Article

Tensor Global Extrapolation Methods Using the n-Mode and the Einstein Products

Alaa El Ichi 1,2, Khalide Jbilou 2,3,* and Rachid Sadaka 1

1 LABMIA-SI, Department of Mathematics, University Mohammed V Rabat, Rabat 10000, Morocco; elichi.alaa@gmail.com (A.E.I.); r_sadaka@hotmail.com (R.S.)
2 Department of Mathematics LMFA, 50 rue F. Buisson, ULCO, 62100 Calais, France
3 Laboratory of Modeling Simulation, Mohammed VI Polytechnic University, Hay Moulay Rachid, Ben Guerir 43150, Morocco
* Correspondence: jbilou@univ-littoral.fr
Received: 9 June 2020; Accepted: 2 August 2020; Published: 5 August 2020

Abstract: In this paper, we present new Tensor extrapolation methods as generalizations of well known vector, matrix and block extrapolation methods such as polynomial extrapolation methods or \(\epsilon\)-type algorithms. We will define new tensor products that will be used to introduce global tensor extrapolation methods. We discuss the application of these methods to the solution of linear and nonlinear tensor systems of equations and propose an efficient implementation of these methods via the global-QR decomposition.

Keywords: Einstein product; Krylov subspaces; sequence transformation; tensor extrapolation

1. Introduction

Scalar extrapolation methods have been developed to accelerate the convergence of some sequences of numbers in \(\mathbb{R}\) or \(\mathbb{C}\). This process consists in transforming a sequence \((s_n)\) converging to \(s\) to new ones by applying some transformation \(T_k, k = 1, \ldots\) defined as follows

\[T_k : s_n \rightarrow T_k^{(n)} = s_n + \sum_{i=1}^{k-1} a_i^{(n)} g_i^{(n)},\]

where \(g_i^{(n)}, i = 1, \ldots, k-1\) is a scalar sequence that defines the method. One of the earlier extrapolation method is the well known Aitken’s \(\Delta^2\) process [1] defined by

\[T_2^{(n)} = s_n - \frac{\Delta s_n}{\Delta^2 s_n},\]

where the first and the second forward differences are defined by \(\Delta s_n = s_{n+1} - s_n\) and \(\Delta^2 s_n = \Delta s_{n+1} - \Delta s_n\) and in that case we have \(g_1^{(n)} = \Delta s_n\). Such a process was first proposed in 1926 by Aitken in [2]. It is well known that under some assumptions, the new Aitken sequence \(T_2^{(n)}\) will converge faster than \((s_n)\) to the same limit \(s\); see [1] for more details. Vector extrapolation methods have been proposed the last decades and among them are the minimal polynomial extrapolation (MPE) [3,4], the reduced rank extrapolation (RRE) method [5] and the modified minimal polynomial extrapolation (MMPE) [6,7]. For more details, see [1,8–12]. A second class of vector sequence transformations contains the topological \(\epsilon\)-algorithm (TEA) [6]. Applications for solving large linear and nonlinear systems of equations have been considered in [8,13].

Vector extrapolation methods were used in many applications such as google page rank by Golub et al. [14,15] and in other fields such as in statistics [16] or for solving discretized Navier–Stokes problems [17]. In the present paper, we consider tensor sequence transformations and propose new tensor extrapolation methods that generalize the classical vector ones. Using the Einstein product, we...
define some new tensor products that will allow us to develop the new methods based on orthogonal or oblique projections onto subspaces of small dimensions. It will be shown that when the tensor sequence is generated linearly then the proposed methods are theoretically equivalent to some tensor Krylov subspace methods such as the tensor version of GMRES and Lanczos methods developed recently in [18,19].

The remainder of this paper is organized as follows. In Section 2, we give notations, some basic definitions and properties related to tensors. In Section 3, we introduce the tensor versions of the vector polynomial extrapolation methods namely the Tensor Global Reduced Rank Extrapolation (TG-RRE), the Tensor Global Minimal Polynomial Extrapolation (TG-MPE) and the Tensor Global Modified Minimal Polynomial Extrapolation (TG-MMPE). We also give a Global tensor version of the topological $\epsilon$-transformation. Section 4 describes the application of the proposed methods for solving linear and nonlinear tensor system of equations. In Section 5, we introduce efficient implementations via the tensor global-QR decomposition and in the last section, we present some numerical experiments.

2. Preliminaries and Notations

In this section, we briefly review some concepts and notions that are used throughout the paper. A tensor is a multidimensional array of data and a natural extension of scalars, vectors and matrices to a higher order, a scalar is a 0th order tensor, a vector is a 1th-order tensor and a matrix is 2th-order tensor. The tensor order is the number of its indices, which is called modes or ways. For a given N-mode tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, the notation $x_{i_1,\ldots,i_N}$ stands for element $(i_1,\ldots,i_N)$ of the tensor $X$. Corresponding to a given tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, the notation $X:,:,\ldots,:$ denotes a tensor in $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{N-1}}$ which is obtained by fixing the last index and is called frontal slice. Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Figure 1 shows the frontal, horizontal and lateral slices of a third order tensor and also a mode-3 tube fiber.

![Figure 1.](image)

The n-mode matrix of a tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted by $X_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N)}$ with $n = 1,\ldots,N$. We have:

$$X_{(n)}(i_{n,j}) = x_{i_1,\ldots,i_N}$$

where $j = 1 + \sum_{k=1,k\neq n}^{N}(i_k - 1)I_k$ with $I_k = \prod_{m=1,m\neq n}^{k-1}I_m$.

An important operation for a tensor is the tensor-matrix multiplication [20], also known as n-mode product of a tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $U \in \mathbb{R}^{I \times I_n}$, the n-mode product is a tensor of size $I_1 \times I_2 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$ whose entries are given by:

$$x_{i_1,\ldots,i_N}U = x_{i_1,\ldots,i_N}u_j$$
We recall the definition and some properties of the Einstein product, see [19].

