Shanks’ transformations, Anderson acceleration, and applications to systems of equations

C. Brezinski 1  M. Redivo–Zaglia 2  Y. Saad 3

1Université de Lille, France
2Università di Padova, Italy
3University of Minnesota, USA

2ggALN - Rome - February 18-19, 2019
Shanks’ transformations in a vector space
Coupled topological Shanks’ transformations (CTST)
Particular methods
Properties
Linear systems connections
The quasi-Newton connection
Anderson and Anderson-type Acceleration methods
In computational sciences it is often necessary to obtain the limit of a sequence of objects of a vector space (scalars, vectors, matrices, ...) that converges slowly to its limit or even diverges.
In computational sciences it is often necessary to obtain the limit of a sequence of objects of a vector space (scalars, vectors, matrices, ...) that converges slowly to its limit or even diverges. In some situations, we may be able to obtain a new sequence that converges faster to the same limit by modifying the method that produced the original sequence.
In computational sciences it is often necessary to obtain the limit of a sequence of objects of a vector space (scalars, vectors, matrices, ...) that converges slowly to its limit or even diverges. In some situations, we may be able to obtain a new sequence that converges faster to the same limit by modifying the method that produced the original sequence. However, in many instances, the process by which the sequence is produced is hidden into a black box or too cumbersome for this approach to be practical.
In computational sciences it is often necessary to obtain the limit of a sequence of objects of a vector space (scalars, vectors, matrices, ...) that converges slowly to its limit or even diverges. In some situations, we may be able to obtain a new sequence that converges faster to the same limit by modifying the method that produced the original sequence. However, in many instances, the process by which the sequence is produced is hidden into a black box or too cumbersome for this approach to be practical. Thus, a common solution is to transform this sequence, by means of a sequence transformation, into a new sequence which, under some assumptions, converges faster.
In computational sciences it is often necessary to obtain the limit of a sequence of objects of a vector space (scalars, vectors, matrices, ...) that converges slowly to its limit or even diverges. In some situations, we may be able to obtain a new sequence that converges faster to the same limit by modifying the method that produced the original sequence. However, in many instances, the process by which the sequence is produced is hidden into a black box or too cumbersome for this approach to be practical. Thus, a common solution is to transform this sequence, by means of a sequence transformation, into a new sequence which, under some assumptions, converges faster. Among these general techniques Shanks’ transformation is arguably the best all-purpose method for accelerating convergence of sequences.
The aim of this talk is to present a general framework for Shanks’ transformation(s) of sequences of elements in a vector space.
The aim of this talk is to present a general framework for Shanks’ transformation(s) of sequences of elements in a vector space.

This framework includes

- the **Minimal Polynomial Extrapolation (MPE)**,
- the **Reduced Rank Extrapolation (RRE)**,
- the **Modified Minimal Polynomial Extrapolation (MMPE)**,
- the **Topological Shanks transformation (TEA and STEA algorithm)**,

and in some sense also

- **Anderson Acceleration (AA)**.

Their application to the solution of systems of linear and nonlinear equations will be discussed.
Let \((s_n)\) be a sequence of elements of a vector space \(E\) on \(\mathbb{R}\) or \(\mathbb{C}\), which converges to \(s \in E\), and satisfying, for a fixed value of \(k\) and for all \(n\), the difference equation

\[
\alpha_0(s_n - s) + \cdots + \alpha_k(s_{n+k} - s) = 0,
\]

with \(\alpha_j \in \mathbb{R}\), \(\alpha_0 \alpha_k \neq 0\), and \(\alpha_0 + \cdots + \alpha_k = 1\), a normalization condition which does not restrict generality.
Shanks’ transformations in a vector space

Let \((s_n)\) be a sequence of elements of a vector space \(E\) on \(\mathbb{R}\) or \(\mathbb{C}\), which converges to \(s \in E\), and satisfying, for a fixed value of \(k\) and for all \(n\), the difference equation

\[
\alpha_0(s_n - s) + \cdots + \alpha_k(s_{n+k} - s) = 0,
\]

with \(\alpha_i \in \mathbb{R}\), \(\alpha_0 \alpha_k \neq 0\), and \(\alpha_0 + \cdots + \alpha_k = 1\), a normalization condition which does not restrict generality.

The set of such sequences is called the Shanks’ kernel.
Shanks’ transformations in a vector space

Let \((s_n)\) be a sequence of elements of a vector space \(E\) on \(\mathbb{R}\) or \(\mathbb{C}\), which converges to \(s \in E\), and satisfying, for a fixed value of \(k\) and for all \(n\), the difference equation

\[
\alpha_0(s_n - s) + \cdots + \alpha_k(s_{n+k} - s) = 0,
\]

with \(\alpha_i \in \mathbb{R}\), \(\alpha_0 \alpha_k \neq 0\), and \(\alpha_0 + \cdots + \alpha_k = 1\), a normalization condition which does not restrict generality.

The set of such sequences is called the Shanks’ kernel.

For a fixed value of \(k\), we want to transform \((s_n)\) into a new sequence \((t^{(k)}_n)\) such that, for sequences belonging to the Shanks’ kernel,

\[
t^{(k)}_n = s, \quad \forall n
\]
If the coefficients $\alpha_i$ are known it immediately follows, from the difference equation and the normalization condition, that this Shanks’ sequence transformation is given by

$$t_n^{(k)} = \alpha_0 s_n + \cdots + \alpha_k s_{n+k}.$$
Shanks’ transformations in a vector space

If the coefficients $\alpha_i$ are known it immediately follows, from the difference equation and the normalization condition, that this Shanks’ sequence transformation is given by

$$t_n^{(k)} = \alpha_0 s_n + \cdots + \alpha_k s_{n+k}.$$  

To determine the $k + 1$ coefficients $\alpha_i$ we need to set-up a linear system of $k$ (scalar) equations, in addition to the normalization condition.
Shanks’ transformations in a vector space

If the coefficients $\alpha_i$ are known it immediately follows, from the difference equation and the normalization condition, that this Shanks’ sequence transformation is given by

$$t^{(k)}_n = \alpha_0 s_n + \cdots + \alpha_k s_{n+k}.$$ 

To determine the $k+1$ coefficients $\alpha_i$ we need to set-up a linear system of $k$ (scalar) equations, in addition to the normalization condition.

