AN ADAPTIVE SHIFTED POWER METHOD FOR COMPUTING GENERALIZED TENSOR EIGENPAIRS

TAMARA G. KOLDA† AND JACKSON R. MAYO†

Abstract. Several tensor eigenpair definitions have been put forth in the past decade, but these can all be unified under generalized tensor eigenpair framework, introduced by Chang, Pearson, and Zhang (2009). Given mth-order, n-dimensional real-valued symmetric tensors A and B, the goal is to find \( \lambda \in \mathbb{R} \) and \( x \in \mathbb{R}^n, x \neq 0 \) such that \( Ax^m = \lambda Bx^m \). Different choices for B yield different versions of the tensor eigenvalue problem. We present our generalized eigenproblem adaptive power method (GEAP) method for solving the problem, which is an extension of the shifted symmetric higher-order power method (SS-HOPM) for finding Z-eigenpairs. A major drawback of SS-HOPM was that its performance depended in choosing an appropriate shift, but our GEAP method also includes an adaptive method for choosing the shift automatically.

Key words. tensor eigenvalues, E-eigenpairs, Z-eigenpairs, \( l^2 \)-eigenpairs, generalized tensor eigenpairs, shifted symmetric higher-order power method (SS-HOPM), generalized eigenproblem adaptive power (GEAP) method

AMS subject classifications. 15A18, 15A69

1. Introduction. Suppose \( A \) is a real-valued, mth-order, n-dimensional tensors and \( x \) is a real-valued n-vector. We let \( Ax^{m-1} \) denote the n-vector defined by

\[
(Ax^{m-1})_{i_1} = \sum_{i_2=1}^{n} \cdots \sum_{i_m=1}^{n} a_{i_1 \ldots i_m} x_{i_2} \cdots x_{i_m} \quad \text{for} \quad i_1 = 1, \ldots, n.
\]

We let \( Ax^m \) denote the scalar defined by \( Ax^m = x^T (Ax^{m-1}) \). We say the tensor \( A \) is symmetric if its entries are invariant under permutation. We say the tensor \( A \) is positive definite if \( Ax^m > 0 \) for all \( x \neq 0 \).

The notion of generalized eigenpairs has been defined for tensors by Chang, Pearson, and Zhang [1] as follows. Let \( A \) and \( B \) be real-valued, mth-order, n-dimensional symmetric tensors. Assume further that \( m \) is even and \( B \) is positive definite. We say \( (\lambda, x) \in \mathbb{R} \times \{ \mathbb{R}^n \setminus \{0\} \} \) is a generalized eigenpair (also known as a \( B \)-eigenpair) if

\[
Ax^{m-1} = \lambda Bx^{m-1}.
\]

Taking the dot product with \( x \), it is clear that any solution satisfies

\[
\lambda = \frac{Ax^m}{Bx^m}.
\]

The advantage of the generalized eigenpair framework is that it nicely encapsulates multiple definitions of tensor eigenvalues, as follows.

- A Z-eigenpair [7, 4] is defined as a pair \( (\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \) such that

\[
Ax^{m-1} = \lambda x \quad \text{and} \quad ||x|| = 1.
\]
This is equivalent to a generalized tensor eigenpair with $\mathcal{B} = \mathcal{E}$, the identity tensor such that $\mathcal{E} x^{m-1} = \|x\|^{m-2} x$ for all $x \in \mathbb{R}^n$ [1]. Note that, unlike ordinary tensor $Z$-eigenpairs, generalized tensor eigenpairs allow arbitrary rescaling of the eigenvector $x$ with no effect on the eigenvalue $\lambda$. In this way, the generalized tensor eigenvalue problem preserves the homogeneity of the corresponding matrix eigenproblem.

- An $H$-eigenpair is defined as a pair $(\lambda, x) \in \mathbb{R} \times \{\mathbb{R}^n \setminus \{0\}\}$ such that
  \[ \mathcal{A} x^{m-1} = \lambda x^{[m]} . \]  

Here $x^{[m]}$ denotes elementwise power, i.e., $(x^{[m]})_i \equiv x_i^m$, for $i = 1, \ldots, n$. This is equivalent to a generalized tensor eigenpair with $b_{i_1 i_2 \ldots i_m} = \delta_{i_1 i_2 \ldots i_m}$ [1].

- Let $\mathbf{D}$ be a symmetric $n \times n$ matrix and assume $m = 4$. We say $(\mathbf{A}, \mathbf{x})$ is a $D$-eigenpair [8] if
  \[ \mathbf{A} x^{m-1} = \lambda \mathbf{D} x \quad \text{and} \quad x^T \mathbf{D} x = 1. \]

This is equivalent to a $\mathcal{B}$-eigenpair where $\mathcal{B}$ is the symmetrized tensor outer product of $\mathbf{D}$ with itself [1].

In this paper, we describe a method for computing generalized eigenpairs. Our method is a generalization of the shifted symmetric higher-order power method (SS-HOPM) that we previously introduced for computing $Z$-eigenvalues [3]. In addition to generalizing the method, we have also significantly improved it by adding an adaptive method for choosing the shift. To derive the method, we reformulate the generalized eigenproblem, (1.1), as a nonlinear program such that any generalized eigenpair is equivalent to a KKT point in §3. We develop an adaptive, monotonically convergent, shifted power method for solving the optimization problem in §4. We call our method the Generalized Eigenproblem Adaptive Power (GEAP) method. In §5, we show that the GEAP method is much faster than the SS-HOPM method for finding $Z$-eigenpairs due to its adaptive shift selection. Furthermore, the GEAP method is shown to find other types of generalized eigenpairs, by illustrating it on examples from related literature as well as a randomly generated example. This is the only known method for finding generalized eigenpairs besides direct numerical solution; we survey related work in §6.

2. Notation and preliminaries. A symmetric tensor has entries that are invariant under any permutation of its indices. More formally, a real-valued, $m$th-order, $n$-dimensional tensor $\mathcal{A}$ is symmetric if
  \[ a_{i_{p(1)} \ldots i_{p(m)}} = a_{i_1 \ldots i_m} \text{ for all } i_1, \ldots, i_m \in \{1, \ldots, n\} \text{ and } p \in \Pi_m, \]
where $\Pi_m$ denotes the space of all $m$-permutations. We let $\mathcal{S}^{[m,n]}$ denote the space of all symmetric, real-valued, $m$th-order, $n$-dimensional tensors.