Definition 1. Let $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_L \times K_1 \times \ldots \times K_N}$, $\mathcal{B} \in \mathbb{R}^{K_1 \times \ldots \times K_N \times I_1 \times \ldots \times I_M}$, the Einstein product of the tensors $\mathcal{A}$ and $\mathcal{B}$ is the tensor of size $(I_1 \times \ldots \times I_L \times I_1 \times \ldots \times I_M)$ defined as:

$$(\mathcal{A} \circ \mathcal{B})_{i_1 \ldots i_L 1 \ldots 1} = \sum_{k_1, k_2, \ldots, k_N=1}^{K_1, K_2, \ldots, K_N} a_{i_1 \ldots i_L k_1} b_{k_1 \ldots k_N}$$

Next, we give a proposition that will be used later.

The trace of the tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is given by $\text{tr}(\mathcal{A}) = \sum_{i_1 \ldots i_N} a_{i_1 \ldots i_N 1 \ldots 1}$. The inner product of two tensors of the same order $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is given by

$$\langle \mathcal{A}, \mathcal{B} \rangle = \text{tr}(\mathcal{B}^T \circ \mathcal{A}),$$

and the associated norm is defined by:

$$||\mathcal{A}||^2 = \text{tr}(\mathcal{A}^T \circ \mathcal{A}).$$

Definition 3 ([21]). A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ is called the inverse of the square tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_N \times I_1 \times \ldots \times I_N}$ and denoted by $\mathcal{A}^{-1}$ if it satisfies:

$$\mathcal{A} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{A} = I$$

where $I$ the identity tensor is called a unit tensor or identity tensor which all its entries are zero except for the diagonal entries $I_{i_1, \ldots, i_N 1, \ldots, 1} = 1$.

In the following, we introduce the new $\square^{[N+M+1]}$ product between two tensors which is a generalization of diamond product introduced in [22].

Definition 4. The $\square^{[N+M+1]}$ tensor product between two $(M + N + 1)$-mode tensors $\mathcal{X} = [X_1, \ldots, X_l] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times \ldots \times I_M}$ and $\mathcal{Y} = [Y_1, \ldots, Y_p] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times \ldots \times I_M}$ where $X_i \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times \ldots \times I_M}$ and $Y_j \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times \ldots \times I_M}$, is the $\ell \times p$ matrix whose $(i, j)$ entries, $i = 1, \ldots, \ell$ and $j = 1, \ldots, p$, is given by

$$(\mathcal{X} \square^{[N+M+1]} \mathcal{Y})_{i,j} = \langle X_i, Y_j \rangle = \text{tr}(Y_j^T \circ X_i).$$

Next, we give a proposition that will be used later.
Proposition 1. Assume that \( \mathcal{U} = [\mathcal{U}_1, \ldots, \mathcal{U}_m] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M} \) is an \((N + M + 1)\)-mode tensor such that \( \mathcal{U}_1, \ldots, \mathcal{U}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M} \) and let \( y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m \). Then for an arbitrary \((N + M + 1)\)-mode tensor \( \mathcal{Y} = [\mathcal{Y}_1, \ldots, \mathcal{Y}_m] \) such that \( \mathcal{Y}_1, \ldots, \mathcal{Y}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M} \), we have:

\[
\mathcal{Y} \boxtimes_{(N + M + 1)} (\mathcal{U} \boxtimes_{(N + M + 1)} y) = (\mathcal{Y} \boxtimes_{(N + M + 1)} (\mathcal{U}) \mathcal{Y}),
\]

Proof. The proof comes easily from the definition of the two involved tensor products. \( \Box \)

Definition 5. The set of \((N + M)\)-mode tensors \( \mathcal{U}_1, \ldots, \mathcal{U}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M} \) is called orthonormal if

\[
\langle \mathcal{U}_i, \mathcal{U}_j \rangle = \delta_{ij}(= 1 \text{ if } i = j \text{ and } 0 \text{ elsewhere}).
\]

Remark 1. Suppose that \( \mathcal{U} = [\mathcal{U}_1, \ldots, \mathcal{U}_m] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M} \) is an \((N + M + 1)\)-mode tensor such that \( \mathcal{U}_1, \ldots, \mathcal{U}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_{K_1} \times K_2 \times \ldots \times K_M}. \) If the \((M + N)\)-mode tensors \( \mathcal{U}_1, \ldots, \mathcal{U}_m \) are orthonormal, then we have:

\[
\mathcal{U} \boxtimes_{(N + M + 1)} \mathcal{U} = I_m.
\]

3. Tensor Extrapolation Methods

In this section, we present two classes of new tensor global extrapolation methods. The first class contains the tensor polynomial based methods while the second class is devoted to the global tensor topological \( \epsilon \)-algorithm.

3.1. Tensor Global-Polynomial Extrapolation Methods

Let \((S_n)\) be a sequence of tensors of \( \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M}\) and consider the transformation \( T_k, k = 1, 2, \ldots \) from \( \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M} \) onto \( \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M} \) and defined by

\[
T_k^{(n)} = T_k(S_n) = S_n + \sum_{j=0}^{k-1} a_j^{(k)} G_j(n)
\]

\[= S_n + H_k(n) \tilde{S}_{(N+M+1)}(k)\]

(3)

where \( a^{(k)} = (a_0^{(k)}, \ldots, a_{k-1}^{(k)})^T \in \mathbb{R}^k \) and \( H_k(n) = [G_0(n), \ldots, G_{k-1}(n)] \) is the \((N+M+1)\)-mode tensor defined from the given auxiliary tensor sequences \((G_i(n))_n \in \mathbb{R}^{I_1 \times \ldots \times I_N \times K_1 \times \ldots \times K_M; i = 0, \ldots, k-1}. \)