If the sequence to be transformed does not belong to the Shanks’ kernel, the coefficients $\alpha_i$ can still be computed by the same system but they now depend on $k$ and $n$, and the transformed sequence is still given as above.
If the coefficients $\alpha_i$ are known it immediately follows, from the difference equation and the normalization condition, that this Shanks’ sequence transformation is given by

$$t_n^{(k)} = \alpha_0 s_n + \cdots + \alpha_k s_{n+k}.$$ 

To determine the $k+1$ coefficients $\alpha_i$ we need to set-up a linear system of $k$ (scalar) equations, in addition to the normalization condition.

If the sequence to be transformed does not belong to the Shanks’ kernel, the coefficients $\alpha_i$ can still be computed by the same system but they now depend on $k$ and $n$, and the transformed sequence is still given as above.

We will now present a general framework including all sequence transformations whose kernel is the set of sequences satisfying the difference equation defining the Shanks’ kernel.
For building these schemes we need the notions of \textbf{extended Schur complement} and \textbf{Schur determinantal formula}. 

Let $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$. We assume that $A \in E$ and that $B$ is a row formed by $q$ elements of $E$. $C$ is a vector of dimension $q$, and $D$ a square and invertible $q \times q$ matrix. The extended Schur complement of $D$ in $M$ is defined as $(M/D) = A - BD^{-1}C \in E$. 

Let $\det(M)$ be the element of $E$ obtained by expanding $M$ with respect to its first row by the classical rules. The extended Schur determinantal formula is $\det(M) = \det(D) \cdot (M/D)$. 
For building these schemes we need the notions of **extended Schur complement** and **Schur determinantal formula**. Let

\[ M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}. \]

We assume that \( A \in E \) and that \( B \) is a row formed by \( q \) elements of \( E \). \( C \) is a vector of dimension \( q \), and \( D \) a square and invertible \( q \times q \) matrix.
For building these schemes we need the notions of **extended Schur complement** and **Schur determinantal formula**.

Let

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$ 

We assume that $A \in E$ and that $B$ is a row formed by $q$ elements of $E$. $C$ is a vector of dimension $q$, and $D$ a square and invertible $q \times q$ matrix.

The **extended Schur complement** of $D$ in $M$ is defined as

$$(M/D) = A - BD^{-1}C \in E.$$
For building these schemes we need the notions of extended Schur complement and Schur determinantal formula. Let

\[ M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}. \]

We assume that \( A \in E \) and that \( B \) is a row formed by \( q \) elements of \( E \). \( C \) is a vector of dimension \( q \), and \( D \) a square and invertible \( q \times q \) matrix.

The extended Schur complement of \( D \) in \( M \) is defined as

\[ (M/D) = A - BD^{-1}C \in E. \]

Let \( \det(M) \) be the element of \( E \) obtained by expanding \( M \) with respect to its first row by the classical rules.

The extended Schur determinantal formula is

\[ \det(M) = \det(D) \cdot (M/D). \]
We assume that we know a sequence \((t_n)\) of elements of \(E\), called the coupled sequence, that satisfies

\[ \alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0, \]

for all \(n\), where the coefficients \(\alpha_i\) are the same as in the difference equation for Shanks’ kernel.
Coupled topological Shanks’ transformations

We assume that we know a sequence \((t_n)\) of elements of \(E\), called the **coupled sequence**, that satisfies

\[
\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0,
\]

for all \(n\), where the coefficients \(\alpha_i\) **are the same** as in the difference equation for Shanks’ kernel.

The corresponding Shanks’ sequence transformation is called a **Coupled Topological Shanks’ Transformation (CTST)**.
How to compute the coefficients $\alpha_i$?

Let us now see how to compute the coefficients $\alpha_i$.

For that purpose, we have to transform the equation in $E$ satisfied by the coupled sequence into an equation in $\mathbb{R}$. 

Three strategies for writing a linear system that yields the coefficients $\alpha_i$ can be employed, and we will now discuss them.

In what follows, $y$ and $y_i$ are elements of $E^*$. 

Let us now see how to compute the coefficients $\alpha_i$.

For that purpose, we have to transform the equation in $E$ satisfied by the coupled sequence into an equation in $\mathbb{R}$.

Let $u$ be an element of $E^*$, the algebraic dual space of $E$, that is the set of linear functionals on $E$ and let $v \in E$.

We denote by $\langle u, v \rangle \in \mathbb{R}$ the duality product between $u \in E^*$ and $v \in E$. 
Let us now see how to compute the coefficients $\alpha_i$.

For that purpose, we have to transform the equation in $E$ satisfied by the coupled sequence into an equation in $\mathbb{R}$.

Let $u$ be an element of $E^*$, the algebraic dual space of $E$, that is the set of linear functionals on $E$ and let $v \in E$.

We denote by $\langle u, v \rangle \in \mathbb{R}$ the duality product between $u \in E^*$ and $v \in E$.

Three strategies for writing a linear system that yields the coefficients $\alpha_i$ can be employed, and we will now discuss them.
Let us now see how to compute the coefficients $\alpha_i$.

For that purpose, we have to transform the equation in $E$ satisfied by the coupled sequence into an equation in $\mathbb{R}$.

Let $u$ be an element of $E^*$, the algebraic dual space of $E$, that is the set of linear functionals on $E$ and let $v \in E$.

We denote by $\langle u, v \rangle \in \mathbb{R}$ the duality product between $u \in E^*$ and $v \in E$.

Three strategies for writing a linear system that yields the coefficients $\alpha_i$ can be employed, and we will now discuss them.

In what follows, $y$ and $y_i$ are elements of $E^*$. 
1. The polynomial extrapolation strategy (Pol)

Let \( \alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0 \).

