors. Unlike the matrix case, it turns out to be necessary to perform Arnoldi style orthogonalization of the generated vectors explicitly. For matrices, once an initial vector has been selected, the whole sequence is determined uniquely. For tensors, there are many ways in which the vectors can be generated, and we describe a few alternatives.

Restarted Arnoldi methods [16, 18] and a variant called the Krylov-Schur method [19, 20] are used for computing eigenvalues of large sparse matrices. We sketch a generalization of the Krylov-Schur procedure to tensors.

The aim of this paper is not to give a comprehensive theory for tensor Krylov methods, but rather to present some concepts that we believe are new and potentially important, and to perform a preliminary exploration of those concepts.

The paper is organized as follows. The necessary tensor concepts are introduced in Section 2. The Arnoldi and Golub-Kahan procedures are sketched in Section 3. In Section 4 we describe three variants of Krylov methods for tensors, and in Section 5 we propose a generalization of the Krylov-Schur method. Numerical examples are given in Section 6.

## 2 Tensor Concepts

### 2.1 Notation and preliminaries.

Tensors will be denoted by calligraphic letters, e.g.  $\mathcal{A}, \mathcal{B}$ , matrices by capital roman letters and vectors by small roman letters. In order not to burden the presentation with too much detail, we sometimes will not explicitly mention the dimensions of matrices and tensors, and assume that they are such that

the operations are well-defined. The whole presentation will be in terms of tensors of order three (3-tensors). The generalization to order- $N$  tensors is obvious.

Let  $\mathcal{A}$  denote a tensor in  $\mathbb{R}^{J \times K \times L}$ . We will use the term tensor in a restricted sense, i.e. as a 3-dimensional array of real numbers, equipped with some algebraic structures to be defined. The different “dimensions” of the tensor are referred to as *modes*. We will use both standard subscripts and “MATLAB-like” notation: a particular tensor element will be denoted in two equivalent ways:

$$\mathcal{A}(i, j, k) = a_{ijk}.$$

We will refer to subtensors in the following way. A subtensor obtained by fixing one of the indices is called a *slice*, e.g.,

$$\mathcal{A}(i, :, :).$$

Such a slice is usually considered as a 3-order tensor.

A *fibre* is a subtensor, where all indices but one are fixed,

$$\mathcal{A}(i, :, k).$$

It is customary in numerical linear algebra to write out column vectors with the elements on top of each other, and row vectors with the elements after each other on a line. This becomes inconvenient when we are dealing with more than two modes. Therefore we will allow ourselves to write all vectors organized vertically. It will be clear from the context which mode the vectors belong to.

## 2.2 Tensor-Matrix Multiplication

We define mode- $p$  *multiplication of a tensor by a matrix* as follows. For concreteness we first let  $p = 1$ . The mode-1 product of a tensor  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  by a matrix  $W \in \mathbb{R}^{M \times J}$  is defined

$$\mathbb{R}^{M \times K \times L} \ni \mathcal{B} = (W)_1 \cdot \mathcal{A}, \quad b_{mkl} = \sum_{j=1}^J w_{mj} a_{jkl}. \quad (3)$$

This means that all column vectors (mode-1 fibres) in the 3-tensor are multiplied by the matrix  $W$ . Similarly, mode-2 multiplication by a matrix  $X$  means that all row vectors (mode-2 fibres) are multiplied by the matrix  $X$ . Mode-3 multiplication is analogous.

In the case when tensor-matrix multiplication is performed in all modes in the same formula, we omit the subscripts and write

$$(X, Y, Z) \cdot \mathcal{A}, \quad (4)$$

where the mode of each multiplication is understood from the order in which the matrices are given.

The notation (4) was suggested by Lim [7]. An alternative notation was earlier given in [6]. Our  $(W)_p \cdot \mathcal{A}$  is the same as  $\mathcal{A} \times_p W$  in that system.

It is convenient to introduce a separate notation for multiplication by a transposed matrix  $V \in \mathbb{R}^{J \times M}$ :

$$\mathbb{R}^{M \times K \times L} \ni \mathcal{C} = \left(V^T\right)_1 \cdot \mathcal{A} = \mathcal{A} \cdot (V)_1, \quad c_{mkl} = \sum_{j=1}^J a_{jkl} v_{jm}. \quad (5)$$

Let  $u \in \mathbb{R}^J$  be a column vector and  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  a tensor. Then

$$\mathcal{B} \ni \mathbb{R}^{1 \times K \times L} := \left(u^T\right)_1 \cdot \mathcal{A} = \mathcal{A} \cdot (u)_1 \equiv B \in \mathbb{R}^{K \times L}. \quad (6)$$

Thus we identify a tensor with a singleton dimension with a matrix. Similarly, with  $u \in \mathbb{R}^J$  and  $w \in \mathbb{R}^L$ , we will identify

$$\mathcal{U} \ni \mathbb{R}^{1 \times K \times 1} := \mathcal{A} \cdot (u, w)_{1,3} \equiv v \in \mathbb{R}^K, \quad (7)$$

i.e., a tensor of order three with two singleton dimensions is identified with a vector, here in mode 2.

### 2.3 Inner Product, Norm, and Contractions

Given two tensors  $\mathcal{A}$  and  $\mathcal{B}$  of the same dimensions, we define the *inner product*,

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{j,k,l} a_{jkl} b_{jkl}. \quad (8)$$

The corresponding *tensor norm* is

$$\|\mathcal{A}\| = \langle \mathcal{A}, \mathcal{A} \rangle^{1/2}. \quad (9)$$

This *Frobenius norm* will be used throughout the paper. As in the matrix case, the norm is invariant under orthogonal transformations, i.e.

$$\|\mathcal{A}\| = \|(U, V, W) \cdot \mathcal{A}\| = \|\mathcal{A} \cdot (U, V, W)\|,$$

for orthogonal matrices  $U$ ,  $V$ , and  $W$ . This follows immediately from the fact that mode- $p$  multiplication by an orthogonal matrix does not change the Euclidean length of the mode- $p$  fibres.

The following lemma will be needed.