We will see later how to choose the vector \( a^{(k)} \) obtained from \((T_k^{(n)})\) as follows:

\[
\tilde{T}_k(S_n) = S_{n+1} + \sum_{j=0}^{k-1} a_j^{(k)} G_j(n + 1)
\]

\[= S_{n+1} + H_k(n + 1) \tilde{S}_{(N+M+1)}(k).
\]

Notice that the scalars \( a_j^{(k)} \) are the same in the expressions of \( T_k(S_n) \) and \( \tilde{T}_k(S_n) \). We define now the generalized residual of \( T_k^{(n)} \) by

\[
\mathcal{R}_k^n = \mathcal{R}(T_k^{(n)}) = \tilde{T}_k(S_n) - T_k(S_n)
\]

\[= (S_{n+1} - S_n) + \sum_{j=0}^{k-1} a_j^{(k)} (G_j(n + 1) - G_j(n))
\]

\[= \Delta S_n + \sum_{j=0}^{k-1} a_j^{(k)} (\Delta G_j(n)).
\]
Then we get
\[
\widetilde{R}(T_k^{(n)}) = \Delta S_n + \Delta H_k(n) \tilde{x}_{(N+M+1)\alpha^{(k)}},
\]
(6)
where the first forward differences \(\Delta S_n = S_{n+1} - S_n\) and \(\Delta H_k(n) = [\Delta G_0(n), \ldots, \Delta G_{k-1}(n)] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M \times k}\). The vector \(\alpha^{(k)}\) is obtained from the orthogonality relation:
\[
\widetilde{R}(T_k^{(n)}) \in (\text{span}\{\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)}\})^\perp
\]
(7)
where \(\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M}\) are given tensors.

Here, the \(\text{span}\{\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)}\}\) is the tensor subspace generated by the tensors \(\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)}\).

If \(\mathbf{H}_{k,n}\) and \(\mathbf{L}_{k,n}\) denote the tensor subspaces \(\mathbf{H}_{k,n} = \text{span}\{\Delta G_0(n), \ldots, \Delta G_{k-1}(n)\}\) and \(\mathbf{L}_{k,n} = \text{span}\{\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)}\}\), then from (6) and (7), the generalized residual satisfies the following relations:
\[
\widetilde{R}(T_k^{(n)}) - \Delta S_n \in \mathbf{H}_{k,n}
\]
(8)
and
\[
\widetilde{R}(T_k^{(n)}) \in (\mathbf{L}_{k,n})^\perp.
\]
(9)

Let \(\mathbf{L}_{k,n} = [\gamma_0^{(n)}, \ldots, \gamma_{k-1}^{(n)}] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M \times k}\), then the relations (29) and (9) could be expressed as follows:
\[
\widetilde{R}(T_k^{(n)}) - \Delta S_n = \Delta H_k(n) \tilde{x}_{(N+M+1)\alpha^{(k)}}
\]
(10)
and
\[
\mathbf{L}_{k,n} \boxplus (N+M+1) \widetilde{R}(T_k^{(n)}) = 0.
\]
(11)

Assuming that \((\mathbf{L}_{k,n} \boxplus (N+M+1) \Delta H_k(n))\) is nonsingular, the vector \(\alpha^{(k)}\) appearing in the expression (6) of the generalized residual \(\widetilde{R}(T_k^{(n)})\) is given by
\[
\alpha^{(k)} = -(\mathbf{L}_{k,n} \boxplus (N+M+1) \Delta H_k(n))^{-1}(\mathbf{L}_{k,n} \boxplus (N+M+1) \Delta S_n).
\]
(12)

Therefore, the approximation \(T_k^{(n)}\) is computed as follows
\[
T_k^{(n)} = S_n + H_k(n) \tilde{x}_{(N+M+1)\alpha^{(k)}}
\]
(13)

For the tensor global polynomial extrapolation methods, namely Tensor Global MPE (TG-MPE), Tensor Global RRE (TG-RRE) and Tensor Global MMPE (TG-MMPE), the auxiliary sequences are given as
\[
\mathcal{G}_\ell^{(n)} = \Delta S_{(n+\ell)}, \ \ell = 0, \ldots, k - 1; \ n \geq 0
\]

Let \(\Delta^\ell V_k(n)\) be the tensor defined by
\[
\Delta^\ell V_k(n) = [\Delta^\ell S_n, \ldots, \Delta^\ell S_{n+k-1}] \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times K_1 \times K_2 \times \ldots \times K_M \times k}, \ \ i = 1, 2,
\]
where \(\Delta^2\) is the second forward difference \(\Delta^2 S_n = \Delta S_{n+1} - \Delta S_n\). In this case, using the relation (12) and the fact that \(H_k(n) = \Delta V_k(n)\), and \(\Delta H_k(n) = \Delta^2 V_k(n)\), the approximations \(T_k^{(n)}\) given in (13) can be expressed as:
\[
T_k^{(n)} = S_n - \Delta V_k(n) \tilde{x}_{(N+M+1)((\mathbf{L}_{k,n} \boxplus (N+M+1) \Delta^2 V_k(n))^{-1}(\mathbf{L}_{k,n} \boxplus (N+M+1) \Delta S_n))}.
\]
(14)
It is clear that $T_k^{(n)}$ exists and is unique if and only if the square matrix $L_{k,n} \otimes (N+M+1)$ $\Delta^2 V_k(n)$ is nonsingular. The generalized residual given in the relation (6) can be expressed as follows:

$$
\tilde{R}(T_k^{(n)}) = \Delta S_n - \Delta^2 V_k(n) \tilde{\otimes} (N+M+1) \Delta^2 V_k(n)^{-1} (L_{k,n} \otimes (N+M+1) \Delta S_n)).
$$

The choice of the tensors $Y_0^{(n)}, \ldots, Y_{k-1}^{(n)} \in \mathbb{R}^{l_1 \times l_2 \times \ldots \times l_n \times k_1 \times k_2 \times \ldots \times k_M}$ required in the orthogonality relation (7) determines the global-polynomial tensor extrapolation method. For the three well known extrapolation polynomial-based methods, we have the following choices

$$
Y_0^{(n)} = \Delta S_{n+\ell} \quad \text{for TG-MPE},
$$

$$
Y_\ell^{(n)} = \Delta^2 S_{n+\ell} \quad \text{for TG-RRE},
$$

$$
Y_{\ell+1}^{(n)} = Y_{n+1} \quad \text{for TG-MMPE}.
$$

Next, we will propose an efficient implementation of these methods. For this purpose, we first express the approximation $T_k^{(n)}$ given in relation (2) in a different way.