The **polynomial extrapolation strategy** is obtained by considering the system of linear equations where \( n \) is fixed and several \( y_i \) (which can depend on \( n \)) are used

\[
\begin{aligned}
\alpha_0 + \cdots + \alpha_k &= 1 \\
\alpha_0 \langle y_i, t_n \rangle + \cdots + \alpha_k \langle y_i, t_{n+k} \rangle &= 0, \quad i = 1, \ldots, k.
\end{aligned}
\]
1. The polynomial extrapolation strategy (Pol)

Let $\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0$.

The **polynomial extrapolation strategy** is obtained by considering the system of linear equations where $n$ is fixed and several $y_i$ (which can depend on $n$) are used

\[
\begin{align*}
\alpha_0 + \cdots + \alpha_k &= 1 \\
\alpha_0 \langle y_i, t_n \rangle + \cdots + \alpha_k \langle y_i, t_{n+k} \rangle &= 0, \quad i = 1, \ldots, k.
\end{align*}
\]

Thus, **we obtain the transformation**

\[
t^{(k)}_n = \begin{bmatrix}
    s_n & \cdots & s_{n+k} \\
    \langle y_1, t_n \rangle & \cdots & \langle y_1, t_{n+k} \rangle \\
    \vdots & \ddots & \vdots \\
    \langle y_k, t_n \rangle & \cdots & \langle y_k, t_{n+k} \rangle
\end{bmatrix}
\begin{bmatrix}
    1 & \cdots & 1 \\
    \langle y_1, t_n \rangle & \cdots & \langle y_1, t_{n+k} \rangle \\
    \vdots & \ddots & \vdots \\
    \langle y_k, t_n \rangle & \cdots & \langle y_k, t_{n+k} \rangle
\end{bmatrix}.
\]

that can be also written as
1. The polynomial extrapolation strategy (Pol)

\[
\begin{align*}
\mathbf{t}_n^{(k)} &= \begin{vmatrix}
\mathbf{s}_n & \Delta \mathbf{s}_n & \cdots & \Delta \mathbf{s}_{n+k-1} \\
\langle \mathbf{y}_1, \mathbf{t}_n \rangle & \langle \mathbf{y}_1, \Delta \mathbf{t}_n \rangle & \cdots & \langle \mathbf{y}_1, \Delta \mathbf{t}_{n+k-1} \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle \mathbf{y}_k, \mathbf{t}_n \rangle & \langle \mathbf{y}_k, \Delta \mathbf{t}_n \rangle & \cdots & \langle \mathbf{y}_k, \Delta \mathbf{t}_{n+k-1} \rangle 
\end{vmatrix} / \begin{vmatrix}
\langle \mathbf{y}_1, \Delta \mathbf{t}_n \rangle & \cdots & \langle \mathbf{y}_1, \Delta \mathbf{t}_{n+k-1} \rangle \\
\vdots & \ddots & \vdots \\
\langle \mathbf{y}_k, \Delta \mathbf{t}_n \rangle & \cdots & \langle \mathbf{y}_k, \Delta \mathbf{t}_{n+k-1} \rangle 
\end{vmatrix}
\end{align*}
\]

Thus, according to the extended Schur determinantal formula, \( \mathbf{t}_n^{(k)} \) can be written as a Schur complement

\[
\mathbf{t}_n^{(k)} = \mathbf{s}_n - \left[ \Delta \mathbf{s}_n, \ldots, \Delta \mathbf{s}_{n+k-1} \right] \mathbf{Y}^T \mathbf{\Delta}_{\mathbf{T}^{(k)}_n}^{-1} \mathbf{Y}^T
\]

with

\[
\mathbf{Y} = \left[ \mathbf{y}_1, \ldots, \mathbf{y}_k \right], \quad \mathbf{\Delta}_{\mathbf{T}^{(k)}_n}, 1
\]

is the first column of the matrix \( \mathbf{T}^{(k)}_n \) (that is \( \mathbf{t}_n \) in this case).

Remark: The notation \( \mathbf{T}^{(k)}_n \) means that the matrix has \( k \) columns and that its first column is \( \mathbf{t}_n \).
1. The polynomial extrapolation strategy (Pol)

\[
\begin{vmatrix}
    s_n & \Delta s_n & \cdots & \Delta s_{n+k-1} \\
    \langle y_1, t_n \rangle & \langle y_1, \Delta t_n \rangle & \cdots & \langle y_1, \Delta t_{n+k-1} \rangle \\
    \vdots & \vdots & \ddots & \vdots \\
    \langle y_k, t_n \rangle & \langle y_k, \Delta t_n \rangle & \cdots & \langle y_k, \Delta t_{n+k-1} \rangle \\
\end{vmatrix}
= \frac{\langle y_1, \Delta t_n \rangle \cdots \langle y_1, \Delta t_{n+k-1} \rangle}{\langle y_1, \Delta t_n \rangle \cdots \langle y_1, \Delta t_{n+k-1} \rangle}.
\]

Thus, according to the extended Schur determinantal formula, \( t_n^{(k)} \) can be written as a Schur complement

\[
t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T_n^{(k)})^{-1} Y^T T_n^{(k)},
\]

with \( Y = [y_1, \ldots, y_k] \), \( T_n^{(k)} = [t_n, \ldots, t_{n+k-1}] \), and where \( T_{n,1}^{(k)} \) is the first column of the matrix \( T_n^{(k)} \) (that is \( t_n \) in this case).

Remark: The notation \( T_n^{(k)} \) means that the matrix has \( k \) columns and that its first column is \( t_n \).
Let again $\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0$.