Let $\mathcal{A} \in \mathcal{S}^{[m,n]}$, then we can define the following tensor-vector products.

\[ \mathcal{A} x^m = \sum_{i_1} \cdots \sum_{i_m=1}^n a_{i_1 i_2 \ldots i_m} x_{i_1} \cdots x_{i_m}, \]  

\[ (\mathcal{A} x^{m-1})_{i_1} = \sum_{i_2} \cdots \sum_{i_m=1}^n a_{i_1 i_2 \ldots i_m} x_{i_2} \cdots x_{i_m} \quad \text{for all } i_1 = 1, \ldots, n, \]

\[ (\mathcal{A} x^{m-2})_{i_1 i_2} = \sum_{i_3} \cdots \sum_{i_m=1}^n a_{i_1 i_2 i_3 \ldots i_m} x_{i_3} \cdots x_{i_m} \quad \text{for all } i_1, i_2 = 1, \ldots, n. \]
Observe that the derivatives of the tensor-vector product w.r.t. $x$ are given by
\[
\nabla (Ax^m) = mAx^{m-1}, \quad \nabla^2 (Ax^m) = m(m-1)Ax^{m-2}.
\]
We say a tensor $A \in \mathbb{S}^{[m,n]}$ is positive definite if
\[
Ax^m > 0 \quad \text{for all} \quad x \in \mathbb{R}^n, x \neq 0.
\]
We let $\mathbb{S}^{[m,n]}_+$ denote the space of positive definite tensors in $\mathbb{S}^{[m,n]}$.

We use the symbol $\oplus$ to mean symmetrized outer product, i.e.,
\[
a \oplus b = ab^T + ba^T.
\]

3. Problem reformulation. Let $\Sigma$ denote the unit sphere, i.e.,
\[
\Sigma = \{ x \in \mathbb{R}^n \mid \|x\|^m = 1 \}.
\]
Let $A \in \mathbb{S}^{[m,n]}$ and $B \in \mathbb{S}^{[m,n]}_+$. Then we may define the nonlinear program
\[
\max f(x) = \frac{Ax^m}{Bx^m} \|x\|^m \quad \text{subject to} \quad x \in \Sigma. \tag{3.1}
\]
The constraint makes the term $\|x\|^m$ in $f(x)$ superfluous; nevertheless, we retain this form since choosing $B = \mathcal{E}$ yields $f(x) = Ax^m$, as in [3].

The details of computing the derivatives are provided in Appendix A. Here we simply state the results as a theorem.

**Theorem 3.1.** Let $f(x)$ be as defined in (3.1). For $x \in \Sigma$, the gradient is
\[
g(x) \equiv \nabla f(x) = \frac{m}{Bx^m} \left[ (Ax^m) x + Ax^{m-1} - \left( \frac{Ax^m}{Bx^m} \right) Bx^{m-1} \right]. \tag{3.2}
\]
For $x \in \Sigma$, the Hessian is
\[
H(x) \equiv \nabla^2 f(x) = \frac{m^2 Bx^m}{(Bx^m)^2} \left( Bx^{m-1} \oplus Bx^{m-1} \right) \\
+ \frac{m}{Bx^m} \left[ (m-1)Ax^{m-2} + Ax^m (I + (m-2)xx^T) + m(Ax^{m-1} \oplus x) \right] \\
- \frac{m}{(Bx^m)^2} \left[ (m-1)Ax^m Bx^{m-2} + m(Ax^{m-1} \oplus Bx^{m-1}) \right] \\
+ mAx^m (x \oplus Bx^{m-1}). \tag{3.3}
\]

These complicated derivatives reduce for $B = \mathcal{E}$. In that case, we have $Bx^m = 1$ and $Bx^{m-1} = x$, so these equations become.
\[
g(x) = mAx^{m-1}, \quad H(x) = m(m-1)Ax^{m-2}.
\]
These match the derivatives of $f(x) = Ax^m$, as proved in [3]. Note that we have used the fact that $(m-1)\mathcal{E}x^{m-2} = I + (m-2)xx^T$ for all $x \in \Sigma$. 

We are considering the nonlinear program in (3.1) because there is a correspondence between it and the generalized tensor eigenvalue problem in (1.1).

**Theorem 3.2.** Any solution \((\lambda, x)\) to (1.1) must be a KKT point of (3.1) with \(\lambda\) as the Lagrange multiplier.

**Proof.** Let constraint \(x \in \Sigma\) be expressed as \(|x|^m = 1\). Then the Lagrangian is

\[
\mathcal{L}(x, \lambda) = f(x) - \lambda(|x|^m - 1).
\]

Hence, using the derivatives in Appendix A, we have

\[
\nabla_x \mathcal{L}(x, \lambda) = \frac{m}{\mathcal{B}x^m} \left( (A x^m) x + A x^{m-1} - \left( \frac{A x^m}{B x^m} \right) B x^{m-1} \right) - m \lambda x = 0.
\]

So, \(x\) is a KKT point of (3.1) with Lagrange multiplier \(\lambda\) as defined in (1.2). \(\square\)

From the previous theorem, there is an equivalence between generalized tensor eigenpairs and KKT points of (3.1). Hence, solving (3.1) yields eigenpairs. An eigenpair may correspond to a local maximum, a local minimum, or a saddle point. For a given eigenpair \((\lambda, x)\) normalized so that \(x \in \Sigma\), we can categorize it by considering the projected Hessian of the Lagrangian, i.e.,

\[
C(\lambda, x) = U^T (H(x) - \lambda m I) U \in \mathbb{R}^{(n-1) \times (n-1)},
\]

where \(U \in \mathbb{R}^{n \times (n-1)}\) is an orthonormal basis for \(x^+\). We can then say the following:

- \(C(\lambda, x)\) positive definite \(\Rightarrow\) local minimum of (3.1),
- \(C(\lambda, x)\) negative definite \(\Rightarrow\) local maximum of (3.1),
- \(C(\lambda, x)\) indefinite \(\Rightarrow\) saddle point of (3.1).

The argument is very similar to that presented in [3] and so is omitted. Optimization approaches cannot easily find saddle points, but they can find local minima and maxima. We describe such an optimization approach in the next section.

**4. Derivation of GEAP algorithm.** We propose to use a property of convex functions of the sphere to develop a monotonically convergent method. We consider an idea originally from [2, 9]; see also [3] for a proof. We have modified the theorem here to focus on its local applicability by considering just an open neighborhood of \(w\) rather than all of \(\mathbb{R}^n\).