**Lemma 2.1.** *Let  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  be given along with three matrices with orthonormal columns,  $U \in \mathbb{R}^{J \times j}$ ,  $V \in \mathbb{R}^{K \times k}$ , and  $W \in \mathbb{R}^{L \times l}$ , where  $j \leq J$ ,  $k \leq K$ , and  $l \leq L$ . Then the least squares problem*

$$\min_{\mathcal{S}} \|\mathcal{A} - (U, V, W) \cdot \mathcal{S}\|$$

has the unique solution

$$\mathcal{S} = \left( U^\top, V^\top, W^\top \right) \cdot \mathcal{A} = \mathcal{A} \cdot (U, V, W).$$

*Proof.* The proof is a straightforward generalization of the corresponding proof for matrices. Enlarge each of the matrices so that it becomes square and orthogonal, i.e., put

$$\widehat{U} = (U U_\perp), \quad \widehat{V} = (V V_\perp), \quad \widehat{W} = (W W_\perp).$$

Then, using the invariance of the norm under orthogonal transformations, we get

$$\begin{aligned} \|\mathcal{A} - (U, V, W) \cdot \mathcal{S}\|^2 &= \left\| \left( \widehat{U}^\top, \widehat{V}^\top, \widehat{W}^\top \right) \cdot [\mathcal{A} - (U, V, W) \cdot \mathcal{S}] \right\|^2 \\ &= \left\| \left( U^\top, V^\top, W^\top \right) \cdot \mathcal{A} - \mathcal{S} \right\|^2 + C^2, \end{aligned}$$

where  $C^2$  consists of terms not depending on  $\mathcal{S}$ . □

The inner product (8) can be considered as a special case of the *contracted product of two tensors*, cf. [12, Chapter 2], which is a tensor (outer) product followed by a contraction along specified modes. Thus, if  $\mathcal{A}$  and  $\mathcal{B}$  are 3-tensors, we define, using essentially the notation of [2],

$$\begin{aligned} \mathcal{C} &= \langle \mathcal{A}, \mathcal{B} \rangle_1, & c_{jklm} &= \sum_{\lambda} a_{\lambda jk} b_{\lambda lm}, & \text{(4-tensor),} \\ \mathcal{D} &= \langle \mathcal{A}, \mathcal{B} \rangle_{1:2}, & d_{jk} &= \sum_{\lambda, \mu} a_{\lambda \mu j} b_{\lambda \mu k}, & \text{(2-tensor),} \\ e &= \langle \mathcal{A}, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \rangle_{1:3}, & e &= \sum_{\lambda, \mu, \nu} a_{\lambda \mu \nu} b_{\lambda \mu \nu}, & \text{(scalar).} \end{aligned}$$

It is required that contracted dimensions are equal in the two tensors. We will refer to the first two as *partial contractions*.

Observe that we let the ordering of the modes in contracted tensor products be implicitly given in the summation. Thus with  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  and  $\mathcal{B} \in \mathbb{R}^{J \times M \times N}$ , we have

$$\mathcal{C} = \langle \mathcal{A}, \mathcal{B} \rangle_1 \in \mathbb{R}^{K \times L \times M \times N}.$$

In general, the modes of the product are those of the non-contracted modes of the first argument, followed by those of the non-contracted modes of the second argument, in their respective orders.

We will also use negative subscripts when the contraction is made in all but a few modes. We will write

$$\langle \mathcal{A}, \mathcal{B} \rangle_{2:3} \equiv \langle \mathcal{A}, \mathcal{B} \rangle_{-1}, \quad \langle \mathcal{A}, \mathcal{B} \rangle_2 \equiv \langle \mathcal{A}, \mathcal{B} \rangle_{-(1,3)}.$$

### 3 Two Krylov Methods for Matrices

In this section we will describe briefly the two matrix Krylov methods that are the starting point of our generalization to tensor Krylov methods.

#### 3.1 The Arnoldi Procedure

The Arnoldi procedure is used to compute a low-rank approximation/factorization (2) of a square, in general nonsymmetric matrix  $A$ . It requires a starting vector  $u_1$ , and in each step the new vector is orthogonalized against all previous vectors using the Gram-Schmidt process. We present the Arnoldi procedure in the style of [19, p. 303].

---

##### Arnoldi Procedure

---

```

for  $j = 1, 2, \dots, k$  do
   $h_j = U_j^\top A u_j$ 
   $h_{j+1,j} u_{j+1} = A u_j - U_j h_j$ 
   $\widehat{H}_j = \begin{pmatrix} \widehat{H}_{j-1} & h_j \\ 0 & h_{j+1,j} \end{pmatrix}$ 
end for

```

---

The constant  $h_{j+1,j}$  is used to normalize the new vector to length 1. Note that the matrix  $\widehat{H}_k$  in the factorization (2) is obtained by collecting the orthonormalization coefficients  $h_j$  and  $h_{j+1,j}$  in an upper Hessenberg matrix.

#### 3.2 Golub-Kahan Bidiagonalization

Let  $A \in \mathbb{R}^{m \times n}$  be a matrix, and let  $\beta_1 u_1, v_0 = 0$ , where  $\|u_1\| = 1$ , be starting vectors. The Golub-Kahan bidiagonalization procedure [10] is defined by the following recursion.

---

##### Golub-Kahan bidiagonalization

---

```

for  $j = 1, 2, \dots, k$  do
  1  $\alpha_j v_j = A^\top u_j - \beta_j v_{j-1}$ 
  2  $\beta_{j+1} u_{j+1} = A v_j - \alpha_j u_j$ 
end for

```

---

The scalars  $\alpha_j, \beta_j$  are chosen to normalize the generated vectors  $v_j, u_j$ . Forming the matrices  $U_k = (u_1 \cdots u_{k+1}) \in \mathbb{R}^{m \times (k+1)}$  and  $V_k = (v_1 \cdots v_k) \in \mathbb{R}^{n \times k}$ , it is straightforward to show that

$$A V_k = U_{k+1} B_{k+1}, \quad A^\top U_k = V_k \widehat{B}_k, \quad (10)$$

where  $V_k^\top V_k = I$ ,  $U_{k+1}^\top U_{k+1} = I$ , and

$$B_{k+1} = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{B}_k \\ \beta_{k+1} e_k^\top \end{pmatrix} \in \mathbb{R}^{(k+1) \times k}$$

is bidiagonal<sup>1</sup>.

Using tensor notation from Section 2.2, and a special case of the identification (7), we may express the two steps of the recursion as

---

```

for  $j = 1, 2, \dots, k$  do
  1  $\alpha_j v_j = A \cdot (u_j)_1 - \beta_j v_{j-1}$ 
  2  $\beta_{j+1} u_{j+1} = A \cdot (v_j)_2 - \alpha_j u_j$ 
end for

```

---

We observe that the  $u_j$  vectors “live” in the first mode (column space) of  $A$ , and we generate the sequence  $u_2, u_3, \dots$ , by multiplication of the  $v_j$  vectors in the second mode, and vice versa.

