MULTILINEAR SINGULAR VALUE DECOMPOSITION FOR STRUCTURED TENSORS

ROLAND BADEAU* AND RÉMY BOYER†

Abstract. The Higher-Order SVD (HOSVD) is a generalization of the Singular Value Decomposition (SVD) to higher-order tensors (*i.e.* arrays with more than two indices) and plays an important role in various domains. Unfortunately, this decomposition is computationally demanding. Indeed, the HOSVD of a third-order tensor involves the computation of the SVD of three matrices, which are referred to as "modes", or "matrix unfoldings". In this paper, we present fast algorithms for computing the full and the rank-truncated HOSVD of third-order structured (symmetric, Toeplitz and Hankel) tensors. These algorithms are derived by considering two specific ways to unfold a structured tensor, leading to structured matrix unfoldings whose SVD can be efficiently computed[1].

Key words. Multilinear SVD, fast algorithms, structured and unstructured tensors.

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction. The subject of multilinear decomposition is now mature [5, 19]. There are essentially two families. The first one is known under the name of CANDECOMP/PARAFAC (CANonical DECOMPosition or PARAllel FACtors model) and was independently proposed in [4, 8]. This decomposition is very useful in several applications and is linked to the tensor rank [9]. The second one is related to the multidimensional rank [6] and is known under the name of Tucker decomposition [21]. This decomposition is a more general form which is often used. Orthogonality constraints are not required in the general Tucker decomposition but if needed one can refer to the Higher-Order Singular Value Decomposition (HOSVD) [6] or multilinear SVD.

The HOSVD is a generalization of the SVD to higher-order tensors (*ie.* arrays with more than two indices). This decomposition plays an important role in various domains, such as harmonic retrieval [17], image processing [10], telecommunications, biomedical applications (magnetic resonance imaging and electrocardiography), web search [20], computer facial recognition [23], handwriting analysis [18], statistical methods involving Independent Component Analysis (ICA) [6].

In [14], it was shown that the HOSVD of a third-order tensor involves the computation of the SVD of three matrices called modes, leading to a high computational cost. A first approach for reducing the complexity of tensor-based methods consists in a dimensionality reduction: only the principal components of the HOSVD are calculated, leading to the *rank-truncated* HOSVD. In this paper, we present a standard and fast algorithm for calculating the full and the rank-truncated HOSVD, which only computes the left factors of the three SVD's. Next, we focus on structured tensors, such as symmetric and Toeplitz tensors, which naturally arise in signal processing methods involving higher-order statistics [11, Chapter 9], and Hankel tensors [17], introduced in the context of the Harmonic Retrieval problem [15], which is at the heart of many signal processing applications. To the best of our knowledge, there are no specific HOSVD algorithms proposed in the literature for exploiting tensors

*GET - Télécom Paris (ENST), département TSI, 46 rue Barrault, 75634 Paris Cedex 13, France, Email: roland.badeau@enst.fr.

†Laboratoire des Signaux et Systèmes (LSS), CNRS, Université Paris XI (UPS), SUPELEC, Gif-Sur-Yvette, France, Email: remy.boyer@lss.supelec.fr.

[1]This work has been partially presented at the IEEE ICASSP 2006 Conference [3].

structures. In this paper however, we show that such tensors can be efficiently decomposed. We first observe that standard unfoldings [14, 12] do not present a particularly noticeable structure even in the case of structured tensors. Consequently, we introduce two different ways to unfold a structured tensor which clarify the link between structured modes and structured tensors. By doing this, we can exploit fast product techniques [7]. A second point of this work concerns Hankel and symmetric tensors. The modes of these structured tensors are column-redundant so it is possible to reduce the computational cost of the HOSVD algorithm by taking the redundant structure of each mode into account. Finally, our fastest implementation of the rank-truncated HOSVD (dedicated to Hankel tensors) has a quasilinear complexity with respects to the tensor dimension.

Note that for applications involving very large tensor dimensions, an even lower complexity may be required. In this case, one may be interested in rank-revealing tensor decompositions which can be computed faster than the rank-truncated HOSVD. Such an approach is developed in [16], based on cross approximation techniques which are derived from LU factorizations [22, 2]. An algorithm is proposed which provides a Tucker-like low rank approximation of unstructured cube tensors, the complexity of which is linear with respects to the tensor dimension in many cases [16]. This linear complexity is nevertheless obtained via an approximated rank reduction, in comparison with an exact Tucker decomposition such as the HOSVD.

2. Preliminaries in multilinear algebra. We present some basic definitions in the context of third-order tensor algebra. These definitions can be extended to order greater than three and we refer the interested reader to [5, 6] for instance.

Tucker's product. The Tucker's product, also called s-mode product, of a third-order complex-valued tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ by a matrix $B \in \mathbb{C}^{J_s \times I_s}$ for $s \in [1:3]$ is defined according to:

$$[\mathcal{A} \times_1 B]_{j_1 i_2 i_3} = \sum_{i_1=0}^{I_1-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_1 i_1}, \tag{2.1}$$

$$[\mathcal{A} \times_2 B]_{i_1 j_2 i_3} = \sum_{i_2=0}^{I_2-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_2 i_2}, \tag{2.2}$$

$$[\mathcal{A} \times_3 B]_{i_1 i_2 j_3} = \sum_{i_3=0}^{I_3-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_3 i_3}, \tag{2.3}$$

where we denoted the entries of $\mathcal{A}$ by $[\mathcal{A}]_{i_1 i_2 i_3}$ with $i_s \in \{0 \dots I_s - 1\}$. We have the following properties:

$$\mathcal{A} \times_s B \times_{s'} C = (\mathcal{A} \times_s B) \times_{s'} C = (\mathcal{A} \times_{s'} C) \times_s B, \tag{2.4}$$

$$(\mathcal{A} \times_s B) \times_s C = \mathcal{A} \times_s (BC). \tag{2.5}$$

Mode of a tensor. There are several ways to represent an $I_1 \times I_2 \times I_3$ third-order complex-valued tensor $\mathcal{A}$ as a collection of matrices.

DEFINITION 2.1. *The modes (also called "matrix unfoldings") A_1, A_2, A_3 are usually defined as follows:*

$$[A_1]_{i_1,i_2I_3+i_3} = [\mathcal{A}]_{i_1i_2i_3}, \tag{2.6}$$

$$[A_2]_{i_2,i_3I_1+i_1} = [\mathcal{A}]_{i_1i_2i_3}, \tag{2.7}$$

$$[A_3]_{i_3,i_1I_2+i_2} = [\mathcal{A}]_{i_1i_2i_3}. \tag{2.8}$$

These matrices are of dimension $(I_1 \times I_2I_3)$, $(I_2 \times I_3I_1)$, $(I_3 \times I_1I_2)$, respectively.