We will now outline the Shanks’ strategy followed by Shanks to obtain his scalar sequence transformation. Shanks considered extracting the $\alpha_i$’s by solving the system of linear equations

$$
\begin{align*}
\alpha_0 + \cdots + \alpha_k &= 1 \\
\alpha_0 \langle y, t_{n+i} \rangle + \cdots + \alpha_k \langle y, t_{n+k+i} \rangle &= 0, \quad i = 0, \ldots, k - 1.
\end{align*}
$$

where now $y$ is a fixed vector, and the index $i$ varies.
It follows that this transformation is

\[
\begin{bmatrix}
  s_n & \cdots & s_{n+k} \\
  \langle y, t_n \rangle & \cdots & \langle y, t_{n+k} \rangle \\
  \vdots & & \vdots \\
  \langle y, t_{n+k-1} \rangle & \cdots & \langle y_k, t_{n+2k-1} \rangle
\end{bmatrix}
\begin{bmatrix}
  1 & \cdots & 1 \\
  \langle y, t_n \rangle & \cdots & \langle y, t_{n+k} \rangle \\
  \vdots & & \vdots \\
  \langle y_k, t_{n+k-1} \rangle & \cdots & \langle y, t_{n+2k-1} \rangle
\end{bmatrix}^{-1} =
\begin{bmatrix}
  s_n & \Delta s_n & \cdots & \Delta s_{n+k-1} \\
  \langle y, t_n \rangle & \langle y, \Delta t_{n} \rangle & \cdots & \langle y, \Delta t_{n+k-1} \rangle \\
  \vdots & \vdots & & \vdots \\
  \langle y, t_{n+k-1} \rangle & \langle y, \Delta t_{n+k-1} \rangle & \cdots & \langle y, \Delta t_{n+2k-2} \rangle
\end{bmatrix}
\begin{bmatrix}
  \langle y, \Delta t_n \rangle & \cdots & \langle y, \Delta t_{n+k-1} \rangle \\
  \vdots & & \vdots \\
  \langle y, \Delta t_{n+k-1} \rangle & \cdots & \langle y, \Delta t_{n+2k-2} \rangle
\end{bmatrix}.
\]
Thus, according to the extended Schur determinantal formula, \( t_n^{(k)} \) can be written as a \textbf{Schur complement} (a new result)

\[
t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T_n^{(k)})^{-1} Y^T T_n^{(k)},
\]

with now

\[
Y = \begin{pmatrix}
y & z & \cdots & z \\
z & y & \cdots & z \\
\vdots & \vdots & \ddots & \vdots \\
z & z & \cdots & y
\end{pmatrix},
\]

\[
T_n^{(k)} = \begin{pmatrix}
t_n & t_{n+1} & \cdots & t_{n+k-1} \\
t_{n+1} & t_{n+2} & \cdots & t_{n+k} \\
\vdots & \vdots & \ddots & \vdots \\
t_{n+k-1} & t_{n+k} & \cdots & t_{n+2k-2}
\end{pmatrix},
\]

where \( z = \mathbf{0} \in \mathbb{E}^* \), and where \( T_n^{(k)} \) denotes the first column of the matrix \( T_n^{(k)} \) as before.
The accelerated sequence can also be written as

\[ t_n^{(k)} = s_n - \sum_{i=1}^{k} \beta_i \Delta s_{n+i-1}, \]

in which \( \beta_i = -(\alpha_i + \cdots + \alpha_k) \) for \( i = 1, \ldots, k \).
3. The least-squares strategy (Lsq)

The accelerated sequence can also be written as

\[ t^{(k)}_n = s_n - \sum_{i=1}^{k} \beta_i \Delta s_{n+i-1}, \]

in which \( \beta_i = -(\alpha_i + \cdots + \alpha_k) \) for \( i = 1, \ldots, k \).

Proceeding similarly for the sequence \( (t_n) \), we would obtain the relation

\[ t_n - \sum_{i=1}^{k} \beta_i \Delta t_{n+i-1} = 0. \]

The vector \( b = (\beta_1, \ldots, \beta_k)^T \in \mathbb{R}^k \) is obtained by solving the \((p + 1) \times k\) least-squares system

\[ [\Delta t_n, \ldots, \Delta t_{n+k-1}] b =_{LS} t_n. \]
It then follows that \textbf{the sequence transformation is given by}

\[ t^{(k)}_n = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] [(\Delta T^{(k)}_n)^T \Delta T^{(k)}_n]^{-1} (\Delta T^{(k)}_n)^T t_n. \]

Since \( t_n = T^{(k)}_{n,1} \), this formula is a \textbf{particular case of the polynomial extrapolation strategy} with, now, \( Y = \Delta T^{(k)}_n \).
3. The least-squares strategy (Lsq)

It then follows that the sequence transformation is given by
\[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}][\Delta T_n^{(k)}]^T \Delta T_n^{(k)}]^{-1}(\Delta T_n^{(k)})^T t_n. \]

Since \( t_n = T_n^{(k)} \), this formula is a particular case of the polynomial extrapolation strategy with, now, \( Y = \Delta T_n^{(k)} \).

By the Schur determinantal formula, we also have
\[ t_n^{(k)} = \begin{vmatrix} s_n & \Delta s_n \cdots \Delta s_{n+k-1} \\ (\Delta T_n^{(k)})^T t_n & (\Delta T_n^{(k)})^T \Delta T_n^{(k)} \end{vmatrix} / (\Delta T_n^{(k)})^T \Delta T_n^{(k)}. \]
We will now discuss the choice of the coupled sequence \((t_n)\) that satisfies \(\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0\). There are two common ways of selecting this sequence.

**General choice (Gen)**
Writing the difference equation of Shanks' kernel \(\alpha_0 (s_n - s_{n-1}) + \cdots + \alpha_k (s_{n+k} - s_{n+k-1}) = 0\) for the indices \(n+1\) and \(n\), and subtracting, we see that, for example, the sequence \(t_n = \Delta s_n\) satisfies the coupled relation.

**Fixed-point choice (Fxp)**
Consider the fixed point problem \(s = g(s)\) in \(E\), or equivalently, the nonlinear equation \(f(s) = 0\). We select \(t_n = f(s_n)\) where \((s_n)\) is an arbitrary sequence.
We will now discuss the choice of the coupled sequence \((t_n)\) that satisfies \(\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0\).

There are two common ways of selecting this sequence.

- **General choice (Gen)**
  Writing the difference equation of Shanks’ kernel
  \(\alpha_0 (s_n - s) + \cdots + \alpha_k (s_{n+k} - s) = 0\)
  for the indices \(n+1\) and \(n\), and subtracting, we see that, for example, the sequence \(t_n = \Delta s_n\) satisfies the coupled relation.
We will now discuss the choice of the coupled sequence \((t_n)\) that satisfies \(\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0\).