## 4 Tensor Krylov Methods

### 4.1 A Minimal Krylov Recursion

Let  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  be a given tensor of order three. It is now straightforward to generalize the Golub-Kahan procedure. The mode-1 sequence of vectors  $(u_j)$  will be generated by multiplication of mode-2 and mode-3 vectors  $(v_j)$  and  $(w_j)$  by the tensor, and similarly for the other sequences. The newly generated vector is immediately orthogonalized against all the previous ones in its mode, using the Gram-Schmidt process.

To start the recursion two vectors are needed, and we assume that  $u_1$  and  $v_1$  are given. In the algorithm description it is understood that  $U_{\nu-1} = (u_1, u_2, \dots, u_{\nu-1})$ , etc. For reasons that will become clear later, we will refer to this recursion as a minimal Krylov recursion.

---

#### Minimal Krylov recursion

---

```

 $\hat{h} w_1 = \mathcal{A} \cdot (u_1, v_1)_{1,2}$ 
for  $\nu = 2, 3, \dots, k$  do
   $h_u = \mathcal{A} \cdot (U_{\nu-1}, v_{\nu-1}, w_{\nu-1})$ 

```

---

<sup>1</sup>Note that the two sequences of vectors become orthogonal automatically; this is related to the fact that the bidiagonalization procedure is equivalent to the Lanczos process applied to the two symmetric matrices  $AA^\top$  and  $A^\top A$ .

$$\begin{aligned}
\hat{h}u_\nu &= \mathcal{A} \cdot (v_{\nu-1}, w_{\nu-1})_{2,3} - U_{\nu-1}h_u \\
h_v &= \mathcal{A} \cdot (u_\nu, V_{\nu-1}, w_{\nu-1}) \\
\hat{h}v_\nu &= \mathcal{A} \cdot (u_\nu, w_{\nu-1})_{1,3} - V_{\nu-1}h_v \\
h_w &= \mathcal{A} \cdot (u_\nu, v_\nu, W_{\nu-1}) \\
\hat{h}w_\nu &= \mathcal{A} \cdot (u_\nu, v_\nu)_{1,2} - W_{\nu-1}h_w \\
\text{end for}
\end{aligned}$$


---

The constants  $\hat{h}$  are used to normalize the generated vectors to length 1. Assuming the process does not break down<sup>2</sup>, i.e. we obtain a new vector  $u_\nu$  which is linear combination of the vectors in  $U_{n\nu-1}$ , it is straightforward to show that the generated vectors are orthogonal: Obviously  $h_u$  is a vector in  $\mathbb{R}^{\nu-1}$ ; we can write

$$\begin{aligned}
\hat{h}U_{\nu-1}^\top u_\nu &= \left(U_{\nu-1}^\top\right)_1 \cdot \left[\mathcal{A} \cdot (v_{\nu-1}, w_{\nu-1})_{2,3}\right] - h_u \\
&= \mathcal{A} \cdot (U_{\nu-1}, v_{\nu-1}, w_{\nu-1}) - h_u = 0.
\end{aligned}$$

To our knowledge there is no simple way of writing this minimal Krylov recursion as a tensor Krylov factorization. However, having generated three matrices  $U_j$ ,  $V_k$ , and  $W_l$ , we can easily compute a low-rank tensor approximation of  $\mathcal{A}$  using Lemma 2.1.

## 4.2 A Maximal Krylov Recursion

Note that when a new  $u_\nu$  is generated in the minimal Krylov procedure, then we use the most recently computed  $v_{\nu-1}$  and  $w_{\nu-1}$ . In fact, we might choose any combination of previously computed  $v_\mu$  and  $w_\lambda$  that have not been used before to generate a  $u$ -vector:

Let  $\mu \leq \nu - 1$  and  $\lambda \leq \nu - 1$ , and consider the computation of a new  $u$ -vector, which we may write

$$\begin{aligned}
h_u &= \mathcal{A} \cdot (U_{\nu-1}, v_\mu, w_\lambda) \\
h_{\nu\mu\lambda}u_\nu &= \mathcal{A} \cdot (v_\mu, w_\lambda)_{2,3} - U_{\nu-1}h_u
\end{aligned}$$


---

Thus if we are prepared to use all previously computed  $v$ - and  $w$ -vectors, then we have a much richer combinatorial structure, which we illustrate in the following diagram. Assume that  $u_1$  and  $v_1$  are given. Then in the first steps of the Krylov procedure the following vectors can be generated by combining previous vectors.

---

<sup>2</sup>This may indeed happen, but can be easily fixed by taking an orthogonal vector to the previously computed vectors.

$$\begin{array}{l|l}
1. \{u_1\} \times \{v_1\} \longrightarrow w_1 & \\
2. \{v_1\} \times \{w_1\} \longrightarrow u_2 & \\
3. \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \times \{w_1\} \longrightarrow \begin{Bmatrix} v_2 \\ v_3 \end{Bmatrix} & \\
4. \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \times \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \longrightarrow \begin{Bmatrix} (w_1) \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{Bmatrix} & \\
5. \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \times \begin{Bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{Bmatrix} \longrightarrow \begin{Bmatrix} (u_2) \\ u_3 \\ \vdots \\ u_{19} \end{Bmatrix} & \\
6. \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{19} \end{Bmatrix} \times \begin{Bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{Bmatrix} \longrightarrow \begin{Bmatrix} (v_2) \\ (v_3) \\ v_4 \\ \vdots \\ v_{115} \end{Bmatrix} & 
\end{array}$$

Previously computed vectors are within parentheses. Of course, we can only generate new orthogonal vectors as long as the number of vectors is smaller than the dimension of that mode. Further, if at a certain stage in the procedure we have generated  $\nu$  and  $\mu$  vectors in two modes, then we can generate altogether  $\lambda = \nu\mu$  vectors in the third mode (where we do not count the starting vector in that mode, if there was one).

The algorithm can be written (somewhat informally) as follows.