The dimensions of the vector spaces generated by the columns of the modes of $\mathcal{A}$ are called column rank (or 1-mode rank) R_1, row rank (or 2-mode rank) R_2 and 3-mode rank R_3, respectively.

2.1. Multilinear SVD (HOSVD).

THEOREM 2.2 (Third-Order SVD [6, 21]).
Every $I_1 \times I_2 \times I_3$ tensor $\mathcal{A}$ can be written as the product:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \tag{2.9}$$

where $\times_s$ represents the Tucker s-mode product [6], $U^{(s)}$ is an unitary $I_s \times I_s$ matrix and $\mathcal{S}$ is an all-orthogonal and ordered $I_1 \times I_2 \times I_3$ tensor. All-orthogonality means that the matrices $S_{i_s=\alpha}$, obtained by fixing the s^{th} index to α, are mutually orthogonal with respect to (w.r.t.) the standard inner product. Ordering means that $\|S_{i_s=0}\| \geqslant \|S_{i_s=1}\| \geqslant \ldots \geqslant \|S_{i_s=I_s-1}\|$ for all possible values of s. The Frobenius-norms $\|S_{i_s=i}\|$, symbolized by $\sigma_i^{(s)}$, are the s-mode singular values of $\mathcal{A}$ and the columns of $U^{(s)}$ are the s-mode singular factors.

This decomposition is a generalization of the SVD because the diagonality of the matrix containing the singular values, in the matrix case, is a special case of all-orthogonality. Also, the HOSVD of a second-order tensor (matrix) yields the matrix SVD, up to trivial indeterminacies. The matrix of s-mode singular factors, $U^{(s)}$, can be found as the matrix of left singular vectors of the mode A_s, defined in (2.6)–(2.8). The s-mode singular values correspond to the singular values of this matrix unfolding. Note that the s-mode singular factors of a tensor, corresponding to the nonzero s-mode singular values, form an orthonormal basis for its s-mode vector subspace, like in the matrix case.

The core tensor $\mathcal{S}$ can then be computed (if needed) by bringing the matrices of s-mode singular factors to the left side of equation (2.9):

$$\mathcal{S} = \mathcal{A} \times_1 {U^{(1)}}^H \times_2 {U^{(2)}}^H \times_3 {U^{(3)}}^H \tag{2.10}$$

where $(.)^H$ denotes the conjugate transpose.

Mode decompositions. Expression (2.9) can be written in terms of modes as follows:

$$A_1 = U^{(1)} S_1 \left(U^{(3)} \otimes U^{(2)}\right)^H,$$

$$A_2 = U^{(2)} S_2 \left(U^{(3)} \otimes U^{(1)}\right)^H,$$

$$A_3 = U^{(3)} S_3 \left(U^{(1)} \otimes U^{(2)}\right)^H,$$

where $\otimes$ denotes the Kronecker product and S_1, S_2 and S_3 denote respectively the first, second and third modes of the core tensor $\mathcal{S}$.

2.2. HOSVD algorithm for unstructured tensors. In this section we present an efficient implementation of the HOSVD in the general framework of unstructured tensors, from which our fast algorithms for structured tensors will be derived in section 4. Let $I = \frac{1}{3}(I_1 + I_2 + I_3)$. The computational costs of the various algorithms presented below are related to the flop (*floating point operation*) count. For example, a dot product of I-dimensional vectors involves $2I$ flops (I multiplications plus I additions).

The calculation of the HOSVD of tensor $\mathcal{A}$ requires the computation, for all $s \in [1:3]$, of the left factor $U^{(s)}$ in the full SVD of matrix A_s, as defined above. Note that in many applications, we are interested in computing the HOSVD truncated at ranks (M_1, M_2, M_3), which means that we only compute the M_s first columns of the matrix $U^{(s)}$ (M_s is often supposed to be much lower than I_s). We will suppose throughout this paper that this possibly truncated SVD is computed by means of the orthogonal iteration method, although other algorithms such as the Golub-Reinsch SVD and R-SVD [7, pp. 253–254] could also be applied. When computing only the $n \times r$ left factor U in the rank r-truncated SVD of an $n \times m$ matrix A with $n < m$, the orthogonal iteration method consists in recursively computing the $n \times r$ matrix $B_i = A(A^H U_{i-1})$, involving $2r$ matrix / vector products, and the QR factorization $B_i = U_i R_i$ of this $n \times r$ matrix [7, pp. 410–411]. Thus the computational cost of one iteration is $2r\, c(n,m) + 2r^2 n$ flops, where $c(n,m) = 2nm$ is the cost of 1 matrix / vector product, and $2r^2 n$ is the cost of 1 QR factorization [7, pp. 231–232]. Besides, the s-mode A_s has $n = I_s$ rows and $m = \prod_{s' \neq s} I_{s'}$ columns. Assuming that $\prod_{s' \neq s} I_{s'}$ is much greater than I_s, the dominant cost of one iteration for computing $U^{(s)}$ is $4 M_s I_1 I_2 I_3$ flops. Finally, the Tucker product (2.10) can be computed by folding for instance its first mode given by

$$S_1 = {U^{(1)}}^H A_1 \left(U^{(3)} \otimes U^{(2)} \right). \tag{2.11}$$

A fast implementation of equation (2.11) was proposed in [1], whose complexity is $6 M_s I_1 I_2 I_3$ flops, where $M = \frac{1}{3}(M_1 + M_2 + M_3)$. Note that the computation of the Tucker product is generally not needed in applications, this is why it will be omitted in the following developments.

The computational cost of the full and rank-truncated HOSVD is summarized in table 2.1 (the full HOSVD is the same as the rank-truncated HOSVD with $M_s = I_s$ for all $s = 1, 2, 3$). In this table and below, the global cost is provided as a maximum w.r.t. I_1, I_2, I_3, under the constraint $I_1 + I_2 + I_3 = 3I$. In particular, the maximal complexity per iteration is obtained for cube tensors ($I_1 = I_2 = I_3 = I$) and equals $12 M I^3$.

Table 2.1
HOSVD Algorithm for unstructured tensors
(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of A_1	$4 M_1 I_1 I_2 I_3$
SVD of A_2	$4 M_2 I_1 I_2 I_3$
SVD of A_3	$4 M_3 I_1 I_2 I_3$
Global cost	$12 M I^3$

3. Structured tensors and reordered tensor modes.

In this section, we present three tensor structures which are usual in many applications. Next, we introduce new reordered tensor modes which clarify the link between structured tensors and structured modes.