There are two common ways of selecting this sequence.

- **General choice (Gen)**
  Writing the difference equation of Shanks’ kernel
  \[\alpha_0(s_n - s) + \cdots + \alpha_k(s_{n+k} - s) = 0\]
  for the indices \(n+1\) and \(n\), and subtracting, we see that, for example, the sequence \(t_n = \Delta s_n\) satisfies the coupled relation.

- **Fixed-point choice (Fxp)**
  Consider the fixed point problem \(s = g(s)\) in \(E\), or equivalently, the nonlinear equation \(f(s) = 0 \in E\). We select \(t_n = f(s_n)\) where \((s_n)\) is an arbitrary sequence.
Summary of the CTST

The various transformations derived are denoted by the generic term **Coupled Topological Shanks’ Transformation (CTST)**. Each method depends on **two selections**.

- **Three possible strategies for writing the linear system that yields the coefficients** $\alpha_i$:
  - polynomial extrapolation strategy,
  - Shanks’ strategy,
  - least squares-strategy.

- **Two possibilities for choosing the coupled sequence** $(t_n)$:
  - general choice,
  - fixed point choice.

Thus, we obtain **six classes of transformations**

<table>
<thead>
<tr>
<th></th>
<th>Polynomial</th>
<th>Shanks</th>
<th>Least-squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_n$ : General</td>
<td><strong>Pol-Gen</strong></td>
<td><strong>Sha-Gen</strong></td>
<td><strong>Lsq-Gen</strong></td>
</tr>
<tr>
<td>$t_n$ : Fixed Point</td>
<td><strong>Pol-Fxp</strong></td>
<td><strong>Sha-Fxp</strong></td>
<td><strong>Lsq-Fxp</strong></td>
</tr>
</tbody>
</table>
We consider the **Polynomial extrapolation strategy**

\[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T_n^{(k)})^{-1} Y^T T_{n,1}^{(k)}, \]

with \( Y = [y_1, \ldots, y_k] \), \( T_n^{(k)} = [t_n, \ldots, t_{n+k-1}] \), and \( T_{n,1}^{(k)} = t_n \) and we set \( S_n^{(k)} = [s_n, \ldots, s_{n+k-1}] \).
We consider the **Polynomial extrapolation strategy**

\[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}](Y^T \Delta T_n^{(k)})^{-1} Y^T T_n^{(k)}, \]

with \( Y = [y_1, \ldots, y_k] \), \( T_n^{(k)} = [t_n, \ldots, t_{n+k-1}] \), and \( T_{n,1}^{(k)} = t_n \) and we set \( S_n^{(k)} = [s_n, \ldots, s_{n+k-1}] \).

- **The Minimal Polynomial Extrapolation (MPE):**
  
  \[ y_i = \Delta s_{n+i-1} \quad \text{and} \quad t_n = \Delta s_n \]
  
  \[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}][\Delta S_n^{(k)} T \Delta^2 S_n^{(k)}]^{-1} (\Delta S_n^{(k)})^T \Delta s_n. \]
Particular methods (Pol-Gen)

We consider the **Polynomial extrapolation strategy**

\[
t_{n}^{(k)} = s_{n} - [\Delta s_{n}, \ldots, \Delta s_{n+k-1}] (Y^T \Delta \Delta T_{n}^{(k)})^{-1} Y^T T_{n,1}^{(k)},
\]

with \( Y = [y_1, \ldots, y_k] \), \( T_{n}^{(k)} = [t_n, \ldots, t_{n+k-1}] \), and \( T_{n,1}^{(k)} = t_n \) and we set \( S_{n}^{(k)} = [s_{n}, \ldots, s_{n+k-1}] \).

- **The Minimal Polynomial Extrapolation (MPE):**
  \[y_i = \Delta s_{n+i-1} \text{ and } t_n = \Delta s_n\]
  \[t_{n}^{(k)} = s_{n} - [\Delta s_{n}, \ldots, \Delta s_{n+k-1}] [(\Delta S_{n}^{(k)})^T \Delta^2 S_{n}^{(k)}]^{-1} (\Delta S_{n}^{(k)})^T \Delta s_n.\]

- **The Reduced Rank Extrapolation (RRE):**
  \[y_i = \Delta^2 s_{n+i-1} \text{ and } t_n = \Delta s_n\]
  \[t_{n}^{(k)} = s_{n} - [\Delta s_{n}, \ldots, \Delta s_{n+k-1}] [(\Delta^2 S_{n}^{(k)})^T \Delta^2 S_{n}^{(k)}]^{-1} (\Delta^2 S_{n}^{(k)})^T \Delta s_n.\]
We consider the **Polynomial extrapolation strategy**

\[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T_n^{(k)})^{-1} Y^T T_{n,1}^{(k)}, \]

with \( Y = [y_1, \ldots, y_k] \), \( T_n^{(k)} = [t_n, \ldots, t_{n+k-1}] \), and \( T_{n,1}^{(k)} = t_n \) and we set \( S_n^{(k)} = [s_n, \ldots, s_{n+k-1}] \).

- **The Minimal Polynomial Extrapolation (MPE):**
  \[ y_i = \Delta s_{n+i-1} \text{ and } t_n = \Delta s_n \]
  \[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] [(\Delta S_n^{(k)})^T \Delta^2 S_n^{(k)}]^{-1} (\Delta S_n^{(k)})^T \Delta s_n. \]

- **The Reduced Rank Extrapolation (RRE):**
  \[ y_i = \Delta^2 s_{n+i-1} \text{ and } t_n = \Delta s_n \]
  \[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] [(\Delta^2 S_n^{(k)})^T \Delta^2 S_n^{(k)}]^{-1} (\Delta^2 S_n^{(k)})^T \Delta s_n. \]

- **The Modified Minimal Polynomial Extrapolation (MMPE):**
  \[ y_i \text{'s linearly independent linear functionals and } t_n = \Delta s_n \]
  \[ t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta^2 S_n^{(k)})^{-1} Y^T \Delta s_n. \]
The Topological Shanks Transformation (TEA or STEA algorithm) belongs to the Shanks strategy, with \( y \in E^* \) fixed and \( t_n = \Delta s_n \).