**Theorem 4.1 (Koldis and Regalia [2, 9]).** Let \(f(x)\) be a given function, and let \(w \in \Sigma\) such that \(\nabla f(w) \neq 0\). Let \(\Omega\) be an open neighborhood of \(w\), and assume \(f\) is convex and continuously differentiable on \(\Omega\). Define \(v = \nabla f(w)/||\nabla f(w)||\). If \(v \in \Omega\) and \(v \neq w\), then \(f(v) - f(w) > 0\).

**Corollary 4.2.** Let \(f(x)\) be a given function, and let \(w \in \Sigma\) such that \(\nabla f(w) \neq 0\). Let \(\Omega\) be an open neighborhood of \(w\), and assume \(f\) is concave and continuously differentiable on \(\Omega\). Define \(v = -\nabla f(w)/||\nabla f(w)||\). If \(v \in \Omega\) and \(v \neq w\), then \(f(v) - f(w) < 0\).

Hence, if \(f\) is locally convex, then a simple algorithm, i.e.,

\[
x_{\text{new}} = g(x)/||g(x)||.
\]

will yield ascent. Conversely, if \(f\) is locally concave, we can expect descent from \(x_{\text{new}} = -g(x)/||g(x)||\). Unfortunately, the function \(f(x)\) in (3.1) may not be convex or concave.
To fix this, we work with a shifted function,
\[ \tilde{f}(x) = f(x) + \alpha \|x\|^m. \]  
From [3], we have that for \( x \in \Sigma \),
\[ \tilde{g}(x) \equiv \nabla \tilde{f}(x) = g(x) + \alpha mx, \]  
\[ \hat{H}(x) \equiv \nabla^2 \tilde{f}(x) = H(x) + \alpha mI + \alpha m(m-2)xx^T. \]

If we choose \( \alpha \) appropriately, then we can ensure that \( \hat{H}(x) \) is positive or negative definite, ensuring that \( \tilde{f}(x) \) is locally convex or concave. In [3] for the special case of \( \mathcal{B} = \mathcal{E} \), we proposed choosing a single value for \( \alpha \) in SS-HOPM that ensured convexity on the entire sphere. But it is difficult to choose a reasonable value in advance, and poor choices lead to either very slow convergence or a complete lack of convergence. In this work, we propose to choose \( \alpha \) adaptively.

For an arbitrary matrix \( n \times n \) symmetric matrix \( M \), the following notation denotes its eigenvalues: \( \lambda_{\min}(M) = \lambda_1(M) \leq \lambda_2(M) \leq \cdots \leq \lambda_n(M) = \lambda_{\max}(M) \).

**Theorem 4.3.** Assume \( x \in \Sigma \). Let \( H(x) \) and \( \hat{H}(x) \) be defined as in (3.3) and (4.3), respectively. For \( \alpha \geq 0 \), the eigenvalues of \( \hat{H} \) are bounded as
\[ \lambda_i(H) + ma \leq \lambda_i(\hat{H}) \leq \lambda_i(H) + m\alpha + m(m-2)\alpha \] for \( i = 1, \ldots, n \). Likewise, for \( \alpha \leq 0 \), the eigenvalues of \( \hat{H} \) are bounded as
\[ \lambda_i(H) + m\alpha + m(m-2)\alpha \leq \lambda_i(\hat{H}) \leq \lambda_i(H) + ma \]
for \( i = 1, \ldots, n \).

**Proof.** The proof follows immediately from Weyl’s inequality. \( \square \)

In the convex case, our goal is to choose \( \alpha \) so that \( \hat{H} \) is positive semi-definite in a local neighborhood of the current iterate, \( x \). By the smoothness of \( f(x) \) when \( x \) is away from zero, we may argue that for every \( \tau > 0 \), there exists and \( \delta > 0 \) such that \( \hat{H} \) is positive semi-definite for all \( \|x - x_k\| \leq \delta \) whenever \( \lambda_{\min}(\hat{H}) \geq \tau \). In other words, \( \tau \) is the threshold for positive definiteness.

**Corollary 4.4.** Assume \( x \in \Sigma \). Let \( \tau > 0 \). If
\[ \alpha = \max \{ 0, (\tau - \lambda_{\min}(H))/m \}, \] then \( \lambda_{\min}(\hat{H}) \geq \tau \).

In the concave case, our goal is to choose \( \alpha \) so that \( \hat{H} \) is negative semi-definite in a local neighborhood of the current iterate.

**Corollary 4.5.** Assume \( x \in \Sigma \). Let \( \tau > 0 \). If
\[ \alpha = \min \{ 0, -(\tau + \lambda_{\max}(H))/m \} = -\max \{ 0, \tau - \lambda_{\min}(-H)/m \}, \] then \( \lambda_{\max}(\hat{H}) \leq -\tau \).

From Theorem 4.1, if \( \alpha \) is selected to make \( \tilde{f}(x) \) locally convex, we have
\[ x_+ = \tilde{g}(x)/\|\tilde{g}(x)\| \Rightarrow \tilde{f}(x_+) > \tilde{f}(x) \Rightarrow f(x_+) > f(x) \]
so long as \( x_+ \in \Omega \), the convex neighborhood of \( x \). Even though we are adaptively changing \( \alpha \), we see increase in the original function at each step. A similar argument
Algorithm 1 Generalized Eigenpair Adaptive Power (GEAP) Method

Given tensors $A \in S^{[m,n]}$ and $B \in S_+^{[m,n]}$ and an initial guess $\hat{x}_0$. Let $\beta = 1$ if we want to find local maxima (and the function is convex); otherwise, let $\beta = -1$, indicating that we are seeking local minima (and the function is concave). Let $\tau$ be the tolerance on being positive/negative definite.

1: $x_0 \leftarrow \hat{x}_0 / \|\hat{x}_0\|
2: \text{for } k = 0, 1, \ldots \text{ do}
3: \quad \lambda_k \leftarrow A x_k^m / B x_k^m
4: \quad H_k \leftarrow H(x_k)
5: \quad \alpha_k \leftarrow \beta \max\{0, (\tau - \lambda_{\min}(\beta H_k))/m\}
6: \quad \hat{x}_{k+1} \leftarrow \beta(A x_k^{m-1} - \lambda_k B x_k^{m-1} + (\alpha_k + \lambda_k) B x_k^m x_k)
7: \quad x_{k+1} = \hat{x}_{k+1} / \|\hat{x}_{k+1}\|
8: \text{end for}

The potential problem with this approach is that it may be the case that $x_+ \not\in \Omega$. If that happens, we may observe that the function values (i.e., $\lambda_k$) are not increasing (or decreasing) monotonically as expected. To fix this, we make a more conservative choice for $\tau$ (at least temporarily), which will in turn enforce a more conservative choice of $\alpha$. Thus far in our experiments, such contingencies have not been necessary, so we have not included the details in the algorithm.