Using the relation (2), the fact that $Y_\ell^{(n)} = \Delta S_{n+\ell}$, $\ell = 0, \ldots, k-1$; $n \geq 0$, and $\tilde{R}(T_k^{(n)}) \in (\span \{Y_0^{(n)}, \ldots, Y_{k-1}^{(n)}\})^\perp$, the (TG-RRE), (TG-MPE) and (TG-MMPE) extrapolation methods produce approximations $T_k^{(n)}$ of the form

$$
T_k^{(n)} = \sum_{j=0}^{k} Y_j^{(n)} S_{n+j},
$$

where

$$
\sum_{j=0}^{k} \gamma_j^{(n)} = 1 \quad \text{and} \quad \sum_{j=0}^{k} \eta_{\ell,j}^{(n)} = 0 \quad 0 \leq \ell < k,
$$

with $\eta_{\ell,j} = \langle Y_\ell^{(n)}, \Delta S_{n+j} \rangle$.

The system of linear Equation (17) can be written as

$$
\begin{align*}
\gamma_0^{(k)} + \gamma_1^{(k)} + \ldots + \gamma_k^{(k)} &= 1 \\
\gamma_0^{(k)} \langle Y_0^{(n)}, \Delta S_n \rangle + \gamma_1^{(k)} \langle Y_0^{(n)}, \Delta S_{n+1} \rangle + \ldots + \gamma_k^{(k)} \langle Y_0^{(n)}, \Delta S_{n+k} \rangle &= 0 \\
\gamma_0^{(k)} \langle Y_1^{(n)}, \Delta S_n \rangle + \gamma_1^{(k)} \langle Y_1^{(n)}, \Delta S_{n+1} \rangle + \ldots + \gamma_k^{(k)} \langle Y_1^{(n)}, \Delta S_{n+k} \rangle &= 0 \\
\vdots \\
\gamma_0^{(k)} \langle Y_{k-1}^{(n)}, \Delta S_n \rangle + \gamma_1^{(k)} \langle Y_{k-1}^{(n)}, \Delta S_{n+1} \rangle + \ldots + \gamma_k^{(k)} \langle Y_{k-1}^{(n)}, \Delta S_{n+k} \rangle &= 0
\end{align*}
$$

Let $\beta_l^{(k)} = \gamma_0^{(k)} / \gamma_l^{(k)}$ for $0 \leq l \leq k$, then

$$
\gamma_l^{(k)} = \frac{\beta_l^{(k)}}{\sum_{l=0}^{k} \beta_l^{(k)}} \quad \text{for} \quad 0 \leq l < k \quad \text{and} \quad \beta_k^{(k)} = 1.
$$

With these notations, the linear system of Equation (18) becomes

$$
\begin{align*}
\beta_0^{(k)} \langle Y_0^{(n)}, \Delta S_n \rangle + \ldots + \beta_{k-1}^{(k-1)} \langle Y_0^{(n)}, \Delta S_{n+k-1} \rangle &= -\langle Y_0^{(n)}, \Delta S_{n+k} \rangle \\
\beta_0^{(k)} \langle Y_1^{(n)}, \Delta S_n \rangle + \ldots + \beta_{k-1}^{(k-1)} \langle Y_1^{(n)}, \Delta S_{n+k-1} \rangle &= -\langle Y_1^{(n)}, \Delta S_{n+k} \rangle \\
\beta_0^{(k)} \langle Y_{k-1}^{(n)}, \Delta S_n \rangle + \ldots + \beta_{k-1}^{(k-1)} \langle Y_{k-1}^{(n)}, \Delta S_{n+k-1} \rangle &= -\langle Y_{k-1}^{(n)}, \Delta S_{n+k} \rangle.
\end{align*}
$$
The above system of equations can also be expressed in the following compact form

\[
(L_{k,n} \square^{(N+M+1)} \Delta V_k(n)) \beta^{(k)} = -(L_{k,n} \square^{(N+M+1)} \Delta S_{n+k})
\]

where \( \beta^{(k)} = [\beta_0^{(k)}, \ldots, \beta_{k-1}^{(k)}]^T \). Assume now that \( \gamma_0^{(k)}, \gamma_1^{(k)}, \ldots, \gamma_k^{(k)} \) have been calculated and introduce the new variables

\[
\delta_0^{(k)} = 1 - \gamma_0^{(k)}, \quad \delta_j^{(k)} = \delta_{j-1}^{(k)} - \gamma_j^{(k)}, \quad 1 \leq j < k \quad \text{and} \quad \delta_{k-1}^{(k)} = \gamma_k^{(k)}.
\]

then the tensor approximation \( T_k^{(n)} \) can be expressed as

\[
T_k^{(n)} = S_n + \sum_{j=0}^{k-1} \delta_j^{(k)} \Delta S_{n+j} = S_n + \Delta V_k(n) \tilde{\times}^{(N+M+1)} \delta^{(k)}
\]

where \( \delta^{(k)} = [\delta_0^{(k)}, \ldots, \delta_{k-1}^{(k)}]^T. \)

3.2. The tensor Global Topological \( \epsilon \)-Transformation

In [6], Brezinski proposed a generalization of the scalar \( \epsilon \)-algorithm for vector sequences called the topological \( \epsilon \)-algorithm (TEA). The matrix case has been introduced by Jbilou and Sadok in ([9]). In this section we define the tensor global topological \( \epsilon \)-transformation (TG-TET).