\[
\begin{align*}
t^{(k)}_n &= s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T^{(k)}_n)^{-1} Y^T T^{(k)}_{n,1}, \\
Y &= \begin{pmatrix}
y & z & \cdots & z \\
z & y & \cdots & z \\
\vdots & \vdots & \ddots & \vdots \\
z & z & \cdots & y
\end{pmatrix}, \\
T^{(k)}_n &= \begin{pmatrix}
s_n & s_{n+1} & \cdots & s_{n+k-1} \\
s_{n+1} & s_{n+2} & \cdots & s_{n+k} \\
\vdots & \vdots & \ddots & \vdots \\
s_{n+k-1} & s_{n+k} & \cdots & s_{n+2k-2}
\end{pmatrix}, \\
z &= 0 \in E^*, \\
T^{(k)}_{n,1} & \text{ first column of the matrix } T^{(k)}_n.
\end{align*}
\]
Particular methods (Sha-Gen)

- The **Topological Shanks Transformation (TEA or STEA algorithm)** belongs to the **Shanks strategy**, with $y \in E^*$ fixed and $t_n = \Delta s_n$.

\[
    t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] (Y^T \Delta T_n^{(k)})^{-1} Y^T T_n^{(k)},
\]

\[
    Y = \begin{pmatrix}
    y & z & \cdots & z \\
    z & y & \cdots & z \\
    \vdots & \vdots & \ddots & \vdots \\
    z & z & \cdots & y
    \end{pmatrix},
    \quad
    T_n^{(k)} = \begin{pmatrix}
    s_n & s_{n+1} & \cdots & s_{n+k-1} \\
    s_{n+1} & s_{n+2} & \cdots & s_{n+k} \\
    \vdots & \vdots & \ddots & \vdots \\
    s_{n+k-1} & s_{n+k} & \cdots & s_{n+2k-2}
    \end{pmatrix},
\]

$z = 0 \in E^*$, $T_{n,1}^{(k)}$ first column of the matrix $T_n^{(k)}$.

**Remark:** The MMPE and the (S)TEA can treat sequences of elements of a general vector space (vectors, matrices, tensors, . . .), while the MPE, and the RRE can only be applied to vector and to matrices sequences (in this last case, by using the Frobenius inner product as duality product).
In all cases, the formulae for $t_n^{(k)}$ have the same structure, independently from the choice of $y$ and $y_i$, namely

$$t_n^{(k)} = s_n - [\Delta s_n, \ldots, \Delta s_{n+k-1}] \gamma,$$

where $\gamma$ is the solution of the system

$$(Y^T \Delta T_n^{(k)}) \gamma = Y^T T_{n,1}^{(k)}$$
The preceding result can be generalized by isolating any column $i$ in the determinants, and it leads to this new result.

**Theorem**

The following expression holds for any $i = 0, \ldots, k$,

$$
t_n^{(k)} = \begin{vmatrix} s_{n+i} & \Delta S_n^{(k)} & \left/ Y^T \Delta T_n^{(k)} \right. \\ Y^T T_n^{(k)} & Y^T \Delta T_n^{(k)} & \end{vmatrix}
$$

with $\Delta S_n^{(k)} = [\Delta s_n, \ldots, \Delta s_{n+k-1}]$, that is

$$
t_n^{(k)} = s_{n+i} - \Delta S_n^{(k)} \gamma(i),
$$

with

$$
\gamma(i) = (Y^T \Delta T_n^{(k)})^{-1} Y^T T_n^{(k)}.
$$
Properties of CTST

From the previous Theorem it is possible to state Galerkin orthogonality conditions valid for all coupled topological Shanks’ transformations, which establishes a link with projection methods.

**Theorem**

We set

\[
\tilde{t}^{(k)}_n = s_{n+i+1} - [\Delta s_{n+1}, \ldots, \Delta s_{n+k}] \gamma^{(i)}, \quad i = 0, \ldots, k - 1,
\]

where \( \gamma^{(i)} \) is the solution of the system

\[
(Y^T \Delta T_n^{(k)}) \gamma^{(i)} = Y^T T_n^{(k)} T_{n+i,1}.
\]

If \( \forall n, t_n = \Delta s_n \), then

\[
Y^T (\tilde{t}^{(k)}_n - t^{(k)}_n) = 0.
\]
We have the following well known result, valid also for all the CTST, showing that these transformations provide *direct methods for solving systems of linear equations*. 

**Theorem**

Assume that the $s_n$’s are vectors in $\mathbb{R}^p$ that are generated by the linear recurrence

$$s_{n+1} = Ms_n + d,$$

$s_0$ arbitrary, where $I - M$ is invertible.

Then for all three strategies, with $t_n = \Delta s_n$, $\forall n$, we have

$$t_0^{(m)} = s = (I - M)^{-1}d,$$

where $m$ is the degree of the minimal polynomial of $M$ for the vector $s_0 - s$. 

Other properties:

- The **TEA**, the **RRE**, the **MMPE**, and the **MPE** are **Krylov subspace methods**.
- The sequence \((x_k)\) obtained by **Lanczos’ method** for solving the system of linear equations \(Ax = (I - M)x = d\) starting from \(x_0\) (which can be implemented by the **biconjugate gradient algorithm** of Fletcher), and the sequence \((t_{(k)}^0)\) obtained by applying the **TEA** with \(y = r_0 = (I - M)x_0 - d\) to the sequence generated by \(s_{n+1} = Ms_n + d\) with \(s_0 = x_0\) are identical.
Consider a system of *p nonlinear equations* in *p* unknowns

\[ f(x) = g(x) - x = 0 \in \mathbb{R}^p. \]
Consider a system of \( p \) nonlinear equations in \( p \) unknowns

\[
f(x) = g(x) - x = 0 \in \mathbb{R}^p.
\]

We consider the following iterative method for computing the fixed point \( x \) of \( g \)