3.1. Structured tensors.

DEFINITION 3.1 (Toeplitz tensors). *A* Toeplitz *tensor is a structured tensor which satisfies the following property: for all* $i_1 \in \{0 \dots I_1 - 1\}$, $i_2 \in \{0 \dots I_2 - 1\}$, $i_3 \in \{0 \dots I_3 - 1\}$, $\forall k \in \{0 \dots \min(I_1 - i_1, I_2 - i_2, I_3 - i_3) - 1\}$,

$$[\mathcal{A}]_{i_1+k, i_2+k, i_3+k} = [\mathcal{A}]_{i_1 i_2 i_3}.$$

Below, any permutation of 3 elements will be denoted $\pi = (\pi_1, \pi_2, \pi_3)$ where $\pi_1, \pi_2, \pi_3 \in \{1, 2, 3\}$, according to the following definition

$$\pi : \quad (i_1, i_2, i_3) \quad \mapsto \quad (i_{\pi_1}, i_{\pi_2}, i_{\pi_3}).$$

DEFINITION 3.2 (Symmetric tensors). *A cube* $(I \times I \times I)$ *tensor* $\mathcal{A}$ *which is unchanged by any permutation* π *is called a* symmetric *tensor:*

$$\forall i_1, i_2, i_3 \in \{0, \dots, I-1\}, [\mathcal{A}]_{\pi(i_1, i_2, i_3)} = [\mathcal{A}]_{i_1 i_2 i_3}.$$

EXAMPLE 1 (Fast higher PCA for real moment and cumulant).

*The HOSVD can be viewed (*cf.* reference [13]) as a higher Principal Component Analysis (PCA). This technique is often used as a data dimensional reduction for moment and cumulant tensors [6]. Third-order moment and cumulant tensors are defined according to*

$$[\mathcal{M}]_{t_1 t_2 t_3} = E\{x(t_1)x(t_2)x(t_3)\}, \tag{3.1}$$

$$\begin{aligned} [\mathcal{C}]_{t_1 t_2 t_3} = \quad & E\{x(t_1)x(t_2)x(t_3)\} + 2E\{x(t_1)\}E\{x(t_2)\}E\{x(t_3)\} \\ - \quad & E\{x(t_1)\}E\{x(t_2)x(t_3)\} - E\{x(t_2)\}E\{x(t_1)x(t_3)\} - E\{x(t_3)\}E\{x(t_1)x(t_2)\}. \end{aligned} \tag{3.2}$$

where $t_1, t_2, t_3 \in \{0 \dots I-1\}$, *and* $x(t)$ *is a real random process.*

Moment and cumulant tensors, defined in (3.1) and (3.2), are symmetric tensors according to definition 3.2. The proof is straightforward and can be generalized to larger orders [5]. Moreover, if $x(t)$ *is a third-order stationary process, the moment and cumulant tensors defined in (3.1) and (3.2) are third-order Toeplitz tensors according to definition 3.2. Indeed, if* $x(t)$ *is a stationary process, its probability distribution is invariant to temporal translations. This property implies* $[\mathcal{C}]_{t+i_1, t+i_2, t+i_3} = [\mathcal{C}]_{i_1 i_2 i_3}$.

DEFINITION 3.3 (Hankel tensors). *A Hankel tensor is a structured tensor whose coefficients* $[\mathcal{A}]_{i_1 i_2 i_3}$ *only depend on* $i_1 + i_2 + i_3$.

Note that a cube Hankel tensor is symmetric. Hankel tensors were introduced in [17] in the context of the Harmonic Retrieval problem [15]. This problem is at the heart of many signal processing applications.

EXAMPLE 2 (Definition and properties of the harmonic model).

We consider the complex harmonic model defined according to:

$$x_n = \sum_{m=1}^{M} \alpha_m z_m^n, \quad \textit{for} \quad n \in [0 : N-1] \tag{3.3}$$

where N is the analysis duration and M is the known number of components, $z_m = e^{\delta_m + i\phi_m}$ is called the m^{th} pole of x_n where $i = \sqrt{-1}$, ϕ_m is called the m^{th} angular-frequency belonging to $(-\pi, \pi]$, and δ_m is the m^{th} damping factor. In the sequel, we assume that all the poles are distinct. In addition, $\alpha_m = a_m e^{i\phi_m}$ is the non-zero m^{th} complex amplitude, ie.*, $a_m \neq 0, \forall m$. Besides, we define the Vandermonde matrices $Z^{(I_1)}$, $Z^{(I_2)}$ and $Z^{(I_3)}$ associated to model* (3.3)*, according to* $[Z^{(I_s)}]_{n,m} = z_m^n$*, and we assume that* $M \leq \min(I_1, I_2, I_3)$*. Then the Hankel tensor $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$ associated to model (3.3) is diagonalizable according to*

$$\mathcal{A} = \mathcal{D} \times_1 Z^{(I_1)} \times_2 Z^{(I_2)} \times_3 Z^{(I_3)}$$

where $\times_i$ denotes the i-th Tucker's product and

$$[\mathcal{D}]_{jk\ell} = \begin{cases} \alpha_j & \textit{if } j = k = \ell \\ 0 & \textit{otherwise} \end{cases}$$

is a hyper-cubic $M \times M \times M$ super-diagonal core tensor. As a consequence, the Hankel tensor $\mathcal{A}$ is a rank-(M, M, M) tensor. Following standard subspace-based parametric estimation methods, the harmonic model can then be estimated by computing the rank M-truncated HOSVD of tensor $\mathcal{A}$ [17].

3.2. Modes of structured tensors.

As mentioned in the introduction, standard unfoldings of structured tensors [14, 12] do not present a particularly noticeable structure. Consequently, we introduce in this section two different ways to unfold a structured tensor which clarify the link between structured modes and structured tensors.