In Algorithm 2, we show the specialization of the method to the Z-eigenvalue problem. This is the same as SS-HOPM, except for the adaptive shift. Note that unlike Algorithm 1, this algorithm can be used even when $m$ is odd.

Algorithm 2 Z-Eigenpair Adaptive Power Method

Given tensor $A \in S^{[m,n]}$ and an initial guess $\hat{x}_0$. Let $\beta = 1$ if we want to find local maxima (and the function is convex); otherwise, let $\beta = -1$, indicating that we are seeking local minima (and the function is concave). Let $\tau$ be the tolerance on being positive/negative definite.

1: $x_0 \leftarrow \hat{x}_0 / \|\hat{x}_0\|
2: \text{for } k = 0, 1, \ldots \text{ do}
3: \quad \lambda_k \leftarrow A x_k^m
4: \quad H_k \leftarrow m(m-1)A x_k^{m-2}
5: \quad \alpha_k \leftarrow \beta \max\{0, (\tau - \lambda_{\min}(\beta H_k))/m\}
6: \quad \hat{x}_{k+1} \leftarrow \beta(A x_k^{m-1} + \alpha_k x_k)
7: \quad x_{k+1} = \hat{x}_{k+1} / \|\hat{x}_{k+1}\|
8: \text{end for}

5. Numerical experiments. In all numerical experiments, we used the following settings. We set $\tau = 10^{-6}$, where $\tau$ is the tolerance on being positive or negative definite. We consider the iterates to be converged once $|\lambda_{k+1} - \lambda_k| \leq 10^{-15}$.

5.1. Comparison to SS-HOPM. The following example is originally from [2] and was used in evaluating the SS-HOPM algorithm in [3]. Our goal is to compute the Z-eigenpairs and show that the adaptive method is faster than using a fixed value.
Adaptive Shifted Power Method for Computing Generalized Tensor Eigenpairs

for the shift.

**Example 5.1 (Kofidis and Regalia Z-eigenpairs [2, Example 1]).** Our goal is to compute the Z-eigenpairs. Let $\mathbf{A} \in \mathbb{S}^{[4,3]}$ be the symmetric tensor defined by

$$
\begin{align*}
& a_{1111} = 0.2883, \quad a_{1112} = -0.0031, \quad a_{1113} = 0.1973, \quad a_{1122} = -0.2485, \\
& a_{1123} = -0.2939, \quad a_{1133} = 0.3847, \quad a_{1222} = 0.2972, \quad a_{1223} = 0.1862, \\
& a_{1233} = 0.0919, \quad a_{1333} = -0.3619, \quad a_{2222} = 0.1241, \quad a_{2223} = -0.3420, \\
& a_{2233} = 0.2127, \quad a_{2333} = 0.2727, \quad a_{3333} = -0.3054.
\end{align*}
$$

Let $\mathbf{B} = \mathbf{E}$. A complete list of the 11 Z-eigenpairs is provided in Table B.1; there are three maxima and three minima. □

Table 5.1 shows the eigenvalues computed by Algorithm 2 for Example 5.1. There are six different experiments, varying $\beta \in \{-1, 1\}$ and $\alpha$. We used 100 random starting guesses, each entry selected uniformly at random from the interval $[-1, 1]$; the same set of random starts was used for each set of experiments. The table lists the number of occurrences of each eigenpair in the 100 experiments, the median number of iterations until convergence, and the number of runs that violated monotonicity, including the maximum violation.

The first three experiments use $\beta = 1$ to look for local maxima. The first experiment varies $\alpha$, the second uses $\alpha = 2$ (as in [3]), and the third uses $\alpha = 10$. All three variations find all three local maxima. The results for $\alpha = 2$ and the adaptive method are nearly identical — they find the same local maxima with the same frequency. The difference is that $\alpha = 2$ uses more iterations than the adaptive shift. Choosing $\alpha = 10$ is similar, except now five of the runs do not converge within the allotted 500 iterations. There was no breakdown in monotonicity, and these runs would converge eventually. If the shift is too small (e.g., $\alpha = 0$), then some or all of the runs may fail to converge [3].

The last three experiments use $\beta = -1$ to find local minima. Again, we vary $\alpha$ using an adaptive choice along with $\alpha = -2$ and $\alpha = -10$. The adaptive method requires the fewest number of iterations. Each experiments finds all three local minima with the exact same frequencies.

Figure 5.1 shows sample results for one run for computing Z-eigenpairs of $\mathbf{A}$ from Example 5.1. The left hand plot shows the selected shift values at each iteration. The right hand plot shows the convergence of the eigenvalue. The adaptive shift is the fastest to converge.

**5.2. Numerical results for D-eigenpairs.** Next we consider a different type of tensor eigenpair that also conforms to the generalized tensor eigenpair framework.

**Example 5.2 (D-eigenpairs [8]).** Qi, Wang, and Wu [8] propose D-eigenpairs for diffusion kurtosis imaging (DKI). We consider this example here since it can be expressed as a generalized tensor eigenproblem.$^1$ The DKI tensor $\mathbf{A} \in \mathbb{S}^{[4,3]}$ (called

$^1$Note that only four digits of precision for $\mathbf{A}$ and $\mathbf{D}$ are provided in [8]. We were unable to validate the solutions provided in the original paper. It is not clear if this is to a lack of precision or a typo in paper. Here, the problem is rescaled as well: $\mathbf{D}$ is multiplied by $10^2$, $\lambda$ is divided by $10^4$, and $\mathbf{x}$ is divided by 10.
Table 5.1: Eigenpairs computed by GEAP for $\mathcal{A} \in \mathbb{S}^{[2,3]}$ from Example 5.1 with 100 random starts. We report the number of times we converge to each eigenpair, the median number of iterations, and the maximum number of monotonicity violations (along with the maximum violation, if applicable).