1. Set \( s_0 = x_n \).
2. Compute \( s_{i+1} = g(s_i) \) for \( i = 0, \ldots, k - 1 \).
3. Apply the polynomial strategy with the choice 
   \( t_i = g(s_i) - s_i = \Delta s_i \) (\( Pol-Fxp \) or \( Pol-Gen \)) to the iterates \( s_i \), and compute
   \[
   t_0^{(k)} = s_0 - [\Delta s_0, \ldots, \Delta s_{k-1}](Y^T \Delta T_0^{(k)})^{-1}Y^T T_{0,1}^{(k)}.
   \]
4. Set \( x_{n+1} = t_0^{(k)} \).

and then restart the iterative method with \( s_0 = x_{n+1} \).
Any of the preceding methods can be considered as a **quasi-Newton method** with

\[ G_n = [\Delta s_0, \ldots, \Delta s_{k-1}](Y^T \Delta T_0^{(k)})^{-1} Y^T \in \mathbb{R}^{p \times p}, \]

since \( T_0^{(k)} = t_0 = f(x_n) \) and in the preceding iterative method we have

\[ x_{n+1} = x_n - G_n f(x_n), \]

where \( G_n \) is an approximation of \([f'(x_n)]^{-1}\).
The quasi-Newton connection

Any of the preceding methods can be considered as a
**quasi-Newton method** with

\[ G_n = [\Delta s_0, \ldots, \Delta s_{k-1}](Y^T \Delta T_0^{(k)})^{-1} Y^T \in \mathbb{R}^{p \times p}, \]

since \( T_{0,1}^{(k)} = t_0 = f(x_n) \) and in the preceding iterative method we have

\[ x_{n+1} = x_n - G_n f(x_n), \]

where \( G_n \) is an approximation of \( [f'(x_n)]^{-1} \).

**Remark:** A similar restarting procedure with the other methods described above leads to methods that, under some assumptions, converge quadratically to the fixed point \( x \) of \( g \) when \( k = p \) (H. Le Ferrand, K. Jbilou, H. Sadok, A. Messaoudi, ...).
The quasi-Newton connection

Any of the preceding methods can be considered as a quasi-Newton method with

$$G_n = [\Delta s_0, \ldots, \Delta s_{k-1}] (Y^T \Delta T_{0}^{(k)})^{-1} Y^T \in \mathbb{R}^{p \times p},$$

since $T_{0,1}^{(k)} = t_0 = f(x_n)$ and in the preceding iterative method we have

$$x_{n+1} = x_n - G_n f(x_n),$$

where $G_n$ is an approximation of $[f'(x_n)]^{-1}$.

**Remark:** A similar restarting procedure with the other methods described above leads to methods that, under some assumptions, converge quadratically to the fixed point $x$ of $g$ when $k = p$ (H. Le Ferrand, K. Jbilou, H. Sadok, A. Messaoudi, ...).

We will now consider some particular cases of these methods, when applied to systems of equations.
If in the preceding restarted iterative method we consider the Modified Minimal Polynomial Extrapolation (MMPE) \( t_n = \Delta s_n \), with \( k = p \) and \( y_i = e_i \) (the vectors of the canonical basis of \( \mathbb{R}^p \)), we obtain a method due to Henrici (El. Numer. Anal., 1964).

It can be recursively implemented by the \textit{H-algorithm} (C. Brezinski, H. Sadok) or by the \textit{S\beta-algorithm} (K. Jbilou).
If in the preceding restarted iterative method we consider the Modified Minimal Polynomial Extrapolation (MMPE) ($t_n = \Delta s_n$), with $k = p$ and $y_i = e_i$ (the vectors of the canonical basis of $\mathbb{R}^p$), we obtain a method due to Henrici (El. Numer. Anal., 1964).

It can be recursively implemented by the $H$-algorithm (C. Brezinski, H. Sadok) or by the $S\beta$-algorithm (K. Jbilou).
Let me consider the Reduced Rank Extrapolation (RRE) (with \(t_{n+i} = \Delta s_{n+i}, y_i = \Delta^2 s_{n+i-1}\) for \(i = 0, \ldots, k - 1\)).

We fix \(n = 0\). Thus we use the forward differences \(\Delta s_0, \ldots, \Delta s_k\).

In the linear case, \(t_0^{(k)}\) is the solution obtained at the k-th step of the full GMRES.

If the linear iterations are restarted from \(t_0^{(k)}\), then the RRE and GMRES\((k)\) are mathematically equivalent.

Thus GMRES can also be considered as a quasi-Newton method.
Anderson Acceleration (AA) is aimed at the solution of systems of nonlinear equations \( f(x) = g(x) - x = 0 \) by constructing its own sequence of vectors \((x_k)\).
Anderson Acceleration (AA) is aimed at the solution of systems of nonlinear equations $f(x) = g(x) - x = 0$ by constructing its own sequence of vectors $(x_k)$.

AA first consists of choosing $x_0$, $m \geq 1$ and computing $x_1 = g(x_0) = x_0 + f_0$, where for $i = 0, 1, \ldots$, we define $f_i = f(x_i)$. After, we have to construct the new iterates $x_{k+1}$ for $k = 1, 2, \ldots$. 
Anderson Acceleration (AA) is aimed at the solution of systems of nonlinear equations \( f(x) = g(x) - x = 0 \) by constructing its own sequence of vectors \((x_k)\).

AA first consists of choosing \(x_0, m \geq 1\) and computing \(x_1 = g(x_0) = x_0 + f_0\), where for \(i = 0, 1, \ldots\), we define \(f_i = f(x_i)\).

After, we have to construct the new iterates \(x_{k+1}\), for \(k = 1, 2, \ldots\). We set \(m_k = \min(m, k)\) and we denote

\[
X_k = [x_{k-m_k}, \ldots, x_{k-1}], \quad F_k = [f_{k-m_k}, \ldots, f_{k-1}],
\]
Anderson Acceleration (AA) is aimed at the solution of systems of nonlinear equations \( f(x) = g(x) - x = 0 \) by constructing its own sequence of vectors \((x_k)\).