---

### Maximal Krylov recursion

---

```

 $\hat{h}w_1 = \mathcal{A} \cdot (u_1, v_1)_{1,2}$ 
 $\nu = \mu = \lambda = 1$ 
while  $\nu \leq \nu_{\max}$  and  $\mu \leq \mu_{\max}$  and  $\lambda \leq \lambda_{\max}$  do
  %— u-loop —%
  for all  $(\bar{\mu}, \bar{\lambda})$  such that  $\bar{\mu} \leq \mu$  and  $\bar{\lambda} \leq \lambda$  do
    if the pair  $(\bar{\mu}, \bar{\lambda})$  has not been used before then
       $\nu = \nu + 1$ 
       $h_u = \mathcal{A} \cdot (U_{\nu-1}, v_{\bar{\mu}}, w_{\bar{\lambda}})$ 
       $h_{\nu\bar{\mu}\bar{\lambda}}u_\nu = \mathcal{A} \cdot (v_{\bar{\mu}}, w_{\bar{\lambda}})_{2,3} - U_{\nu-1}h_u$ 
       $\mathcal{H}(:, \bar{\mu}, \bar{\lambda}) = \begin{pmatrix} h_u \\ h_{\nu\bar{\mu}\bar{\lambda}} \end{pmatrix}$  % Mode 1
    end if
  end for
  %— v-loop —%
  for all  $(\bar{\nu}, \bar{\lambda})$  such that  $\bar{\nu} \leq \nu$  and  $\bar{\lambda} \leq \lambda$  do
    if the pair  $(\bar{\nu}, \bar{\lambda})$  has not been used before then
       $\mu = \mu + 1$ 
       $h_v = \mathcal{A} \cdot (u_{\bar{\nu}}, V_{\mu-1}, w_{\bar{\lambda}})$ 
       $h_{\bar{\nu}\mu\bar{\lambda}}v_\mu = \mathcal{A} \cdot (u_{\bar{\nu}}, w_{\bar{\lambda}})_{1,3} - V_{\mu-1}h_v$ 
       $\mathcal{H}(\bar{\nu}, :, \bar{\lambda}) = \begin{pmatrix} h_v \\ h_{\bar{\nu}\mu\bar{\lambda}} \end{pmatrix}$  % Mode 2
    end if
  end for

```

```

    end if
  end for
  %— w-loop —%
  for all  $(\bar{\nu}, \bar{\mu})$  such that  $\bar{\nu} \leq \nu$  and  $\bar{\mu} \leq \mu$  do
    if the pair  $(\bar{\nu}, \bar{\mu})$  has not been used before then
       $\lambda = \lambda + 1$ 
       $h_w = \mathcal{A} \cdot (u_{\bar{\nu}}, v_{\bar{\mu}}, W_{\lambda-1})$ 
       $h_{\bar{\nu}\bar{\mu}\lambda} w_\lambda = \mathcal{A} \cdot (u_{\bar{\nu}}, v_{\bar{\nu}})_{1,2} - W_{\lambda-1} h_w$ 
       $\mathcal{H}(\bar{\nu}, \bar{\mu}, :) = \begin{pmatrix} h_w \\ h_{\bar{\nu}, \bar{\mu}, \lambda} \end{pmatrix}$  % Mode 3
    end if
  end for
end while

```

---

The algorithm has three main loops, and it is maximal in the sense that in each such loop we generate as many new vectors as can be done, before proceeding to the next main loop. Consider the  $u$ -loop (the other loops are analogous). The vector  $h_u$  is a mode-1 vector of dimension  $\nu - 1$  (we here refer to the identification (7)). The orthonormalization coefficients are collected in a tensor  $\mathcal{H}$ . Each time a new  $u$ -vector has been computed, the dimension of the first mode is increased by one, by filling out with a zero at the bottom of each 1-mode fibre (except the fibre at position  $(\bar{\mu}, \bar{\lambda})$ ). Continuing with the  $v$ -loop, the dimension of the coefficient tensor  $\mathcal{H}$  increases in the second mode.

It is clear that  $\mathcal{H}$  has a zero-nonzero structure that resembles that of a Hessenberg matrix. In fact matricizing<sup>3</sup>  $\mathcal{H}$  in any mode will yield a Hessenberg matrix, i.e. the matricizations  $H^{(1)}$ ,  $H^{(2)}$  and  $H^{(3)}$  are all of Hessenberg form.

If we break the recursion after any complete **for all**-statement, we can interpret it as a Krylov factorization.

**Theorem 4.1** (Tensor Krylov factorizations). *Given the tensor  $\mathcal{A} \in \mathbb{R}^{J \times K \times L}$  and two starting vectors. Assume that by a maximal Krylov procedure we have generated matrices with orthonormal columns, and a tensor  $\mathcal{H}$  of orthonormalization coefficients. Assume that after a complete  $u$ -loop the matrices  $U_j$ ,  $V_k$ , and  $W_l$ , and the tensor  $\mathcal{H}_{jkl} \in \mathbb{R}^{j \times k \times l}$ , have been generated, where  $j \leq J$ ,  $k \leq K$ , and  $l \leq L$ . Then*

$$\mathcal{A} \cdot (V_k, W_l)_{2,3} = (U_j)_1 \cdot \mathcal{H}_{jkl}. \quad (11)$$

---

<sup>3</sup>Any tensor can be reshaped to a matrix with suitable dimensions. An  $m \times n \times l$  tensor  $\mathcal{A}$  can be reshaped to matrices  $A^{(1)} \in \mathbb{R}^{m \times nl}$ ,  $A^{(2)} \in \mathbb{R}^{n \times ml}$  and  $A^{(3)} \in \mathbb{R}^{l \times mn}$ , etc, consider [9, 3, 6, 21] for details.

Further, assume that after a complete  $v$ -loop the tensor is  $\mathcal{H}_{jml} \in \mathbb{R}^{j \times m \times l}$ , and the matrices have corresponding dimensions. Then

$$\mathcal{A} \cdot (U_j, W_l)_{1,3} = (V_m)_2 \cdot \mathcal{H}_{jml}. \quad (12)$$

Similarly, with the corresponding assumptions, after a complete  $w$ -loop,

$$\mathcal{A} \cdot (U_j, V_m)_{1,2} = (W_n)_3 \cdot \mathcal{H}_{jmn}. \quad (13)$$

*Proof.* We prove that (11) holds; the other two equations are analogous. Using the definition of matrix-tensor multiplication we see that  $\mathcal{A} \cdot (V_k, W_l)_{2,3}$  is a tensor in  $\mathbb{R}^{J \times k \times l}$ , where the 1-fibre at any position  $(\mu, \lambda)$  is given by  $\mathcal{A} \cdot (v_\mu, w_\lambda)_{2,3}$ .