Let \( (S_n) \) be a sequence of tensors of \( \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times K_1 \times K_2 \times \cdots \times K_M} \) and consider approximations \( E_k(S_n) = E_k^{(n)} \) of the limit of the tensor sequence \( (S_n)_{n \in \mathbb{N}} \) such that

\[
E_k^{(n)} = S_n + \sum_{i=1}^{k} a_i^{(n)} \Delta S_{n+i-1}, \quad n \geq 0.
\]

where \( a_i^{(n)} \in \mathbb{R} \), for \( i = 1, \ldots, k \). We introduce the new tensor transformation \( \tilde{E}_{k,j}^{(n)} \), \( j = 1, \ldots, k \) defined by

\[
\tilde{E}_{k,j}^{(n)} = S_{n+j} + \sum_{i=1}^{k} a_i^{(n)} \Delta S_{n+i+j-1} \quad j = 1, \ldots, k.
\]

We set \( \tilde{E}_{k,0}^{(n)} = E_k^{(n)} \) and define the \( j \)-th tensor generalized residual as follows

\[
\tilde{R}_j(E_k^{(n)}) = \tilde{E}_{k,j}^{(n)} - \tilde{E}_{k,j-1}^{(n)}.
\]

The coefficients \( a_i^{(n)} \) involved in the expression (24) of \( E_k^{(n)} \) are computed such that each \( j \)-th generalized residual is orthogonal to some chosen tensor \( Y \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times K_1 \times K_2 \times \cdots \times K_M} \), that is

\[
\langle Y, \tilde{R}_j(E_k^{(n)}) \rangle = 0; \quad j = 1, \ldots, k.
\]

Let \( D_{k,n} \) denote the following matrix

\[
D_{k,n} = \begin{pmatrix}
\langle Y, \Delta^2 S_n \rangle & \cdots & \langle Y, \Delta^2 S_{n+k-1} \rangle \\
\langle Y, \Delta^2 S_{n+1} \rangle & \cdots & \langle Y, \Delta^2 S_{n+k} \rangle \\
\vdots & \ddots & \vdots \\
\langle Y, \Delta^2 S_{n+k-1} \rangle & \cdots & \langle Y, \Delta^2 S_{n+2k-2} \rangle
\end{pmatrix}.
\]
Theorem 1. When applied to the sequence generated by (32), TG-RRE, and TG-MPE are tensor Krylov subspace methods and are mathematically equivalent to the tensor global GMRES and the tensor global Arnoldi methods, respectively.

Proof. Notice first that for tensor linear sequences (32) the generalized residual becomes the true residual. In fact, from (32), we have \( \Delta S_n = B - A * N S_n = R(S_n) \) the residual of the tensor \( S_n \). Since \( \Delta^2 S_n = -A * N \Delta S_n \) we have \( \Delta^2 \nu_k(n) = [\Delta^2 S_n, \ldots, \Delta^2 S_{n+k-1}] = -A * N \Delta \nu_k(n) \) where \( \Delta \nu_k(n) = [\Delta S_n, \ldots, \Delta S_{n+k-1}] \). Consequently using (6) and (32), the generalized residual of the approximation \( (T_k(n)) \) is the true residual \[
\tilde{R}(T_k(n)) = R(T_k(n)) = B - A * N T_k(n).
\]
For simplicity and unless specified otherwise, we set \( n = 0, T_k^{(0)} = T_k, S_0 = X_0 \) the initial guess and drop the index \( n \) in all our notations. When applied to the sequence generated by the linear relation (32), the TG-RRE, TG-MMPE and the TG-MPE above produce approximations \( X_k = T_k \) such that the corresponding residual \( R_k = B - A * N T_k \) satisfies the relations

\[
R_k - \Delta S_0 \in \tilde{H}_k = -A * N \tilde{V}_k
\]

\[
R_k \in (\tilde{L}_k)^\perp
\]

where \( \tilde{V}_k = \text{span} \{\Delta S_0, \ldots, \Delta S_{k-1}\} \) and \( \tilde{L}_k \equiv \tilde{H}_k \) for TG-RRE, \( \tilde{L}_k \equiv \tilde{V}_k \) for TG-MPE and \( \tilde{L}_k \equiv \tilde{Y}_k = \text{span} \{J_0, \ldots, J_{k-1}\} \) for TG-MMPE where \( J_0, \ldots, J_{k-1} \) are some chosen tensors.

Notice that, since \( \tilde{V}_k = \mathbb{K}_m(A, R_0) \) (the tensor Krylov subspace defined in [18,19]), the extrapolation methods above are tensor global Krylov subspace methods. TG-RRE is an orthogonal projection and is theoretically equivalent to the tensor global GMRES while TG-MPE is oblique projection method and is equivalent to the tensor global Arnoldi method.

As the TG-RRE is an orthogonal projection method, we also have the classical minimization property for the residual

\[
||R_{TG-RRE}|| = \min_{T_k \in X_0 + \mathbb{K}_m(A, R_0)} ||B - A * N T_k||.
\]

When the linear process (32) is convergent, it is more useful in practice to apply the tensor extrapolation methods after some fixed number of iterations. To save memory, we can also use the algorithm in a cycling mode which means that the iterations are restarted after a fixed number \( m \) of iterations. Algorithm 1 is summarized as follows:

Algorithm 1 TG-RRE, TG-MPE and TG-MMPE Algorithms

Step 1. \( k = 0 \) choose \( X_0 \) and the numbers \( p \) and \( m \).