Example 3. *Consider the $4 \times 4 \times 4$ symmetric and Toeplitz tensor* $[\mathcal{A}]_{ijk} = 3 * \max(i, j, k) - \mathrm{sum}(i, j, k)$*. The classical 1-mode is formed of 4 symmetric sub-matrices:*

$$A_1 = \left[\begin{bmatrix} 0 & 2 & 4 & 6 \\ 2 & 1 & 3 & 5 \\ 4 & 3 & 2 & 4 \\ 6 & 5 & 4 & 3 \end{bmatrix} \begin{bmatrix} 2 & 1 & 3 & 5 \\ 1 & 0 & 2 & 4 \\ 3 & 2 & 1 & 3 \\ 5 & 4 & 3 & 2 \end{bmatrix} \begin{bmatrix} 4 & 3 & 2 & 4 \\ 3 & 2 & 1 & 3 \\ 2 & 1 & 0 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 6 & 5 & 4 & 3 \\ 5 & 4 & 3 & 2 \\ 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \right].$$

However, by permuting its columns, we define an other mode A'_1 (referred to below as the type-1 reordered tensor mode*), which is formed of 4 Toeplitz matrices T_0, T_1, T_2 and T_3 (referred to below as the* type-1 oblique sub-matrices*):*

$$A'_1 = \left[\underbrace{\begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \end{bmatrix}}_{T_3} \underbrace{\begin{bmatrix} 4 & 5 \\ 3 & 4 \\ 2 & 3 \\ 4 & 2 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \\ 5 & 3 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 2 \\ 4 & 2 & 0 & 1 \\ 6 & 4 & 2 & 0 \end{bmatrix}}_{T_0} \underbrace{\begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \\ 5 & 3 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 4 & 5 \\ 3 & 4 \\ 2 & 3 \\ 4 & 2 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \end{bmatrix}}_{T_3} \right].$$

It can be noted that this reordered mode satisfies an axial blockwise symmetry with respect to its central oblique sub-matrix T_0. Obviously, the left singular factor $U^{(1)}$ in *the SVD of A_1 is the same as the left singular factor in the SVD of A_1', since both matrices have the same columns. However, we will show below that the SVD of A_1' can be computed efficiently, by exploiting the Toeplitz structure of the oblique sub-matrices T_k.*

EXAMPLE 4. *Consider the $4\times4\times4$ Hankel tensor $[\mathcal{A}]_{ijk} = i+j+k$. The standard 1-mode is formed of 4 Hankel sub-matrices:*

$$A_1 = \left[\begin{bmatrix} 0&1&2&3\\1&2&3&4\\2&3&4&5\\3&4&5&6\end{bmatrix} \begin{bmatrix} 1&2&3&4\\2&3&4&5\\3&4&5&6\\4&5&6&7\end{bmatrix} \begin{bmatrix} 2&3&4&5\\3&4&5&6\\4&5&6&7\\5&6&7&8\end{bmatrix} \begin{bmatrix} 3&4&5&6\\4&5&6&7\\5&6&7&8\\6&7&8&9\end{bmatrix}\right].$$

However, by permuting its columns, we define an other mode A_1'' (referred to below as the type-2 reordered tensor mode*), which is formed of 7 rank-1 matrices $R_0 \dots R_6$ (referred to below as the* type-2 oblique sub-matrices*):*

$$A_1'' = \left[\underbrace{\begin{bmatrix}0\\1\\2\\3\end{bmatrix}}_{R_0} \underbrace{\begin{bmatrix}1&1\\2&2\\3&3\\4&4\end{bmatrix}}_{R_1} \underbrace{\begin{bmatrix}2&2&2\\3&3&3\\4&4&4\\5&5&5\end{bmatrix}}_{R_2} \underbrace{\begin{bmatrix}3&3&3&3\\4&4&4&4\\5&5&5&5\\6&6&6&6\end{bmatrix}}_{R_3} \underbrace{\begin{bmatrix}4&4&4\\5&5&5\\6&6&6\\7&7&7\end{bmatrix}}_{R_4} \underbrace{\begin{bmatrix}5&5\\6&6\\7&7\\8&8\end{bmatrix}}_{R_5} \underbrace{\begin{bmatrix}6\\7\\8\\9\end{bmatrix}}_{R_6}\right].$$

Again, the left singular factor in the SVD of A_1 is the same as the left singular factor in the SVD of A_1'', since both matrices have the same columns. However, we will show below that the SVD of A_1'' can be computed efficiently, by exploiting the rank-1 structure of the oblique sub-matrices R_k, and the Hankel structure of the matrix obtained by removing the repeated columns in A_1''.

In the following section, the type-1 and type-2 oblique sub-matrices will be defined in the general case by "slicing" a third-order tensor according to 2 different oblique directions, as shown in Fig. 3.1.

3.2.1. Oblique sub-matrices of a tensor.

DEFINITION 3.4 (Type-1 and type-2 oblique sub-matrices of a tensor).

For any permutation π, the oblique sub-matrices of a tensor $\mathcal{A}$ are defined as follows:

- *For all $k \in \{0 \dots I_{\pi_3}-1\}$, let $J^{(1)}_{(\pi_2,\pi_3)}(k) = \min(I_{\pi_2}, I_{\pi_3}-k)$. The coefficients of the k^{th} type-1 oblique $I_{\pi_1} \times J^{(1)}_{(\pi_2,\pi_3)}(k)$ sub-matrix of $\mathcal{A}$ are*

$$[T_k^{(\pi)}]_{ij} = [\mathcal{A}]_{\pi^{-1}(i,j,k+j)} \tag{3.4}$$

  where $0 \le i \le I_{\pi_1}-1$ and $0 \le j \le J^{(1)}_{(\pi_2,\pi_3)}(k)-1$.
- *For all $k \in \{0 \dots I_{\pi_2}+I_{\pi_3}-2\}$, let*

$$J^{(2)}_{(\pi_2,\pi_3)}(k) = \min(I_{\pi_2}, I_{\pi_3}, 1+k, I_{\pi_2}+I_{\pi_3}-1-k). \tag{3.5}$$

  The coefficients of the $I_{\pi_1} \times J^{(2)}_{(\pi_2,\pi_3)}(k)$ type-2 oblique sub-matrix of $\mathcal{A}$ are

$$[R_k^{(\pi)}]_{ij} = [\mathcal{A}]_{\pi^{-1}(i,\ \max(k-I_{\pi_3}+1,0)+j,\ \min(k,I_{\pi_3}-1)-j)} \tag{3.6}$$

  where $0 \le i \le I_{\pi_1}-1$ and $0 \le j \le J^{(2)}_{(\pi_2,\pi_3)}(k)-1$.