The 1-fibre at position  $(\mu, \lambda)$  of  $\mathcal{H}$  is equal to

$$\begin{pmatrix} \mathcal{A} \cdot (u_1, v_\mu, w_\lambda) \\ \mathcal{A} \cdot (u_2, v_\mu, w_\lambda) \\ \vdots \\ \mathcal{A} \cdot (u_{\nu-1}, v_\mu, w_\lambda) \\ h_{\nu\mu\lambda} \\ 0 \end{pmatrix},$$

which shows that (11) is equivalent to the expression for  $u_\nu$  in the algorithm.  $\square$

Let  $U_j$  and  $V_k$  be two matrices with orthonormal columns that have been generated by any tensor Krylov method (i.e., not necessarily a maximal one) with tensor  $\mathcal{A}$ . Assume that we then generate a sequence of  $m = jk$  vectors  $(w_1, w_2, \dots, w_m)$  as in the  $w$ -loop of the maximal method. From the proof of Theorem 4.1 we see that we have a tensor-Krylov factorization of the type (13),

$$\mathcal{A} \cdot (U_j, V_k)_{1,2} = (W_m)_3 \cdot \mathcal{H}_{jkm}. \quad (14)$$

### 4.3 Krylov Subspaces for Contracted Tensor Products

Recall that in the Golub-Kahan bidiagonalization procedure columns of the generated matrices  $U_k, V_k$  are orthonormal basis vectors for Krylov subspaces for  $AA^\top$  and  $A^\top A$  respectively. In tensor notation those products may be written as

$$\langle A, A \rangle_{-1} = AA^\top, \quad \langle A, A \rangle_{-2} = A^\top A.$$

For a third order tensor  $\mathcal{A} \in \mathbb{R}^{m \times n \times l}$ , and starting vectors  $u \in \mathbb{R}^m, v \in \mathbb{R}^n, w \in \mathbb{R}^l$  we may consider the matrix Krylov subspaces

$$\begin{aligned} \mathcal{K}_k(\langle \mathcal{A}, \mathcal{A} \rangle_{-1}, u), & \quad \langle \mathcal{A}, \mathcal{A} \rangle_{-1} = A^{(1)}(A^{(1)})^\top \in \mathbb{R}^{m \times m} \\ \mathcal{K}_k(\langle \mathcal{A}, \mathcal{A} \rangle_{-2}, v), & \quad \langle \mathcal{A}, \mathcal{A} \rangle_{-2} = A^{(2)}(A^{(2)})^\top \in \mathbb{R}^{n \times n} \\ \mathcal{K}_k(\langle \mathcal{A}, \mathcal{A} \rangle_{-3}, w), & \quad \langle \mathcal{A}, \mathcal{A} \rangle_{-3} = A^{(3)}(A^{(3)})^\top \in \mathbb{R}^{l \times l} \end{aligned}$$

The expressions to the right in each equation are matricized tensors, see [9, 3, 6, 21]. In this case we reduce a third order tensor to three different matrices, for which we compute the usual matrix subspaces through the Lanczos recurrence. This can be done without computing the matrices explicitly, thus taking advantage of sparsity. The result of such a procedure is three sets of orthonormal basis vectors for each of the modes of the tensor, collected in  $U_j, V_k, W_l$ , say. A low-rank approximation of the tensor can then be obtained using Lemma 2.1.

## 5 A Krylov-Schur-like Method

In this section we discuss the use of a tensor-Krylov method to compute an approximation to the solution of the best low-rank approximation of a tensor, in the Tucker sense, i.e., the solution of the problem

$$\begin{aligned} \min_{S, X, Y, Z} \|\mathcal{A} - (X, Y, Z) \cdot \mathcal{S}\|, \\ \text{subject to } X^\top X = I, Y^\top Y = I, Z^\top Z = I. \end{aligned} \quad (15)$$

The matrices  $X$ ,  $Y$ , and  $Z$  are assumed to be rectangular, so that the solution of (15) gives a low-rank approximation. In [9] we describe a Newton-Grassmann method for solving (15). This and other similar methods [11] are designed for small and medium-size problems for dense (non-sparse) tensors, and cannot be used efficiently for large and sparse problems.

The truncated Higher Order SVD [6] (HOSVD) gives an approximation of the tensor of the type  $\mathcal{A} \approx (X, Y, Z) \cdot \mathcal{S}$ . It orders the information in the tensor in the sense that it concentrates the mass of the tensor  $\mathcal{S}$  close to the position  $(1, 1, 1)$ , but it does not solve the best approximation problem (15).

### 5.1 Krylov-Schur Methods for Matrices

Restarted Arnoldi methods are used for the computation of eigenvalues and eigenvectors of large, sparse matrices<sup>4</sup> [16, 18]. These methods were further developed in [19, 20]; as those algorithms are based on computing a Schur decomposition of the matrix of orthonormalization coefficients, they are called Krylov-Schur methods.

In the method of [19] a Krylov-Schur factorization of the form

$$AU = US + ub^\top, \quad (16)$$

is computed, where  $S$  is triangular. Then the factorization is partitioned

$$A(U_1 \ U_2) = (U_1 \ U_2) \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} + u(b_1^\top \ b_2^\top); \quad (17)$$

---

<sup>4</sup>For instance, the MATLAB functions `eigs` and `svds` are implementations of implicitly restarted Arnoldi methods.

obviously

$$AU_1 = U_1 S_{11} + ub_1^\top, \quad (18)$$

has the same structure as (16). The basic idea of the restarted eigenvalue algorithm is to compute (17), and make sure that the approximations of the wanted eigenvalues are in  $S_{11}$ . Then, one truncates the decomposition and restarts the Krylov sequence from (18). By repeating this procedure, increasingly more accurate approximations of the wanted eigenvalues are computed.

An alternative method would be to continue the Krylov sequence until the eigenvalue approximations in  $S$  are accurate enough. However, the restarted approach has the advantage that the column dimension of  $U$  remains relatively small, thus keeping the storage requirements low.