Step 2. Basic iteration

Set \( T_0 = X_0 \)

\[
Z_0 = T_0,
\]

\[
Z_{j+1} = (I - A) * N Z_j + B, \quad j = 0, \ldots, p - 1
\]

Step 3. Extrapolation schema

\[
S_0 = Z_p
\]

\[
S_{n+1} = (I - A) * N S_n + B, \quad n = 0, \ldots, m
\]

Compute the approximation \( T_m^{(0)} \) by TG-RRE, TG-MPE or TG-MMPE.

Set \( X_0 = T_m^{(0)} \), \( k = k + 1 \) and go to step 2.

4.2. Application to Non Linear Tensor Systems

Consider the nonlinear system of tensor equations

\[
G(X) = B
\]

(33)

with \( G(X) \) an operator from \( \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_M} \) onto \( \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_M} \), and \( X^* \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N \times I_1 \times I_2 \times \ldots \times I_M} \) the solution of the system of Equation (33). For any arbitrary tensor \( X \), the residual is given by

\[
R(X) = G(X) - X
\]

(34)

Let \( (S_n)_n \) be the sequence of tensors generated from an initial guess \( S_0 \) as follows

\[
S_{n+1} = G(S_n), \quad j = 0, 1, \ldots
\]
To get approximate solutions to a solution of (33), we apply the tensor extrapolation methods above to the sequence \((S_n)\). As for the linear case, the different steps are summarized in the following Algorithm 2.

**Algorithm 2** TG-RRE, TG-MPE and TG-MMPE for nonlinear systems

Step 1. \(k = 0\) choose \(X_0\) and the numbers \(p\) and \(m\).
Step 2. Basic iteration
   
   Set \(T_0 = X_0\)
   
   \(W_0 = T_0\)
   
   \(W_{j+1} = G(W_j)\quad j = 0, \ldots, p - 1\)

Step 3. Extrapolation scheme
   
   \(S_0 = W_p\)
   
   If \(||S_1 - S_0||_F < \epsilon\) stop
   
   otherwise generate \(S_{n+1} = G(S_n)\), \(n = 0, \ldots, m\)

   Compute the approximation \(T_m^{(0)}\) by TG-RRE, TG-MPE or TG-MMPE.

Set \(X_0 = T_m^{(0)}\), \(k = k + 1\) and go to step 2.

**Remark 2.** As we stated earlier, when applied to linearly generated tensor sequences, the proposed tensor extrapolation methods produce the same iterates as some well known tensor Krylov subspace methods but differ in the way that the approximations are computed. In the case where the process of generating the sequence \((S_n)\) is not known and what is known is only the terms of this sequences, Krylov-based methods or Newton-type methods could not be used and in that case, extrapolation methods are well come. Such a problem can be found for example in some statistical-problems; see for example [16] for the application of the vector ε-algorithm of Wynn [23] in the expectation–maximization (EM) algorithm to find maximum likelihood estimates from incomplete or missing data.

5. The Global-QR Implementation of TG-MPE/TG-RRE

The purpose of this section is to give an efficient implementation of TG-MPE and TG-RRE using a generalisation of the technique based on the QR decomposition and given in [11] for vector MPE and vector RRE methods. We first introduce the Tensor Global-QR decomposition. Let \(U \in \mathbb{R}^{L_1 \times L_2 \times \ldots \times L_M \times H_1 \times H_2 \times \ldots H_N \times K}\), be an \((M + N + 1)\)-mode tensor with the column tensors \(U_0, \ldots, U_{k-1} \in \mathbb{R}^{H_1 \times H_2 \times \ldots H_M \times H_{k} \times \ldots H_N}\). Then, there is an \((M + N + 1)\)-mode orthogonal tensor \(Q = [Q_0, \ldots, Q_{k-1}] \in \mathbb{R}^{H_1 \times H_2 \times \ldots \times H_M \times H_k \times \ldots H_N}\) satisfying \(Q \otimes (N+M+1)\) \(Q = I_{k \times k}\) and an upper triangular matrix \(R \in \mathbb{R}^{k \times k}\) such that

\[
U = Q \times (M+N+1) R^T
\]

(35)

The tensor decomposition (35) will be called the Tensor Global-QR (TG-QR) decomposition of \(A\) and is summarized in the following Algorithm 3.

**Algorithm 3** The global-QR decomposition

1. Given \(U_0\), and compute the scalar \(r_{00}\) and the tensor \(Q_0\) by \(r_{00} = \langle U_0, U_0 \rangle^\frac{1}{2}\) and \(Q_0 = \frac{U_0}{r_{00}}\).
2. For \(j = 1, \ldots, k\)
   
   (a) \(W = U_j\)
   
   (b) For \(i = 0, \ldots, j - 1\) do
      
      \(r_{ij} = \langle Q_i, W \rangle; W = W - r_{ij}Q_i\)
   
   End for
   
   (c) \(r_{jj} = \langle W, W \rangle^\frac{1}{2}\)
   
   (d) \(Q_j = \frac{W}{r_{jj}}\)
   
   (e) End for
3. End
To show that the tensor $\mathcal{Q}$ is orthogonal, we just proceed by induction on $k$. The coefficients of the matrix $R$ are the $r_{ij}$’s given by Algorithm 3. Next, we give a proposition to be used.