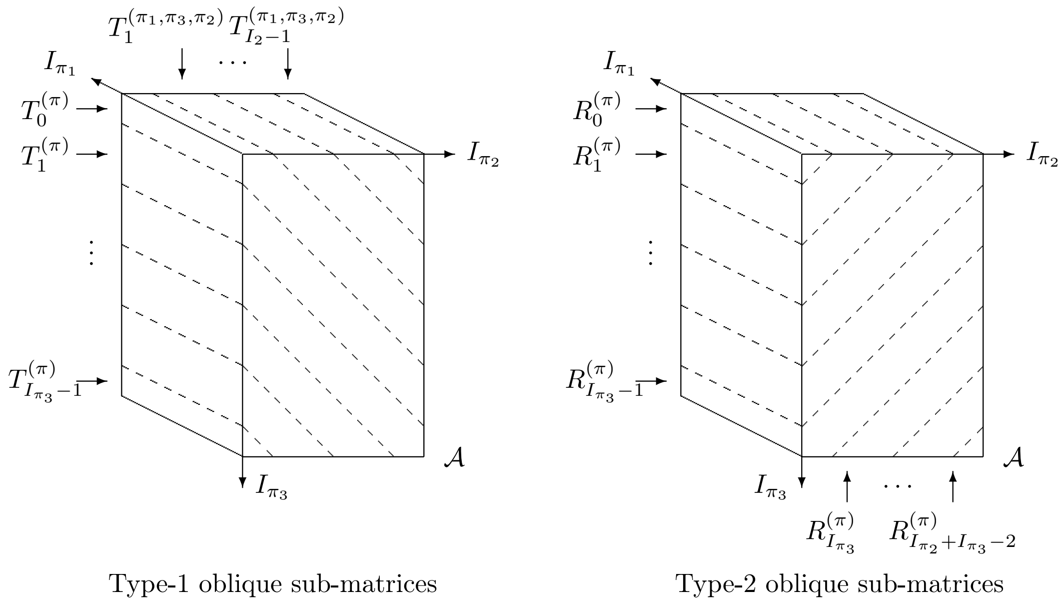

FIG. 3.1. *Type-1 and type-2 oblique sub-matrices of a tensor*

PROPOSITION 3.5.
1. *If $\mathcal{A}$ is an $(I \times I \times I)$ symmetric tensor, then $\forall k \in \{0 \dots I-1\}$, all type-1 oblique sub-matrices $T_k^{(\pi)}$ are equal (*i.e. *$\forall k$, $T_k^{(\pi)} = T_k$ does not depend on π).*
2. *If $\mathcal{A}$ is a Toeplitz tensor, then for all permutation π and index $k \in \{0 \dots I_{\pi_3}-1\}$, the type-1 oblique sub-matrix $T_k^{(\pi)}$ is Toeplitz.*
3. *If $\mathcal{A}$ is a Hankel tensor, all columns of the type-2 oblique sub-matrix $R_k^{(\pi)}$ are equal.*

Proof.
1. If the tensor $\mathcal{A}$ is symmetric, equation (3.4) yields $[T_k^{(\pi)}]_{ij} = [\mathcal{A}]_{i,j,k+j}$, which does not depend on π.
2. Applying equation (3.4) to $i+1$ and $j+1$ (for all $0 \leq i < I_{\pi_1}-1$ and $0 \leq j < J^{(1)}_{(\pi_2,\pi_3)}(k)-1$) yields $[T_k^{(\pi)}]_{i+1,j+1} = [\mathcal{A}]_{\pi^{-1}(i+1,j+1,k+j+1)}$. However, since the tensor $\mathcal{A}$ is Toeplitz, $[\mathcal{A}]_{\pi^{-1}(i+1,j+1,k+j+1)} = [\mathcal{A}]_{\pi^{-1}(i,j,k+j)} = [T_k^{(\pi)}]_{ij}$. Therefore $[T_k^{(\pi)}]_{i+1,j+1} = [T_k^{(\pi)}]_{ij}$, which means that the matrix T_k is Toeplitz.
3. If $[\mathcal{A}]_{i_1 i_2 i_3}$ is of the form $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$, equation (3.6) shows that for all permutation π and index $k \in \{0 \dots I_{\pi_2}+I_{\pi_3}-2\}$, $[R_k^{(\pi)}]_{ij} = x_{(i+k)}$ does not depend on j.

□

3.2.2. Reordered tensor modes.
Below we introduce the type-1 and type-2 reordered tensor modes, formed by concatenating the type-1 and type-2 oblique sub-matrices.

DEFINITION 3.6.
The type-1 reordered tensor modes *are defined by concatenating the type-1 oblique sub-matrices:*
- *A'_1 is the $I_1 \times (I_2 I_3)$ matrix $[T_{I_2-1}^{(1,3,2)}, \dots, T_0^{(1,3,2)} = T_0^{(1,2,3)}, \dots, T_{I_3-1}^{(1,2,3)}]$,*

- *A'_2 is the $I_2 \times (I_3 I_1)$ matrix* $[T^{(2,1,3)}_{I_3-1}, \ldots, T^{(2,1,3)}_{0} = T^{(2,3,1)}_{0}, \ldots, T^{(2,3,1)}_{I_1-1}]$,
- *A'_3 is the $I_3 \times (I_1 I_2)$ matrix* $[T^{(3,2,1)}_{I_1-1}, \ldots, T^{(3,2,1)}_{0} = T^{(3,1,2)}_{0}, \ldots, T^{(3,1,2)}_{I_2-1}]$.

In the same way, the type-2 reordered tensor modes *are defined by concatenating the type-2 oblique sub-matrices:*

- *A''_1 is the $I_1 \times (I_2 I_3)$ matrix* $[R^{(1,2,3)}_{0}, \ldots, R^{(1,2,3)}_{I_2+I_3-2}]$,
- *A''_2 is the $I_2 \times (I_3 I_1)$ matrix* $[R^{(2,3,1)}_{0}, \ldots, R^{(2,3,1)}_{I_3+I_1-2}]$,
- *A''_3 is the $I_3 \times (I_1 I_2)$ matrix* $[R^{(3,1,2)}_{0}, \ldots, R^{(3,1,2)}_{I_1+I_2-2}]$.

Proposition 3.7.

1. *For all $s = 1, 2, 3$, the mode A_s and the reordered modes A'_s, A''_s admit the same singular values and left singular vectors.*
2. *If $\mathcal{A}$ is a symmetric tensor, then $A'_1 = A'_2 = A'_3$, and this unique mode admits an axial blockwise symmetry w.r.t. its central oblique sub-matrix.*

Proof.

1. For all $s = 1, 2, 3$, the columns of the reordered modes A'_s and A''_s form a permutation of the columns of the mode A_s defined in section 2.
2. This is a corollary of point 2 in proposition 3.5.

□

4. Fast algorithms for computing the HOSVD of structured tensors. In this section, the reordered tensor modes introduced above are used to efficiently compute the HOSVD of structured tensors. The first improvement consists in exploiting the column-redundancy of symmetric and Hankel tensors. To further reduce the computational cost, we then exploit the fast matrix-vector product techniques specific to Toeplitz and Hankel matrices.