(a) $\alpha$ adaptive, $\beta = 1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 53    | 0.8893    | 0.6672 0.2471 -0.7027 | 30 | 0 |
| 20    | 0.8169    | 0.8412 -0.2635 0.4722 | 34 | 0 |
| 18    | 0.3633    | 0.2676 0.6447 0.7160 | 26 | 0 |

(b) $\alpha = 2$, $\beta = 1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 53    | 0.8893    | 0.6672 0.2471 -0.7027 | 49 | 1 (10^-11) |
| 20    | 0.8169    | 0.8412 -0.2635 0.4722 | 45 | 0 |
| 18    | 0.3633    | 0.2676 0.6447 0.7160 | 57 | 0 |

(c) $\alpha = 10$, $\beta = 1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 48    | 0.8893    | 0.6672 0.2471 -0.7027 | 192 | 0 |
| 29    | 0.8169    | 0.8412 -0.2635 0.4722 | 185 | 0 |
| 18    | 0.3633    | 0.2676 0.6447 0.7160 | 261 | 0 |
| 5     | — Failed to converge after 500 iterations — | |

(d) $\alpha$ adaptive, $\beta = -1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 22    | -0.0451   | 0.7797 0.6135 0.1250 | 18 | 0 |
| 37    | -0.5629   | 0.1762 -0.1796 0.9678 | 17 | 0 |
| 41    | -1.0954   | 0.5915 -0.7467 -0.3043 | 17 | 0 |

(e) $\alpha = -2$, $\beta = -1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 22    | -0.0451   | 0.7797 0.6135 0.1250 | 34 | 0 |
| 37    | -0.5629   | 0.1762 -0.1796 0.9678 | 20 | 0 |
| 41    | -1.0954   | 0.5915 -0.7467 -0.3043 | 21 | 1 (10^-11) |

(f) $\alpha = -10$, $\beta = -1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|-----------|-----|------|------|
| 22    | -0.0451   | 0.7797 0.6135 0.1250 | 186 | 0 |
| 37    | -0.5629   | 0.1762 -0.1796 0.9678 | 103 | 0 |
| 41    | -1.0954   | 0.5915 -0.7467 -0.3043 | 94 | 0 |

$\mathcal{W}$ in the original paper) is the symmetric tensor defined by the unique elements:

\[
\begin{align*}
    a_{1111} &= 0.4982, & a_{1112} &= -0.0582, & a_{1113} &= -1.1719, & a_{1122} &= 0.2236, \\
    a_{1123} &= -0.0171, & a_{1133} &= 0.4597, & a_{1222} &= 0.4880, & a_{1223} &= 0.1852, \\
    a_{1233} &= -0.4087, & a_{1333} &= 0.7639, & a_{2222} &= 0.0000, & a_{2223} &= -0.6162, \\
    a_{2233} &= 0.1519, & a_{2333} &= 0.7631, & a_{3333} &= 2.6311.
\end{align*}
\]
The tensor $\mathcal{B}$ is the symmetrized outer product of the matrix $\mathbf{D}$ with itself where

$$
\mathbf{D} = \begin{bmatrix}
1.755 & 0.035 & 0.132 \\
0.035 & 1.390 & 0.017 \\
0.132 & 0.017 & 4.006
\end{bmatrix},
$$

so $\mathcal{B}$ is the tensor whose unique elements are given by

- $b_{1111} = 3.0800$, $b_{1112} = 0.0614$, $b_{1113} = 0.2317$, $b_{1122} = 0.8140$,
- $b_{1123} = 0.0130$, $b_{1133} = 2.3571$, $b_{1222} = 0.0468$, $b_{1223} = 0.0616$,
- $b_{1233} = 0.0482$, $b_{1333} = 0.5288$, $b_{2222} = 1.9321$, $b_{2223} = 0.0236$,
- $b_{2233} = 1.8563$, $b_{2333} = 0.0681$, $b_{3333} = 16.0480$.

There are a total of 13 distinct real-valued D-eigenpairs, computed by solving the polynomial equations using Mathematica and listed in Table B.2; there are four maxima and three minima. \(\square\)

Table 5.2 shows the eigenpairs calculated by Algorithm 1. With 100 random starts, we find the four local maxima with $\beta = 1$. Likewise, with 100 random starts, we find the three local minima with $\beta = -1$. There are no violations to monotonicity.

**5.3. Generalized eigenpairs for randomly generated $\mathcal{A}$ and $\mathcal{B}$.** Here we consider a randomly generated problem. In order to create such a problem, we need a method to generate a positive definite $\mathcal{B}$. We use the notation $\mathcal{B} = (\mathcal{E}, \mathbf{S}, \ldots, \mathbf{S}) \in \mathbb{S}^{[m,n]}$ to denote tensor-matrix multiplication in which the tensor $\mathcal{E}$ is multiplied by a matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ in every mode, i.e.,

$$
b_{i_1 \cdots i_m} = \sum_{j_1=1}^{n} \cdots \sum_{j_m=1}^{n} e_{j_1 \cdots j_m} s_{i_1 j_1} \cdots s_{i_m j_m}.
$$

**Theorem 5.3.** Let $\mathbf{S} \in \mathbb{R}^{n \times n}$ be symmetric. For $m$ even, define $\mathcal{B} \in \mathbb{S}^{[m,n]}$ as $\mathcal{B} = (\mathcal{E}, \mathbf{S}, \ldots, \mathbf{S})$. If $(\mu, \mathbf{x})$ is a real-valued eigenpair of $\mathbf{S}$ and $\|\mathbf{x}\| = 1$, then $(\lambda, \mathbf{x})$ is a $Z$-eigenpair of $\mathcal{B}$ with $\lambda = \mu^n$. 
Table 5.2: Eigenpairs computed by GEAP for $\mathcal{A} \in S^{[2,3]}$ and $D \in S^{[2,3]}$ from Example 5.2 with 100 random starts. We report the number of times we converge to each eigenpair, the median number of iterations, and the maximum number of monotonicity violations (along with the maximum violation, if applicable).

(a) $\beta = 1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|----------|-----|------|-------|
| 39    | 0.5356   | 0.9227 -0.1560 -0.3526 | 39  | 0     |
| 19    | 0.4359   | 0.5388 0.8342 -0.1179 | 48  | 0     |
| 25    | 0.2514   | 0.3564 -0.8002 0.4823 | 67  | 0     |
| 25    | 0.2219   | 0.2184 0.3463 0.9124 | 34  | 0     |

(b) $\beta = -1$

| # occ. | $\lambda$ | $x$ | its. | viol. |
|-------|----------|-----|------|-------|
| 39    | -0.0074  | 0.3669 0.5346 -0.7613 | 13  | 0     |
| 37    | -0.1242  | 0.9439 0.1022 0.3141 | 51  | 0     |
| 24    | -0.3313  | 0.2810 -0.9420 -0.1837 | 27  | 0     |

Proof. Let $(\mu, x)$ be an eigenpair of $\mathcal{S}$ such that $\|x\| = 1$. Then, noting that $\mathcal{S}^T = \mathcal{S}$, we have

$$\mathcal{B}x^{m-1} = SE(Sx)^{m-1} = SE(\mu x)^{m-1} = \mu^{m-1}SE^{m-1}x = x^{m}.$$ 

Example 5.4 (Random). To generate a random symmetric tensor $\mathcal{A} \in S^{[6,4]}$, we select random entries from $[-1, 1]$ and symmetrize the result. The tensor entries are listed in Figure C.1, rounded to 4 decimal places.