## 5.2 Sketch of a Restarted Krylov Approach for Tensors

Assume that we have computed a low-rank approximation of a tensor

$$\mathcal{A} \approx (U_j, V_k, W_l) \cdot \mathcal{S}, \quad \mathcal{S} \in \mathbb{R}^{j \times k \times l},$$

by a tensor-Krylov method, using either Lemma 2.1 to compute  $\mathcal{S}$  or a factorization from Theorem 4.1. Then a best approximation in the sense of (15) of  $\mathcal{S}$ ,

$$\mathcal{S} \approx (X_\nu, Y_\mu, Z_\lambda) \cdot \tilde{\mathcal{S}}, \quad \tilde{\mathcal{S}} \in \mathbb{R}^{\nu \times \mu \times \lambda}, \quad (19)$$

is computed, where  $\nu < j$ ,  $\mu < k$ , and  $\lambda < l$ . Enlarge  $X_\nu$ ,  $Y_\mu$ , and  $Z_\lambda$  to orthogonal matrices (as the dimensions of the matrices is small, this enlargement is cheap),

$$X = (X_\nu \ X_\nu^\perp), \quad Y = (Y_\mu \ Y_\mu^\perp), \quad Z = (Z_\lambda \ Z_\lambda^\perp),$$

and make the change of bases,

$$\mathcal{A} \approx (\hat{U}_j, \hat{V}_k, \hat{W}_l) \cdot \hat{\mathcal{S}},$$

where  $\hat{U}_j = U_j X$ ,  $\hat{V}_k = V_k Y$ , and  $\hat{W}_l = W_l Z$ , and  $\hat{\mathcal{S}} = (X^\top, Y^\top, Z^\top) \cdot \mathcal{S}$ . Truncate the basis vectors to column dimensions  $\nu$ ,  $\mu$ , and  $\lambda$ , respectively, and the tensor  $\hat{\mathcal{S}}$  to  $\hat{\mathcal{S}}_{\text{trunc}} \in \mathbb{R}^{\nu \times \mu \times \lambda}$ . Finally, restart the Krylov procedure from  $\hat{U}_\nu$ ,  $\hat{V}_\mu$ , and  $\hat{U}_\lambda$  and the tensor  $\hat{\mathcal{S}}_{\text{trunc}}$ .

Variations of this refinement and restarting procedure are possible. For instance, one may use the HOSVD instead of the best approximation (19).

## 6 Numerical Examples

The purpose of the examples in this section is to make a preliminary investigation into the usefulness of the concepts proposed. Unfortunately, due to

$p \times q$	$20 \times 40$	$30 \times 60$	$40 \times 80$
THOSVD	29.6	24.7	21.1
$\langle \mathcal{D}, \mathcal{D} \rangle_{-i}$	33.9	28.8	24.9
$\mathcal{K}^{\min}(\mathcal{D}, u, v, w)$	39.5	32.8	27.4
$\mathcal{K}^{\max}(\mathcal{D}, u, v, w)$	35.5	31.5	28.4

Table 1: Relative error (percentage) of three different low rank tensor approximations for the digit tensor  $\mathcal{D}$  using four different methods.

time constraints, we have not had access to any large and sparse tensors. Instead the experiments are made mainly with a relatively small and dense tensor with data from the classification of handwritten digits, [17].

There are several details in the Krylov procedures that we have not touched upon earlier in this paper, but which are necessary to deal with in the actual implementation of the methods. One such algorithmic detail is concerned with the (common) case when the dimension of one particular mode is much smaller than those of the other modes. If one applies the methods as described above then one may have a complete orthogonal matrix for one mode before the bases of the other modes are good enough in some sense. One possible choice for generating new vectors in the other modes is to use the basis vectors of the low-dimensional mode in a cyclic manner. At this stage, since the purpose of this study is only to make a preliminary investigation of the methods, we avoid going into too much details here.

## 6.1 Tensor approximation

We have performed tests using the handwritten digits from the US postal service database. Digits from the database are formed into a tensor  $\mathcal{D}$  of dimensions  $400 \times 1194 \times 10$ . The first mode of the tensor represents pixels<sup>5</sup>, the second mode represents the variation within the different classes and the third mode represents the different classes. We want to find low dimensional subspaces  $U_p$  and  $V_q$  in the first and second mode, respectively, so that we obtain a good approximation of the original tensor, i.e.

$$\mathbb{R}^{400 \times 1194 \times 10} \ni \mathcal{D} \approx (U_p, V_q)_{1,2} \cdot \mathcal{F} \equiv \mathcal{D}_{p,q,10}. \quad (20)$$

An important difference compared to the presentation in the previous sections is that here we want to find only two of three matrices. The class mode of the tensor is not reduced to lower rank, i.e. we are computing a rank- $(p, q, 10)$  approximation of  $\mathcal{D}$ .

Table 1 shows the relative error,  $\|\mathcal{D} - \mathcal{D}_{p,q,10}\|/\|\mathcal{D}\|$ , for four methods and three different low rank approximations. Given the HOSVD

<sup>5</sup>Each digit is smoothed and reshaped to a vector.

$$\mathcal{D} = (U, V, W) \cdot \mathcal{C},$$

the truncated HOSVD subspaces are given by  $U_p = U(:, 1 : p)$  and  $V_q = V(:, 1 : q)$ . In the second row of Table 1 the subspaces

$$\mathcal{K}_p(\langle \mathcal{D}, \mathcal{D} \rangle_{-1}, u_0) \quad \text{and} \quad \mathcal{K}_q(\langle \mathcal{D}, \mathcal{D} \rangle_{-2}, v_0)$$

are generated using a Lanczos recursion of the symmetric matrices  $\langle \mathcal{D}, \mathcal{D} \rangle_{-1}$  and  $\langle \mathcal{D}, \mathcal{D} \rangle_{-2}$ . These matrices need not to be formed explicitly in order to obtain the sought subspaces. The starting vectors  $u_0$  and  $v_0$  were chosen as the means of the first and second mode fibers, respectively, of  $\mathcal{D}$ . The subspaces in the third row are generated with the minimal, and those in the last with the maximal Krylov procedures<sup>6</sup>.

We observe that the truncated HOSVD gives the best approximation, but the Krylov methods are not much worse.