AA first consists of choosing \(x_0, m \geq 1\) and computing \(x_1 = g(x_0) = x_0 + f_0\), where for \(i = 0, 1, \ldots\), we define \(f_i = f(x_i)\).

After, we have to construct the new iterates \(x_{k+1}\), for \(k = 1, 2, \ldots\). We set \(m_k = \min(m, k)\) and we denote

\[
X_k = [x_{k-m_k}, \ldots, x_{k-1}], \quad F_k = [f_{k-m_k}, \ldots, f_{k-1}],
\]

We compute the vector \(\theta^{(k)}\) that solves

\[
\min_{\theta \in \mathbb{R}^{m_k}} \| f_k - \Delta F_k \theta \|_2.
\]

that is

\[
\theta^{(k)} = (\Delta F_k^T \Delta F_k)^{-1} \Delta F_k^T f_k.
\]
Anderson Acceleration method

We have

\[
\bar{x}_k = x_k - \sum_{i=k-m}^{k-1} \theta_i^{(k)} \Delta x_i = x_k - \Delta X_k \theta^{(k)}
\]

\[
\bar{f}_k = f_k - \sum_{i=k-m}^{k-1} \theta_i^{(k)} \Delta f_i = f_k - \Delta F_k \theta^{(k)}.
\]

Then, the next iterate of Anderson’s method is computed as

\[
x_{k+1} = \bar{x}_k + \beta_k \bar{f}_k = x_k + \beta_k f_k - (\Delta X_k + \beta_k \Delta F_k) \theta^{(k)},
\]

where \( \beta_k \) is a parameter, usually positive (often \( \beta_k = 1 \)).
Anderson Acceleration method

We have

\[ \bar{x}_k = x_k - \sum_{i=k-m_k}^{k-1} \theta_i^{(k)} \Delta x_i = x_k - \Delta X_k \theta^{(k)} \]

\[ \bar{f}_k = f_k - \sum_{i=k-m_k}^{k-1} \theta_i^{(k)} \Delta f_i = f_k - \Delta F_k \theta^{(k)}. \]

Then, the next iterate of Anderson’s method is computed as

\[ x_{k+1} = \bar{x}_k + \beta_k \bar{f}_k = x_k + \beta_k f_k - (\Delta X_k + \beta_k \Delta F_k) \theta^{(k)}, \]

where \( \beta_k \) is a parameter, usually positive (often \( \beta_k = 1 \)).

Anderson Acceleration relates to the polynomial extrapolation strategy with the Fixed point choice!
In fact, if we take

\[ Y = \Delta F_k, \quad t_i = f_i, \]

- with \( s_i = x_i \), we have \( t_{k-m_k}^{(m_k)} = x_k \).
- with \( s_i = f_i \), we have \( t_{k-m_k}^{(m_k)} = f_k \).

**Remark:** RRE is a variant of AA since, in these two methods, the vectors \( \theta^{(k)} \) are chosen in a slightly different way.
Anderson Acceleration method

**Anderson Acceleration algorithm**

Choose $\mathbf{x}_0$, $\beta_0$ and $m \geq 1$.

Compute $\mathbf{x}_1 = \mathbf{x}_0 + \beta_0 \mathbf{f}_0$

For $k = 1, 2, \ldots$

Choose $\beta_k$ and set $m_k = \min(m, k)$.

Compute $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_k)$

Compute

$$\theta^{(k)} = \left( (\Delta \mathbf{F}_k)^T \Delta \mathbf{F}_k \right)^{-1} (\Delta \mathbf{F}_k)^T \mathbf{f}_k$$

$$\bar{\mathbf{x}}_k = \mathbf{x}_k - \sum_{i=k-m_k}^{k-1} \theta^{(k)}_i \Delta \mathbf{x}_i = \mathbf{x}_k - \Delta \mathbf{X}_k \theta^{(k)}$$

$$\bar{\mathbf{f}}_k = \mathbf{f}_k - \sum_{i=k-m_k}^{k-1} \theta^{(k)}_i \Delta \mathbf{f}_i = \mathbf{f}_k - \Delta \mathbf{F}_k \theta^{(k)}$$

with $\mathbf{X}_k = [\mathbf{x}_{k-m_k}, \ldots, \mathbf{x}_{k-1}]$, $\mathbf{F}_k = [\mathbf{f}_{k-m_k}, \ldots, \mathbf{f}_{k-1}]$.

Set $\mathbf{x}_{k+1} = \bar{\mathbf{x}}_k + \beta_k \bar{\mathbf{f}}_k$.

end
Anderson-type Acceleration methods

**Anderson-type Acceleration algorithm**

Choose $x_0$, $\beta_0$ and $m \geq 1$.
Compute $x_1 = x_0 + \beta_0 f_0$

For $k = 1, 2, \ldots$

Choose $\beta_k$ and set $m_k = \min(m, k)$.
Compute $f_k = f(x_k)$

Compute

\[
\theta^{(k)} = \left[ Y^T \Delta F_k \right]^{-1} Y^T f_k
\]

\[
\bar{x}_k = x_k - \sum_{i=k-m_k}^{k-1} \theta_i^{(k)} \Delta x_i = x_k - \Delta X_k \theta^{(k)}
\]

\[
\bar{f}_k = f_k - \sum_{i=k-m_k}^{k-1} \theta_i^{(k)} \Delta f_i = f_k - \Delta F_k \theta^{(k)}
\]

with $X_k = [x_{k-m_k}, \ldots, x_{k-1}]$, $F_k = [f_{k-m_k}, \ldots, f_{k-1}]$.
Set $x_{k+1} = \bar{x}_k + \beta_k \bar{f}_k$.

end

with $Y$ chosen in **MPE, MMPE, TEA** style (AA looks like the **RRE** choice $Y = \Delta F_k$).
A semi-regularization technique can be applied.
Applications of coupled topological Shanks’ transformations to systems of equations, the computation of matrix functions, and the solution of integral equations . . . to be continued.

Stable implementation of the MMPE.

Matrix Shanks’ transformations (paper submitted).

C. Brezinski, M. Redivo Zaglia, Y. Saad, *Applications of coupled topological Shanks transformations to systems of equations, in progress.*
Thank you!