To generate a random positive definite symmetric tensor $\mathcal{B} \in S^{[m,n]}_+ = S_{[m,n]}^{[6,4]}$, we use Theorem 5.3. (Note that this approach samples a convenient subset of $S^{[m,n]}_+$ and does not draw from the entire space.) We compute a matrix $\mathcal{S} = UDU^T \in \mathbb{R}^{4 \times 4}$, where $U \in \mathbb{R}^{4 \times 4}$ is a random orthonormal matrix and $D \in \mathbb{R}^{4 \times 4}$ is a diagonal matrix with entries selected uniformly at random from $[-\gamma, \gamma] \cup \gamma, 1]$ with $\gamma = \sqrt{0.1} = \sqrt{0.1}$. We let $\mathcal{B} = (E, S, \ldots, S)$, so that $\mathcal{S}$ has all its Z-eigenvalues in $[0.1, 1]$ and is positive definite. In this case, the randomly selected diagonal for $D$ is $\{-0.8620, 0.8419, 0.7979, 0.6948\}$. The random $\mathcal{B}$ is given in Figure C.2 and has also been rounded to 4 decimal places. Its minimum Z-eigenvalue (computed by GEAP) is 0.1125 = 0.69486, as expected.

There are a total of 26 real-valued $\mathcal{B}$-eigenpairs of $\mathcal{A}$, listed in Table B.3. There are three maxima and four minima.

Table 5.3 shows the generalized eigenpairs calculated by Algorithm 1. With 1000 random starts, we find the three local maxima with $\beta = 1$. Likewise, with 1000 random starts, we find the four local minima with $\beta = -1$. There are only small violations to monotonicity.

6. Related work. Like its predecessor SS-HOPM [3], the GEAP method has the desirable qualities of guaranteed convergence and simple implementation. Additionally, the adaptive choice of $\alpha$ in GEAP (as opposed to SS-HOPM) means that there are no parameters for the user to specify.

Also like SS-HOPM, the GEAP method can only converge to local maxima and minima of (3.1) and so will miss any saddle point solutions. Nevertheless, the largest
Table 5.3: Eigenpairs computed by GEAP for $A \in S^{[6,4]}$ and $B \in S^{[6,4]}_+$ from Example 5.4 with 1000 random starts. We report the number of times we converge to each eigenpair, the median number of iterations, and the maximum number of monotonicity violations (along with the maximum violation, if applicable).

(a) $\beta = 1$

| # occ. | $\lambda$ | $\mathbf{x}$ | its. | viol. |
|--------|------------|--------------|------|-------|
| 683    | 11.3476    | [0.4064 0.2313 0.8840 0.0716] | 59   | 420 ($10^{-14}$) |
| 128    | 3.7394     | [0.2185 −0.3142 0.2197 −0.2613] | 149  | 11 ($10^{-16}$) |
| 189    | 2.9979     | [0.8224 0.4084 −0.0174 −0.3986] | 23   | 9 ($10^{-15}$) |

(b) $\beta = −1$

| # occ. | $\lambda$ | $\mathbf{x}$ | its. | viol. |
|--------|------------|--------------|------|-------|
| 151    | −1.1507    | [0.1935 0.5444 0.2991 −0.7594] | 88   | 0     |
| 226    | −3.2777    | [0.6888 −0.6272 −0.2914 −0.2174] | 33   | 14 ($10^{-13}$) |
| 140    | −3.6986    | [0.7999 0.4554 0.2814 0.2991] | 22   | 21 ($10^{-13}$) |
| 483    | −6.3985    | [0.0783 0.1345 0.3877 0.9060] | 82   | 73 ($10^{-13}$) |

An alternative to GEAP is to solve (1.1) or (3.1) using a numerical nonlinear, homotopy, or optimization approach. The advantage of GEAP is that it guarantees decrease at each iteration without any globalization techniques (like line search or trust region) and is generally as cheap or cheaper per iteration than any competing numerical method. The disadvantage is that the rate of convergence of GEAP is only linear, as opposed to quadratic for, say, Newton’s method.

Since (1.1) is a polynomial system of equations, we can also consider a polynomial solver approach. This does not scale to larger problems and may be slow even for small problems. Nevertheless, it finds all solutions (even saddle points). We have used the Gröbner basis polynomial solver NSolve in Mathematica to compute the full set of solutions for the problems discussed in this paper.

In terms of methods specifically geared to tensor eigenvalues, most work has focused on computing the largest H-eigenvalue for a nonnegative tensor: [6, 5]. The method of Liu, Zhou, and Ibrahim [5] is guaranteed to always find the largest eigenvalue and also uses a “shift” approach.

7. Conclusions. The paper has proposed two improvements to the SS-HOPM method [3]. First, we have adapted the method to the generalized tensor eigenproblem. Second, we have proposed a method for adaptively and automatically selecting the shift, overcoming a major problem with the SS-HOPM method because choosing the shift too large dramatically slows convergence whereas choosing it too small can cause the method to fail completely.

We have tested our method numerically on several problems from the literature, including computing of Z-, H-, and D-eigenpairs. We have also proposed a novel method for generating random symmetric positive definite tensors.

Appendix A. Useful derivatives. First, we consider the gradient and Hessian
of the general function
\[ f(x) = \frac{f_1(x)f_2(x)}{f_3(x)}. \]

Let \( g_i(x) \) denote \( \nabla f_i(x) \). From matrix calculus, the gradient of \( f(x) \) is
\[ \mathbf{g} = \left( f_1 g_2 + f_2 g_1 - (f_1 f_2 / f_3) g_3 \right) / f_3. \]

Here we have dropped the argument, \( x \), to simply the notation. Let \( H_i(x) \) denote \( \nabla^2 f_i(x) \). The Hessian of \( f(x) \) is
\[ \mathbf{H} = \frac{f_1 f_2}{f_3^2} (\mathbf{g}_3 \otimes \mathbf{g}_3) + \frac{1}{f_3^2} \left[ f_2 \mathbf{H}_1 + f_1 \mathbf{H}_2 + (\mathbf{g}_1 \otimes \mathbf{g}_2) \right] - \frac{1}{f_3^2} \left[ f_1 f_2 \mathbf{H}_3 + f_2 (\mathbf{g}_1 \otimes \mathbf{g}_3) + f_1 (\mathbf{g}_2 \otimes \mathbf{g}_3) \right]. \]