## 6.2 Restarted Krylov Methods

In section 5 we describe a restarted Krylov approach to refine initial subspaces obtained from a tensor Krylov (or other) procedure. Figure 1 illus-

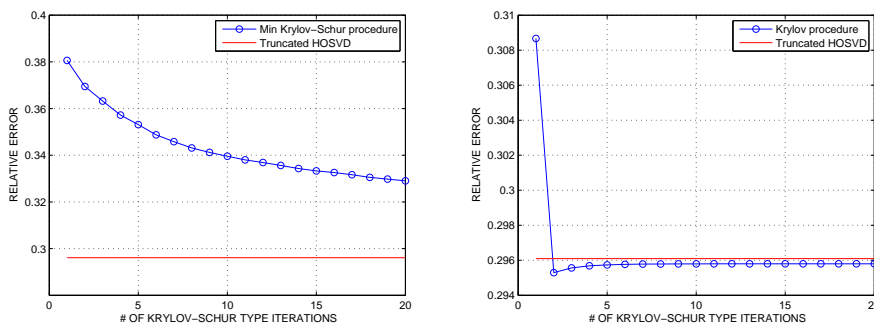


Figure 1: Restarted Krylov methods for tensor approximation. The relative approximation error is illustrated as a function of the number of restarts. Left—Minimal Krylov procedure. Right—Maximal Krylov procedure.

trates two experiments, one using the minimal Krylov procedure (left) and one using the maximal Krylov procedure (right). Given the digit tensor  $\mathcal{D}$ , the goal is to generate first and second mode Krylov subspaces  $U_{20}$  and  $V_{40}$  so that we get a good approximation

$$\mathcal{D} \approx (U_{20}, V_{40})_{1,2} \cdot \mathcal{F}.$$

<sup>6</sup>The procedure in this step involves truncated HOSVD approximation in order to obtain the specified dimensions of the subspaces.

The setup of the tests is as follows. Generate ten-dimensional subspaces  $U_{10}$  and  $V_{10}$ <sup>7</sup>. In the minimal Krylov procedure we extend  $U_{10}$  and  $V_{10}$  to  $U_{60}$  and  $W_{60}$  using the algorithm given in Section 4.1. Then we compute a core tensor  $\mathcal{F}$  using Lemma 2.1. Next we compute a low rank approximation of  $\mathcal{F}$  by a truncated HOSVD, i.e.

$$\mathbb{R}^{60 \times 60 \times 10} \ni \mathcal{F} = (X, Y, Z) \cdot \mathcal{C}, \quad (21)$$

and reduce  $U_{60}$  and  $V_{60}$  to subspaces of specified dimensions through

$$U_{20} = U_{60}X(:, 1 : 20), \quad \text{and} \quad V_{40} = V_{60}Y(:, 1 : 40).$$

An alternative to the truncated HOSVD would be to compute the best low rank-(20, 40, 10) approximation of  $\mathcal{F}$ . The dimensions of the tensor at this step are much smaller than the dimensions of the original tensor, therefore we can afford more expensive algorithms. Then the Krylov procedure is restarted and the matrices  $U_{20}$  and  $V_{40}$  are extended with 30 new vectors, using the minimal Krylov procedure, to  $U_{50}$  and  $V_{70}$ , after which a low-rank approximation is computed, etc. The relative error before each restart is shown in left part of Figure 1.

The restarted maximal Krylov procedure is similar. Given matrices  $U_{20}$  and  $V_{40}$  (and orthogonal matrix  $W_{10}$ ), the first and second mode subspaces are extended by taking all combinations  $V_{40} \times W_{10} \rightarrow U_{400}$  and  $U_{20} \times W_{10} \rightarrow V_{200}$  using expression similar to (14). Combining the old and newly generated matrices  $U = (U_{20} \ U_{400})$  and  $V = (V_{40} \ V_{200})$  we compute a truncated HOSVD approximation of  $\mathcal{A} \cdot (U, V)_{1,2}$ . The orthogonal matrices from the HOSVD are used to make a change of basis in  $U$  and  $V$ , which are then truncated to dimensions 20 and 40, respectively, and the procedure is restarted.

In Figure 1 we also indicate the relative error of the HOSVD approximation of the same dimensions of the tensor. We see that the maximal procedure quickly gives a better approximation than HOSVD. Unfortunately the subspaces generated by the restarted maximal Krylov procedure do not seem to improve after the restarts.

### 6.3 Consistency of Subspaces

In order to investigate if the subspaces generated with different starting vectors differ very much, we performed tests on the digit tensor, which is  $400 \times 1194 \times 10$  tensor, and random starting vectors. Five pairs of subspaces were generated with random starting vectors in (both modes). Table 2 summons the principal angles between subspaces computed with the minimal Krylov procedure and Table 3 summons the principal angles computed with

---

<sup>7</sup>The procedure also involves  $W_{10}$ , but since the third mode dimension of  $\mathcal{D}$  is equal to ten we get we cannot extend it further.

	$U_{(2)}$	$U_{(3)}$	$U_{(4)}$	$U_{(5)}$	$V_{(2)}$	$V_{(3)}$	$V_{(4)}$	$V_{(5)}$	
$U_{(1)}$	0.09	0.11	0.10	0.10	0.64	0.70	0.65	0.68	$V_{(1)}$
$U_{(2)}$		0.10	0.11	0.10		0.72	0.71	0.74	$V_{(2)}$
$U_{(3)}$			0.10	0.10			0.68	0.68	$V_{(3)}$
$U_{(4)}$				0.12				0.63	$V_{(4)}$

Table 2: The table shows the mean value of the 10 smallest principal angles between subspaces generated with different starting vectors. The subspaces are generated with *minimal* tensor Krylov procedure. Left—First mode subspace  $U$ . Right—Second mode subspace.

	$U_{(2)}$	$U_{(3)}$	$U_{(4)}$	$U_{(5)}$	$V_{(2)}$	$V_{(3)}$	$V_{(4)}$	$V_{(5)}$	
$U_{(1)}$	0.17	0.17	0.18	0.18	0.11	0.10	0.11	0.10	$V_{(1)}$
$U_{(2)}$		0.16	0.15	0.18		0.12	0.09	0.11	$V_{(2)}$
$U_{(3)}$			0.20	0.19			0.11	0.11	$V_{(3)}$
$U_{(4)}$				0.20				0.11	$V_{(4)}$

Table 3: The table shows the mean value of the 10 smallest principal angles between subspaces generated with different starting vectors. The subspaces are generated with *maximal* tensor Krylov procedure. Left—First mode subspace  $U$ . Right—Second mode subspace.

the maximal Krylov procedure. No restarts were made. For example, in the minimal Krylov procedure the mean of the ten smallest principal angles between subspaces  $U_{(2)}$  and  $U_{(3)}$  is 0.10 (boxed).

Ideally we wish to obtain subspaces that are close to each other (indicated with small principal angles) regardless the choice of starting vector. Using the minimal procedure we get small angles when comparing first mode subspaces. But the second mode subspaces are differing much more, indicating the need to take larger dimensional subspaces or restart the Krylov procedure. The results from the maximal Krylov procedure are more satisfying due to smaller principal angles between subspaces generated with different starting vectors.