4.1. Algorithms exploiting column-redundancy. Here we suppose that the s-mode of tensor $\mathcal{A}$ is redundant, *e.g.* some columns of the s-mode are equal (this is the case of symmetric and Hankel tensors for instance). We aim at exploiting this redundancy in order to efficiently implement the HOSVD of $\mathcal{A}$. Toward this end, we define the $I_s \times J_s$ matrix H_s as the matrix obtained by removing the repeated columns in the s-mode ($J_s \leq \prod_{s' \neq s} I_{s'}$), and we denote $d^{(s)}_k$ the number of occurrences of the k^{th} column of H_s in the s-mode. Then we consider the $I_s \times I_s$ correlation matrix of the s-mode: $C^{(s)} = A_s \, {A_s}^H$. It is clear that this matrix can be factorized as

$$C^{(s)} = H_s \, D_s^2 H_s^H,$$

where

$$D_s = \operatorname{diag}\left(\sqrt{d^{(s)}_0} \ldots \sqrt{d^{(s)}_{J_s-1}}\right)$$

(if the s-mode is not redundant, then we define H_s as the s-mode itself and D_s is defined as the $J_s \times J_s$ identity matrix). As a consequence, the M_s highest singular values and left singular vectors of the s-mode of dimensions $I_s \times \prod_{s' \neq s} I_{s'}$ are the same as those of the smaller $I_s \times J_s$ matrix $H_s \, D_s$.

Algorithms for symmetric tensors. In the case of $(I \times I \times I)$ symmetric tensors, we proved in point 2 of proposition 3.7 that $A'_1 = A'_2 = A'_3$, and that this unique mode admits an axial blockwise symmetry. Therefore we can define

- the non-redundant matrix $H_s = [T^{(1,2,3)}_0, \ldots, T^{(1,2,3)}_{I-1}]$, $\forall s \in \{1, 2, 3\}$, of dimension $I \times J$ with $J = I(I+1)/2$;

- the weighting factors $d_k^{(s)} = \left\{ \begin{array}{l} 1 \text{ if } 0 \leq k < I \\ 2 \text{ if } I \leq k < J \end{array} \right.$.

In this way, the cost of the (rank-truncated) HOSVD is reduced to that of the (rank-truncated) SVD of $H_s D_s$, which is $2MI^3$ flops per iteration. In particular, it can be noted that the compression and weighting of the modes lead to a complexity 6 times as low as that of the algorithm in table 2.1.

Algorithms for Hankel tensors. In the case of $(I_1 \times I_2 \times I_3)$ Hankel tensors, $[\mathcal{A}]_{i_1 i_2 i_3}$ is of the form $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$, and we proved in point 3 of proposition 3.5 that for all permutation π and index $k \in \{0 \ldots I_{\pi_2} + I_{\pi_3} - 2\}$, $[R_k^{(\pi)}]_{ij} = x_{(i+k)}$. In particular, all columns of the type-2 oblique sub-matrix $R_k^{(\pi)}$ are equal. Therefore for each s-mode we can define

- the non-redundant Hankel matrix $H_s(i,k) = x_{(i+k)}$, of dimension $I_s \times J_s$ with $J_s = (\sum_{s' \neq s} I_{s'}) - 1$;
- the weighting factors $d_k^{(s)} = J^{(2)}_{\{\pi_{s'}\}_{s' \neq s}}(k) = \min(\{I_{s'}\}_{s' \neq s}, 1+k, J_s - k)$ (here $d_k^{(s)}$ is the number of columns of the k^{th} oblique sub-matrix of the s-mode, defined in equation (3.5)). It can be noted that the weighting function $1+k \mapsto d_k^{(s)}$ (plotted in Fig. 4.1) is piecewise linear:

$$d_k^{(s)} = \left\{ \begin{array}{rl} 1+k & \text{if } \; 1 \leq 1+k < \min(\{I_{s'}\}_{s' \neq s}) \\ \min(\{I_{s'}\}_{s' \neq s}) & \text{if } \; \min(\{I_{s'}\}_{s' \neq s}) \leq 1+k \leq \max(\{I_{s'}\}_{s' \neq s}) \\ J_s - k & \text{if } \; \max(\{I_{s'}\}_{s' \neq s}) < 1+k \; \leq J_s \\ 0 & \text{elsewhere.} \end{array} \right. \tag{4.1}$$

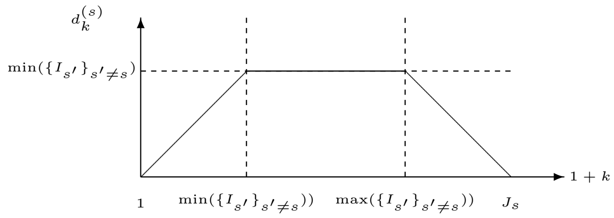

FIG. 4.1. *Weighting function $d_k^{(s)}$ for Hankel tensors*

The fast SVD-based algorithm for computing the full or rank-truncated HOSVD of the Hankel tensor $\mathcal{A}$ is summarized in table 4.1. The compression and weighting of

TABLE 4.1
Fast HOSVD algorithms for Hankel tensors
(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of $H_1\, D_1$	$4M_1 I_1(I_2 + I_3)$
SVD of $H_2\, D_2$	$4M_2 I_2(I_1 + I_3)$
SVD of $H_3\, D_3$	$4M_3 I_3(I_1 + I_2)$
Global cost	$24MI^2$

the modes allow a reduction of the complexity of one order of magnitude w.r.t. the algorithm in table 2.1. If additionally the Hankel tensor is cube ($I_1 = I_2 = I_3 = I$), then it is symmetric, and the three modes are equal. In this case, the global complexity is reduced $8MI^2$ flops.

Table 4.2
Fast HOSVD algorithm for Toeplitz tensors
(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of A'_1	$2M_1(90I^2\log_2(I) + M_1I_1)$
SVD of A'_2	$2M_2(90I^2\log_2(I) + M_2I_2)$
SVD of A'_3	$2M_3(90I^2\log_2(I) + M_3I_3)$
Global cost	$6M\ 90I^2\log_2(I)$

4.2. Algorithms exploiting the Toeplitz or Hankel structure. In the above developments, we assumed that the rank-r rank-truncated SVD of an $n \times m$ matrix with $n < m$ was computed by means of the orthogonal iteration method [7, pp. 410–411], which consists in recursively performing $2r$ matrix / vector products and 1 QR factorization of an $n \times r$ matrix (a full SVD corresponds to the case $r = n$). We mentioned that the computational cost of one iteration is $2r\,c(n, m) + 2r^2n$ flops, where $c(n, m)$ is the cost of 1 matrix / vector product, and $2r^2n$ is the cost of 1 QR factorization [7, pp. 231–232].