Now we specialize \( f(x) \) to (3.1): let \( f_1 = A x^m \), \( f_2 = \|x\|^m \), and \( f_3 = B x^m \). The following derivatives are proved in [3]:
\[ g_1 = m A x^{m-1}, \quad H_1 = m(m-1) A x^{m-2}, \]
\[ g_2 = m \|x\|^{m-2} x, \quad H_2 = m \|x\|^{m-2} I + m(m-2) \|x\|^{m-4} xx^T, \]
\[ g_3 = m B x^{m-1}, \quad H_3 = m(m-1) B x^{m-2}. \]

We need only consider the case for \( x \in \Sigma \), so we may assume
\[ f_2 = 1, \quad g_2 = m x, \quad H_2 = m (I + (m-2) xx^T). \]

Putting everything together, we have for \( x \in \Sigma \),
\[ g(x) = \frac{m}{B x^m} \left[ (A x^m) x + A x^{m-1} - \left( \frac{A x^m}{B x^m} \right) B x^{m-1} \right]. \]

For the Hessian, assuming \( x \in \Sigma \), we have
\[ H(x) = \frac{m^2 A x^m}{(B x^m)^2} (B x^{m-1} \otimes B x^{m-1}) + \frac{m}{B x^m} \left[ (m-1) A x^{m-2} + A x^m (I + (m-2) xx^T) + m (A x^{m-1} \otimes x) \right] - \frac{m}{(B x^m)^2} \left[ (m-1) A x^m B x^{m-2} + m (A x^{m-1} \otimes B x^{m-1}) + m A x^m (x \otimes B x^{m-1}) \right]. \]

**Appendix B. Complete lists of real eigenpairs.** A polynomial system solver (NSolve) using a Gröbner basis method is available in Mathematica and has been employed to generate a complete list of eigenpairs for the examples in this paper in Tables B.1–B.3.

**Appendix C. Randomly generated tensors for Example 5.4.** The randomly generated tensors used for Example 5.4 are specified in Figure C.1 and Figure C.2.
| Table B.1: All Z-eigenpairs for $\mathcal{A} \in \mathbb{S}^{[4,3]}$ from Example 5.1 |
|------------------|------------------|------------------|------------------|
| $\lambda$       | $x^l$            | $C(\lambda, x)$ | type             |
| $-1.0954$        | $0.5915 -0.7467 -0.3043$ | 1.86, 2.75       | Maxima           |
| $-0.5629$        | $0.1762 -0.1796 0.9678$  | 1.63, 2.38       | Maxima           |
| $-0.0451$        | $0.7797 0.6135 0.1250$   | 0.82, 1.25       | Maxima           |
| $0.1735$         | $0.3357 0.9073 0.2531$   | -1.10, 0.86      | Saddle           |
| $0.2433$         | $0.9895 0.0947 0.1088$   | -1.19, -1.46     | Saddle           |
| $0.2625$         | $0.3138 -0.4425 -0.8870$ | 0.62, -2.17      | Saddle           |
| $0.2682$         | $0.6099 0.4362 0.6616$   | -1.18, 0.79      | Saddle           |
| $0.3633$         | $0.2676 0.6447 0.7160$   | -1.18, -0.57     | Minima           |
| $0.5105$         | $0.3598 -0.7780 0.5150$   | 0.59, -2.34      | Saddle           |
| $0.8169$         | $0.8412 -0.2635 0.4722$   | -2.26, -0.90     | Minima           |
| $0.8893$         | $0.6672 0.2471 -0.7027$   | -1.85, -0.89     | Minima           |

| Table B.2: All D-eigenpairs for $\mathcal{A} \in \mathbb{S}^{[4,3]}$ and $\mathcal{D} \in \mathbb{S}^{[2,3]}$ from Example 5.2 |
|------------------|------------------|------------------|------------------|
| $\lambda$       | $x^l$            | $C(\lambda, x)$ | type             |
| $-0.3313$        | $0.2389 -0.7741 -0.1509$ | 1.02, 2.11       | Minima           |
| $-0.1242$        | $0.6577 0.0712 0.2189$   | 0.35, 1.25       | Minima           |
| $-0.0074$        | $0.2161 0.3149 -0.4485$   | 0.36, 0.46       | Minima           |
| $0.0611$         | $0.6113 -0.4573 0.1181$   | -0.63, -1.14     | Saddle           |
| $0.1039$         | $0.3314 0.5239 0.3084$   | -0.46, 0.63      | Saddle           |
| $0.2099$         | $0.2440 -0.1250 0.4601$   | -0.32, 0.07      | Saddle           |
| $0.2056$         | $0.1211 -0.2367 -0.4766$   | -0.29, 0.13      | Saddle           |
| $0.2219$         | $0.1143 0.1812 0.4773$   | -0.08, -0.20     | Maxima           |
| $0.2431$         | $0.0943 -0.6840 0.2985$   | 0.18, -1.11      | Saddle           |
| $0.3187$         | $0.2485 -0.5529 0.3363$   | -0.14, -0.71     | Minima           |
| $0.3827$         | $0.6236 0.3954 -0.1678$   | -1.58, 0.32      | Saddle           |
| $0.4359$         | $0.4336 0.6714 -0.0949$   | -0.43, -1.64     | Maxima           |
| $0.5356$         | $0.6638 -0.1123 -0.2537$   | -0.48, -1.43     | Maxima           |

REFERENCES

[1] K. C. Chang, K. Pearson, and T. Zhang, *On eigenvalue problems of real symmetric tensors*, Journal of Mathematical Analysis and Applications, 359 (2009), pp. 416–422.

[2] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order symmetric tensors*, SIAM Journal on Matrix Analysis and Applications, 23 (2002), pp. 863–884.

[3] T. G. Kolda and J. R. Mayo, *Shifted power method for computing tensor eigenpairs*, SIAM Journal on Matrix Analysis and Applications, 32 (2011), pp. 1095–1124.

[4] L.-H. Lim, *Singular values and eigenvalues of tensors: A variational approach*, in CAMSAP’05: Proceeding of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005, pp. 120–132.

[5] Y. Liu, G. Zhou, and N. F. Ibrahim, *An always convergent algorithm for the largest eigenvalue of an irreducible nonnegative tensor*, Journal of Computational and Applied Mathematics, 235 (2010), pp. 286–292.