## 6.4 Handwritten Digit Classification

In this section we will shortly describe how a low rank tensor approximation may be used in a context for classifying handwritten digits. For details consider [17]. Given an approximation as in equation (20), which is illustrated in Figure 2, we want save memory usage and computation by only using the small core tensor  $\mathcal{F}$  for constructing the models for the classes. These savings may be huge, since usually the size of  $\mathcal{F}$  is a very small fraction (less than one percent) of the size of  $\mathcal{D}$ . The models for the different classes are simply the dominant low dimensional subspaces of the ten class slices

of  $\mathcal{F}$ . A unknown digit is classified by first projecting it down to  $U_p$  and then determining (by a new projection to the ten class-specific subspaces) the subspace that gives the smallest residual. The dimension of the class-specific subspaces are typically very small—usually in the order 5-15.

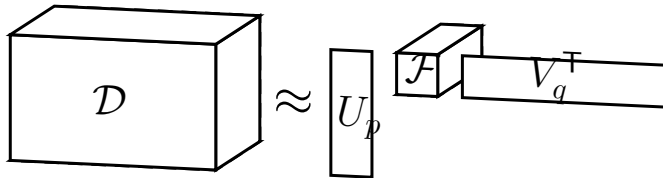


Figure 2: Illustration of low-rank tensor approximation for the digit tensor. Reduction is made in pixel and digit mode but not in the class mode.

Figure 3 makes a comparison in classification rate when the subspaces  $U_p$  and  $V_q$  in tensor approximation (20) are given by the methods described in the paper, i.e. truncated HOSVD, minimal and maximal Krylov procedure and Krylov spaces of contracted tensor products. The classification rate are

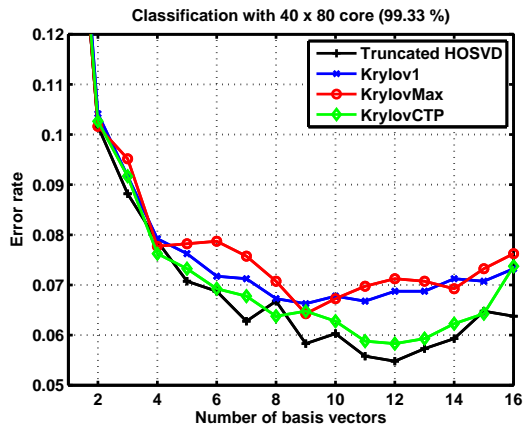


Figure 3: Misclassification rate for handwritten digits using subspaces generated with various methods. Tests with 2-16 dimensional subspaces of  $\mathcal{F}(:, :, i)$ ,  $i = 1, \dots, 10$  are illustrated.

quite similar for all methods suggesting the usefulness of all the methods.

## 7 Conclusions and Future Work

In this paper we propose several ways in which one may generalize matrix Krylov methods to tensors. It is demonstrated that some of the matrix concepts can be generalized, in particular the Krylov factorization, which can be used for computing a low-rank approximation of a tensor (in the

Tucker sense). We also propose a generalization of the Krylov-Schur method for the computation of an approximate best low-rank approximation of a large and sparse tensor.

Our very preliminary numerical results indicate that the Krylov methods are useful for low-rank approximation of larger tensor.

As the research on tensor Krylov methods is in a very early stage, there are numerous questions that need to be answered, and which will be the subject of our continued research. We have hinted to some in the text; here we list a few others.

- What are the convergence properties?
- How to construct a method that gives an intermediate between the minimal and maximal sequence?
- Which variant is more economical in terms of the number of tensor-vector operations, taking into account the convergence rate?
- If one of the modes is of small dimension, so that a complete basis is quickly computed, how can one modify the recursion so that one can use the already computed information as efficiently as possible?

## References

- [1] B. Bader, R. Harshman, and T. Kolda. Temporal analysis of social networks using three-way DEDICOM. Technical Report SAND2006-2161, Sandia National Laboratories, Albuquerque, NM, 2006.
- [2] B. Bader and T. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32:635–653, 2006.
- [3] B. W. Bader and T. G. Kolda. Efficient matlab computations with sparse and factored tensors. Technical report, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006.
- [4] Brett W. Bader, Richard A. Harshman, and Tamara G. Kolda. Temporal analysis of semantic graphs using ASALSAN. In *ICDM 2007: Proceedings of the 7th IEEE International Conference on Data Mining*, pages 33–42, October 2007.
- [5] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali. Cross-language information retrieval using PARAFAC2. Technical Report SAND2007-2706, Sandia National Laboratories, Albuquerque, NM, 2007.

- [6] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000.
- [7] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.*, 30(3):1084–1127, 2008.
- [8] Daniel M. Dunlavy, Tamara G. Kolda, and W. Philip Kegelmeyer. Multilinear algebra for analyzing data with multiple linkages. Technical Report SAND2006-2079, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, April 2006.
- [9] L. Eldén and B. Savas. A Newton–Grassmann method for computing the best multi-linear rank- $(r_1, r_2, r_3)$  approximation of a tensor. Technical Report LITH-MAT-R-2007-6-SE, Department of Mathematics, Linköping University, 2007. To appear in *SIAM J. Matrix Anal. Appl.*
- [10] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal. Ser. B*, 2:205–224, 1965.
- [11] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel. Dimensionality reduction for higher-order tensors: algorithms and applications. Internal Report 07-187, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2007. submitted to: *International Journal of Pure and Applied Mathematics*.
- [12] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry*. Interscience Publisher, 1963.
- [13] T. Kolda and B. Bader. The TOPHITS model for higher-order web link analysis. In *Workshop on Link Analysis, Counterterrorism and Security*, 2006.
- [14] T. Kolda, B. Bader, and J. Kenny. Higher-order web link analysis using multilinear algebra. In *Proc. 5th IEEE International Conference on Data Mining, ICDM05*, pages 27–30. IEEE Computer Society Press, 2005.
- [15] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 50:00–00, 2008, to appear.
- [16] R. Lehoucq, D. Sorensen, and C. Yang. *Arpack Users’ Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998.

- [17] B. Savas and L. Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition*, 40:993–1003, 2007.
- [18] D.C. Sorensen. Implicit application of polynomial filters in a  $k$ -step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13:357–385, 1992.
- [19] G. W. Stewart. *Matrix Algorithms II: Eigensystems*. SIAM, Philadelphia, 2001.
- [20] G. W. Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002.
- [21] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. 7th European Conference on Computer Vision (ECCV'02)*, Lecture Notes in Computer Science, Vol. 2350, pages 447–460, Copenhagen, Denmark, 2002. Springer Verlag.