In the following, we will focus on the HOSVD of Toeplitz or Hankel tensors, which can be computed efficiently, using fast matrix / vector products. Indeed, the computational cost of a product between a $p \times q$ Toeplitz or Hankel matrix and a vector can be reduced from $2pq$ flops to $15(p + q)\log_2(p + q)$ flops, by means of Fast Fourier Transforms (FFT) [7, pp. 188-191,201–202].

Algorithms for Toeplitz tensors. In the case of Toeplitz tensors, we mentioned in point 2 of proposition 3.5 that for all permutation π and index $k \in \{0 \ldots I_{\pi_3} - 1\}$, the type-1 oblique sub-matrix $T_k^{(\pi)}$ is Toeplitz. Therefore the oblique modes A'_s are formed of Toeplitz blocks. As a consequence, the computational cost of the multiplication of A'_s by a vector of appropriate dimension can be reduced from $2I^3$ flops to $90I^2\log_2(I)$ flops[2]. By introducing those fast products into the orthogonal iteration method, the cost of the (rank-truncated) SVD of A'_s is reduced to $2M_s(90I^2\log_2(I) + M_sI_s)$ per iteration.

The fast algorithm for computing the full or rank-truncated HOSVD of a Toeplitz tensor is summarized in table 4.2. If additionally the tensor $\mathcal{A}$ is symmetric, then the three modes are equal. Moreover, as shown in section 4.1, the SVD of A'_s can be replaced by that of H_sD_s, where the $I \times \frac{I(I+1)}{2}$ matrix H_s is also block-Toeplitz. Therefore the cost of the SVD of H_sD_s is half that of the SVD of A'_s. As a consequence, the compression and weighting of the modes lead to a complexity 6 times as low as that of the fast HOSVD algorithm in table 4.2.

Algorithms for Hankel tensors. In the case of Hankel tensors, we noted in section 4.1 that the HOSVD could be obtained by computing the SVD of the matrices H_sD_s, where each compressed mode H_s is a Hankel matrix ($H_s(i, k) = x_{(i+k)}$). Therefore we can again use fast matrix-vector products to further reduce the complexity. More precisely, the computational cost of the multiplication of the $I_s \times ((\sum_{s' \neq s} I_{s'}) - 1)$

[2]Under the constraint $I_1 + I_2 + I_3 = 3I$, the maximum cost is obtained for cube tensors ($I_1 = I_2 = I_3 = I$). Besides, left or right multiplying an $I \times k$ oblique sub-matrix $T_{I-k}^{(\pi)}$ by a vector of appropriate dimension normally involves $2Ik$ flops. This complexity is reduced to $15(I + k)\log_2(I + k)$ flops by means of FFT's. Therefore left or right multiplying the block-Toeplitz matrix A'_s by a vector of appropriate dimension normally involves $2\sum_{k=0}^{I-1} 2Ik \sim 2I^3$ flops, or $2\sum_{k=0}^{I-1} 15(I + k)\log_2(I + k) \sim 90I^2\log_2(I)$ by means of FFT's.

Hankel matrix H_s by a vector of appropriate dimension can be reduced from $4I^2$ flops to $45I\log_2(I)$ flops, by means of FFT's. By introducing those fast products into the orthogonal iteration method, the cost of the SVD of H_sD_s is reduced to $2M_s(45I\log_2(I) + M_sI_s)$ per iteration.

The ultra-fast algorithm for computing the full or rank-truncated HOSVD of a Hankel tensor is summarized in table 4.3. Its global cost is provided as a maximum over M_1, M_2, M_3, under the constraint $M_1 + M_2 + M_3 = 3M$. It can be noted that the cost due to the fast matrix / vector products, and the cost due to the QR factorizations can be of the same order of magnitude if $M = O(\log_2(I))$.

Table 4.3
Ultra-fast HOSVD algorithm for Hankel tensors
(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of H_1D_1	$2M_1(45I\log_2(I) + M_1I_1)$
SVD of H_2D_2	$2M_2(45I\log_2(I) + M_2I_2)$
SVD of H_3D_3	$2M_3(45I\log_2(I) + M_3I_3)$
Global cost	$6M(45I\log_2(I) + MI)$

If additionally the Hankel tensor is cube ($I_1 = I_2 = I_3 = I$), then it is symmetric, and the three modes are equal. In this case, the global complexity is three times as low as that of the ultra-fast HOSVD algorithm in table 4.3.

4.3. Comparison of the complexities. The overall costs of the various HOSVD algorithms presented above are summarized in table 4.4 (sorted in decreasing order of complexity). Note that only the complexity upper bounds are given in this table, and that the calculation of the tensor $\mathcal{S}$ is not included. Besides, it can be noted that the FFT-based HOSVD algorithms are not always the fastest, because of the high constants in table 4.4. The best choice for computing the HOSVD actually depends on I, and possibly on M (the dominant cost of all algorithms is linear w.r.t M, except that of the ultra-fast algorithms for Hankel tensors). Fig. 4.2 represents the different complexities for $M = 10$. From this figure we can draw general remarks, which actually stand for any value of parameter M:

- the best algorithm for computing the HOSVD of Toeplitz (resp. symmetric Toeplitz) tensors is that dedicated to such tensors if $I \gtrsim 400$, or that dedicated to unstructured (resp. symmetric) tensors otherwise;
- the best algorithms for computing the HOSVD of symmetric, Hankel and cube Hankel tensors are always those dedicated to such tensors.

In other respects, the comparison between the fast and ultra-fast computations of the HOSVD for Hankel and cube Hankel tensors are sensitive to parameter M, as can be noted in table 4.4. Our simulations showed that:

- for small values of M ($M \ll I$), the ultra-fast algorithm is faster if $I \gtrsim 70$;
- for moderate values of M ($M \simeq I/2$), the ultra-fast algorithm is faster if $I \gtrsim 80$;
- for large values of M ($M \simeq I$), the ultra-fast algorithm is faster if $I \gtrsim 100$.