Proposition 2 ([24]). Let $X \in \mathbb{R}^{l_1 \times l_2 \times \ldots \times l_n}$, $A \in \mathbb{R}^{l_1 \times l_2}$ and $y \in \mathbb{R}^{l_2}$, then we have

$$X \times_{(n)} A \bar{X}_{(n)} y = X \bar{X}_{(n)} (A^T y).$$

For simplicity, we set here $n = 0$ and then the approximation $\mathcal{I}^{(0)}_k$ (defined earlier in (23)) is given by

$$\mathcal{I}^{(0)}_k = S_0 + \sum_{j=0}^{k-1} \delta_j \Delta S_j = S_0 + \Delta V_k(0) \bar{\Delta}(N+M+1) \delta_k$$

Substituting now $V_k = V_k(0) = Q \times_{(N+M+1)} R^T$ and using Proposition 2, we get

$$\mathcal{I}^{(0)}_k = S_0 + \Delta V_k \bar{\Delta}(N+M+1) \delta_k$$

which gives

$$\mathcal{I}^{(0)}_k = S_0 + \sum_{j=0}^{k-1} \theta_j \mathcal{Q}_j,$$  \hspace{1cm} (36)

where $\theta_j$ is the $(j+1)$ component of the column vector $R \delta_k$; the matrix $R$ and the tensors $\mathcal{Q}_j \in \mathbb{R}^{l_1 \times l_2 \times \ldots \times l_n}, j = 0, \ldots, k-1$ are given by Algorithm 3. To compute $\delta_k$, we have first to compute $\beta^{(k)} = (\beta_0^{(k)}, \ldots, \beta^{(k)}_{N-1})^T$ and use the relations given in (22).

For TG-MPE, the coefficients $\gamma^{(k)}$’s of the vector $\gamma^{(k)}$ are determined by computing the vector $\beta^{(k)}$, the solution of the linear system of equations (20)

$$(\Delta V_k \bar{\Delta}^{(N+M+1)} \Delta V_k) \beta^{(k)} = -(\Delta V_k \bar{\Delta}^{(N+M+1)} \Delta S_k)$$  \hspace{1cm} (37)

where

$$(\Delta V_k \bar{\Delta}^{(N+M+1)} \Delta V_k) = \begin{pmatrix}
\langle \Delta S_0, \Delta S_0 \rangle & \ldots & \langle \Delta S_0, \Delta S_{k-1} \rangle \\
\langle \Delta S_1, \Delta S_0 \rangle & \ldots & \langle \Delta S_1, \Delta S_{k-1} \rangle \\
\vdots & \ddots & \vdots \\
\langle \Delta S_{k-1}, \Delta S_0 \rangle & \ldots & \langle \Delta S_{k-1}, \Delta S_{k-1} \rangle
\end{pmatrix} \in \mathbb{R}^{k \times k}$$  \hspace{1cm} (38)

and $(\Delta V_k \bar{\Delta}^{(N+M+1)} \Delta S_k) = (\langle \Delta S_0, \Delta S_k \rangle, \ldots, \langle \Delta S_{k-1}, \Delta S_k \rangle)^T \in \mathbb{R}^k$.

For TG-RRE, $\beta^{(k)}$ are determined by solving the linear system of equation

$$(\Delta^2 V_k \bar{\Delta}^{(N+M+1)} \Delta V_k) \beta^{(k)} = -(\Delta^2 V_k \bar{\Delta}^{(N+M+1)} \Delta S_k)$$

where

$$(\Delta^2 V_k \bar{\Delta}^{(N+M+1)} \Delta V_k) = \begin{pmatrix}
\langle \Delta^2 S_0, \Delta S_0 \rangle & \ldots & \langle \Delta^2 S_0, \Delta S_{k-1} \rangle \\
\langle \Delta^2 S_1, \Delta S_0 \rangle & \ldots & \langle \Delta^2 S_1, \Delta S_{k-1} \rangle \\
\vdots & \ddots & \vdots \\
\langle \Delta^2 S_{k-1}, \Delta S_0 \rangle & \ldots & \langle \Delta^2 S_{k-1}, \Delta S_{k-1} \rangle
\end{pmatrix} \in \mathbb{R}^{k \times k},$$  \hspace{1cm} (39)

and $(\Delta^2 V_k \bar{\Delta}^{(N+M+1)} \Delta S_k) = (\langle \Delta^2 S_0, \Delta S_k \rangle, \ldots, \langle \Delta^2 S_{k-1}, \Delta S_k \rangle)^T \in \mathbb{R}^k$. 
Finally, once $\beta^{(k)}$ is computed, the coefficients $\gamma^{(k)}_l$'s are given by

$$\gamma^{(k)}_l = \frac{\beta^{(k)}_l}{\sum_{l=0}^{k} \beta^{(k)}_l} \quad \text{for} \quad 0 \leq l < k \quad \text{and} \quad \beta^{(k)}_k = 1. \quad (40)$$

The following Algorithm 4 summarizes the main steps.

**Algorithm 4** Implementation of TG-MPE/TG-RRE via the global-QR decomposition

1. $S_0$ is a given initial guess and $k$ is a fixed index.
2. Apply Algorithm 3 to compute the global-QR decomposition of the tensor $\Delta V_k$.
3. Compute $\beta^{(k)}$ by solving the linear system (38) or (39).
4. Compute the coefficients $\gamma^{(k)}_j$ from (40).
5. Compute $\delta^{(k)}$ from the relation (22) and $\theta_j$ is $(j + 1)$ component of the column vector $R\delta^{(k)}$.
6. Compute $T_k^{(0)}$ by $T_k^{(0)} = S_0 + \sum_{j=0}^{k-1} \theta_j Q_j$.

As a numerical test, we consider the linear system of tensor equations given by (31) where $A$ is a random tensor $N = 3$, $I_1 = I_2 = 20, I_3 = 10, J_1 = 20, J_2 = 10$ and $J_3 = 5$ as in [19]. The exact solution is the tensor $X$ whose elements are all one and the right hand side $B$ is given by $B = A^N X$. The computations are carried out using MATLAB 7.4 with machine epsilon about $2 \times 10^{-16}$.

We took $m = 10$, $p = 1$ and stopped the iteration when the relative error norm of the residual was less than $10^{-6}$. Then after $k = 6$ iterations, Algorithm 4 for the tensor RRE method, gives the relative residual norm $R_k / R_0 = 8.5 \times 10^{-7}$ where the initial guess is $X_0 = O$. We notice that the convergence of the original sequence $(S_n)$ is very slow and at $n = 200$, we get a relative residual $R(S_n) / R_0$ of $10^{-1}$. Generally, extrapolation methods are more effective than Krylov-based methods for nonlinear problems. More experimental studies and applications in some areas such as multilinear google pagerank should be considered in the future.

6. Conclusions

In this paper, we introduced new tensor global extrapolation methods to accelerate the convergence of some tensor sequences. The proposed methods are generalisations to the tensor case of some well known vector extrapolation methods such as the reduced rank extrapolation or the topological epsilon algorithm. The new methods were defined as orthogonal or oblique projection processes using the Einstein product and also some new interesting tensor products. We showed how to apply the derived algorithms to tensor linear and nonlinear systems of tensor equations. Application to some interesting problems such as the multilinear page rank is still under investigation.

**Author Contributions:** Formal analysis, A.E.I.; Methodology, R.S.; Writing—original draft, K.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** For the first author, this project was financially supported by Ministry of Europe and Foreign Affairs, Ministry of Higher Education, Research and Innovation and the French Institute of Rabat (PHC TOUBKAL 20XX (French-Morocco bilateral program) Grant Number: 12345AB).

**Acknowledgments:** We would like to thank the two referees for valuable remarks and helpful comments.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Brezinski, C.; Redivo Zaglia, M. *Extrapolation Methods. Theory and Practice*; North Holland Publishing: Amsterdam, The Netherlands, 1991.
2. Aitken, A.C. On Bernoulli’s numerical solution of algebraic equations. *Proc. R. Soc. Edinb.* 1926, 46, 289–305. [CrossRef]
3. Cabay, S.; Jackson, L.W. A polynomial extrapolation method for finding limits and antilimits for vector sequences. *SIAM J. Numer. Anal.* 1976, 13, 734–752. [CrossRef]
4. Mesina, M. Convergence acceleration for the iterative solution of $Ax + f$. *Comput. Meth. Appl. Mech. Eng.* 1977, 10, 165–173. [CrossRef]
5. Eddy, V.P. Extrapolation to the limit of a vector sequence. In *Information Linkage between Applied Mathematics and Industry*; Wang, P.C.C., Ed.; Academic Press: New York, NY, USA, 1979; pp. 387–396.
6. Brezinski, C. Généralisation de la transformation de Shanks, de la table de Padé et l’epsilon algorithm. *Calcolo* 1975, 12, 317–360. [CrossRef]
7. Pugatchev, B.P. Acceleration of the convergence of iterative processes and a method for solving systems of nonlinear equations. *USSR Comput. Math. Math. Phys.* 1978, 17, 199–207. [CrossRef]
8. Jbilou, K.; Sadok, H. Analysis of some vector extrapolation methods for linear systems. *Numer. Math.* 1995, 70, 73–89. [CrossRef]
9. Jbilou, K.; Sadok, H. VMatrix polynomial and epsilon-type extrapolation methods with applications. *Numer. Algorithms* 2015, 68, 107–119. [CrossRef]
10. Jbilou, K.; Messaoudi, A.; Tabaa, K. Some Schur complement identities to matrix extrapolation methods. *Linear Algebra Appl.* 2004, 392, 195–210. [CrossRef]
11. Sidi, A. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *J. Comput. Appl. Math.* 1991, 71, 1990. [CrossRef]
12. Sidi, A.; Ford, W.E.; Smith, D.A. Acceleration of convergence of vector sequences. *SIAM J. Numer. Anal.* 1986, 23, 178–196. [CrossRef]
13. Jbilou, K.; Sadok, H. Vector extrapolation methods. Application and numerical comparison. *J. Comput. Appl. Math.* 2000, 122, 149–165. [CrossRef]
14. Brezinski, C.; Redivo Zaglia, M.; Serra-Capizzano, S. Extrapolation methods for PageRank computation. *Comptes Rendus Math.* 2005, 340, 393–397. [CrossRef]
15. Kamvar, D.; Haveliwala, T.H.; Manning, C.D.; Golub, G.H. Extrapolations methods for accelerating PageRank computations. In Proceedings of the WWW 2003—Twelfth International World Wide Web Conference, Budapest, Hungary, 20–24 May 2003.
16. Kuroda, M.; Sakakihara, M. Accelerating the convergence of the EM algorithm using the vector epsilon-algorithm. *Comput. Stat. Data Anal.* 2006, 51, 1549–1561. [CrossRef]
17. Duminil, S.; Sadok, H.; Silvester, D. Fast solvers for discretized Navier–Stokes problems using vector extrapolation. *Numer. Algorithms* 2014, 66, 89–104. [CrossRef]
18. Elguide, M.; El Ichi, A.; Jbilou, K.; Beik, F.P.A. Tensor GMRES and Golub-Kahan Bidiagonalization methods via the Einstein product with applications to image and video processing. *arXiv* 2020 arXiv:2005.07458.
19. Huang, B.; Xie, Y.; Ma, C. Krylov subspace methods to solve a class of tensor equations via the Einstein product. *Numer. Linear Algebra Appl.* 2019, 26, e2254. [CrossRef]
20. Kolda, T.G.; Bader, B.W. Tensor Decompositions and Applications. *SIAM Rev.* 2009, 3, 455–500. [CrossRef]
21. Liang, M.; Zheng, B. Further results on Moore–Penrose inverses of tensors with application to tensor nearness problems. *Comput. Math. Appl.* 2019, 77, 1282–1293. [CrossRef]
22. Bouyouli, R.; Jbilou, K.; Sadaka, R.; Sadok, H. Convergence properties of some block Krylov subspace methods for multiple linear systems. *J. Comput. Appl. Math.* 2006, 196, 498–511. [CrossRef]
23. Wynn, P. Acceleration techniques for iterated vector and matrix problems. *Math. Comput.* 1962, 16, 301–322. [CrossRef]
24. Beik, F.P.A.; Movahed, F.S.; Ahmadi-Asl, S. On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations. *Numer. Linear Algebra Appl.* 2016, 16, 444–466. [CrossRef]