5. Conclusions. In this paper, we proposed to decrease the computational cost of the full or rank-truncated HOSVD, which is basically $O(MI^3)$, by exploiting the structure of symmetric, Toeplitz, and Hankel tensors. For symmetric and Hankel tensors, our solution is based on the fact that the HOSVD can be reduced to the SVD of three non-redundant (no column are repeated) matrices whose columns are

TABLE 4.4
Complexities of the HOSVD algorithms
(the cost corresponds to a single iteration of the orthogonal iteration method)

Structure	Global cost per iteration
unstructured	$12MI^3$
symmetric	$2MI^3$
Toeplitz (fast)	$540MI^2\log_2(I)$
symmetric Toeplitz (fast)	$90MI^2\log_2(I)$
Hankel (fast)	$24MI^2$
cube Hankel (fast)	$8MI^2$
Hankel (ultra-fast)	$270MI\log_2(I)+6M^2I$
cube Hankel (ultra-fast)	$90MI\log_2(I)+2M^2I$

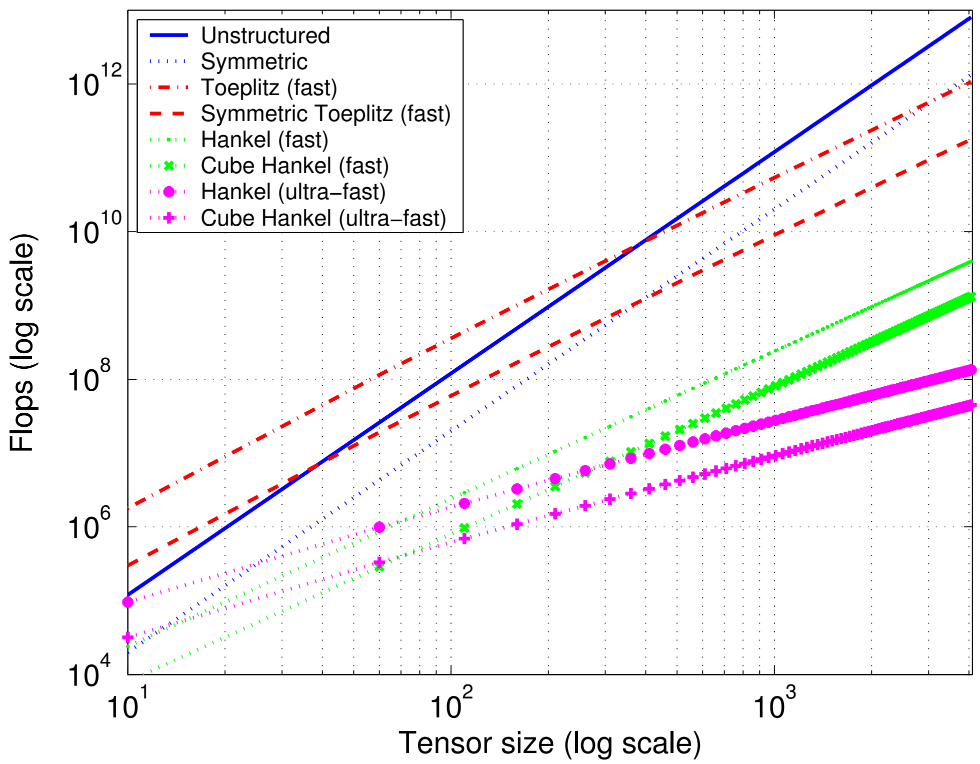

FIG. 4.2. *Flops count vs. size I for $M = 10$.*

multiplied by a given weighting function. In the case of Toeplitz and Hankel tensors, we propose a new way to perform the tensor unfolding which allows fast matrix / vector products. Finally, our fastest implementation of the HOSVD has a complexity of $O(MI\log_2(I))$ in the case of Hankel tensors.

REFERENCES

[1] C. ANDERSON AND R. BRO, *Improving the speed of multi-way algorithms: Part I. Tucker3*, Chemometrics and intelligent laboratory systems, 42 (1998), pp. 93–103.

[2] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.
[3] R. Boyer and R. Badeau, *Adaptive Multilinear SVD for Structured Tensors*, in Proc. of ICASSP'06, vol. 3, Toulouse, France, May 2006, IEEE, pp. 880–883.
[4] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.
[5] P. Comon, *Tensor Decompositions, state of the art and applications*, in IMA Conf. Mathematics in Signal Processing, Warwick, UK, Dec. 2000.
[6] L. De Lathauwer, *Signal Processing based on Multilinear Algebra*, PhD thesis, Katholieke Universiteit Leuven, 1997.
[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore and London, UK, third ed., 1996.
[8] R. A. Harshman and M. E. Lundy, *PARAFAC: Parallel factor analysis*, Computational Statistics and Data Analysis, 18 (1994), pp. 39–72.
[9] T. D. Howell, *Global properties of tensor rank*, Linear Algebra and Applications, 22 (1978), pp. 9–23.
[10] J. Huang, H. Wium, K. Qvist, and K. Esbensen, *Multi-way methods in image analysis - relationships and applications*, Chemometrics and intelligent laboratory systems, 66 (2003), pp. 141–158.
[11] T. Kailath and A. H. Sayed, eds., *Fast Reliable Algorithms for Matrices with Structure*, SIAM Press, Philadelphia, USA, 1999.
[12] H. Kiers, *Towards a standardized notation and terminology in multiway analysis*, Journal of Chemometrics, 14 (2000), pp. 105–122.
[13] P. Kroonenberg, *Three-Mode Principal Component Analysis: Theory and Applications*, DSWO Press, Leiden University, Faculty of Social and Behavioural Sciences, Dept. of Data Theory, Leiden, The Netherlands, 1983.
[14] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[15] X. Liu and N. Sidiropoulos, *Almost Sure Identifiability of Constant Modulus Multidimensional Harmonic Retrieval*, IEEE Trans. Signal Processing, 50 (2002), pp. 2366–2368.
[16] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction in linear time.* (submitted to SIMAX).
[17] J. Papy, L. De Lathauwer, and S. Van Huffel, *Exponential data fitting using multilinear algebra. The single-channel and the multichannel case*, Numerical Linear Algebra and Applications, 12 (2005), pp. 809–826.
[18] B. Savas, *Analyses and Tests of Handwritten Digit Recognition Algorithms*, master's thesis, Department of Mathematics, Linkoping University, Linkoping, 2003.
[19] N. Sidiropoulos, *Low-Rank Decomposition of Multi-Way Arrays: A Signal Processing Perspective*, in IEEE Workshop on Sensor Array and Multichannel processing (SAM2004), Barcelona, Spain, July 2004.
[20] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, *CubeSVD: A Novel Approach to Personalized Web Search*, in Proc. of the International World Wide Web Conference, Chiba, Japan, May 2005, pp. 382–390.
[21] L. Tucker, *Some Mathematical Notes on Three-mode Factor Analysis*, Psychometrika, 31 (1996), pp. 279–311.
[22] E. E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic–skeleton method*, Computing, 4 (2000), pp. 367–380.
[23] H. Wang and N. Ahuja, *Facial Expression Decomposition*, in Proc. of International Conference on Computer Vision (ICCV), Nice, France, Sept. 2003.