[6] M. Ng, L. Qi, and G. Zhou, *Finding the largest eigenvalue of a nonnegative tensor*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 1090–1099.

[7] L. Qi, *Eigenvalues of a real supersymmetric tensor*, Journal of Symbolic Computation, 40 (2005), pp. 1302–1324.

[8] L. Qi, Y. Wang, and E. X. Wu, *D-eigenvalues of diffusion kurtosis tensors*, Journal of Computational and Applied Mathematics, 221 (2008), pp. 150–157.

[9] P. A. Regalia and E. Kofidis, *Monotonic convergence of fixed-point algorithms for ICA*, IEEE Transactions on Neural Networks, 14 (2003), pp. 943–949.
Table B.3: All generalized tensor eigenpairs for $\mathcal{A} \in S^{[6,4]}$ and $\mathcal{B} \in S^{[6,4]}_+$ from Example 5.4

| $\lambda$ | $x^t$ | $\mathcal{C}(\lambda, x)$ evals. | type |
|-----------|-------|-------------------------------|------|
| $-6.3985$ | $[0.0733, 0.1345, 0.3877, 0.9990]$ | $20.43, 4.93, 11.20$ | Minima |
| $-3.5998$ | $[0.7899, 0.4554, 0.2814, 0.2991]$ | $8.05, 10.39, 12.41$ | Minima |
| $-3.2777$ | $[0.6888, -0.6272, -0.2914, -0.2174]$ | $8.27, 3.65, 5.95$ | Minima |
| $-1.7537$ | $[0.6329, -0.2966, -0.6812, -0.2180]$ | $-4.25, 3.00, 5.56$ | Saddle |
| $-1.1507$ | $[0.1935, 0.5444, 0.2991, -0.7594]$ | $0.73, 3.54, 4.20$ | Minima |
| $-1.0696$ | $[0.1372, 0.5068, 0.0665, -0.8485]$ | $-1.54, 3.30, 3.64$ | Saddle |
| $-1.0456$ | $[0.2365, 0.4798, -0.7212, 0.4402]$ | $-1.16, 1.54, 2.57$ | Saddle |
| $-0.7842$ | $[0.5409, 0.3388, 0.4698, 0.6099]$ | $16.02, 8.79, -12.47$ | Saddle |
| $-0.7457$ | $[0.6348, 0.5354, -0.4588, 0.3434]$ | $2.49, 0.94, -1.59$ | Saddle |
| $-0.2542$ | $[0.3900, -0.1333, 0.4946, -0.7652]$ | $-2.51, 2.99, 0.93$ | Saddle |
| $-0.2359$ | $[0.6956, -0.1369, 0.5550, -0.6964]$ | $6.38, 2.23, -1.27$ | Saddle |
| $0.0132$ | $[0.3064, 0.0541, 0.3111, -0.8980]$ | $-5.53, -2.36, 2.21$ | Saddle |
| $0.1633$ | $[0.4278, -0.6578, -0.2545, 0.5652]$ | $-2.42, 3.86, 2.30$ | Saddle |
| $0.3250$ | $[0.5265, 0.4653, 0.0927, 0.7055]$ | $7.50, -12.05, -3.41$ | Saddle |
| $0.5206$ | $[0.3738, -0.4806, -0.6066, 0.5111]$ | $3.19, -2.27, -1.47$ | Saddle |
| $0.5463$ | $[0.5157, -0.3055, -0.3313, -0.7287]$ | $-9.91, -3.67, 1.37$ | Saddle |
| $0.5945$ | $[0.4015, 0.8447, 0.1782, -0.3058]$ | $-3.70, 4.95, 1.87$ | Saddle |
| $0.6730$ | $[0.9634, -0.0099, 0.2396, -0.1204]$ | $-5.84, 7.88, 1.78$ | Saddle |
| $0.8862$ | $[0.3559, 0.8571, -0.1675, -0.3326]$ | $3.55, -2.24, -2.63$ | Saddle |
| $1.2902$ | $[0.9849, 0.0018, -0.1681, 0.0419]$ | $2.20, -5.97, -3.18$ | Saddle |
| $1.4646$ | $[0.7996, 0.4441, 0.4009, -0.3083]$ | $8.41, -2.08, -7.72$ | Saddle |
| $2.9979$ | $[0.8224, 0.4083, -0.0174, -0.3958]$ | $-4.00, -5.46, -6.56$ | Maxima |
| $3.5181$ | $[0.4549, -0.7574, 0.4502, -0.1469]$ | $-9.40, 1.89, -2.83$ | Saddle |
| $3.6087$ | $[0.0340, -0.8989, -0.0373, -0.4353]$ | $0.87, -8.03, -5.77$ | Saddle |
| $3.7914$ | $[0.2185, -0.9142, 0.2197, -0.2013]$ | $-8.72, -0.90, -3.44$ | Maxima |
| $11.3476$ | $[0.4064, 0.2313, 0.8810, 0.0716]$ | $-7.26, -18.98, -21.53$ | Maxima |
Fig. C.1: \( \mathbf{A} \) from Example 5.4

\[
\begin{array}{c|c|c|c|c}
\mathbf{a}_{111111} & = & 0.2888, & \mathbf{a}_{11112} & = & -0.0013, & \mathbf{a}_{111113} & = & -0.1422, & \mathbf{a}_{111114} & = & -0.0323, \\
\mathbf{a}_{111122} & = & -0.1079, & \mathbf{a}_{111123} & = & -0.0899, & \mathbf{a}_{111124} & = & -0.2487, & \mathbf{a}_{11113} & = & 0.0231, \\
\mathbf{a}_{111134} & = & -0.0106, & \mathbf{a}_{111144} & = & 0.0740, & \mathbf{a}_{111222} & = & 0.1490, & \mathbf{a}_{111223} & = & 0.0527, \\
\mathbf{a}_{111224} & = & -0.0710, & \mathbf{a}_{111233} & = & -0.1039, & \mathbf{a}_{111234} & = & -0.0250, & \mathbf{a}_{11124} & = & 0.0169, \\
\mathbf{a}_{111333} & = & 0.2208, & \mathbf{a}_{111334} & = & 0.0662, & \mathbf{a}_{111344} & = & 0.0046, & \mathbf{a}_{111444} & = & 0.0943, \\
\mathbf{a}_{112222} & = & -0.1144, & \mathbf{a}_{112223} & = & -0.1295, & \mathbf{a}_{112224} & = & -0.0484, & \mathbf{a}_{112233} & = & 0.0238, \\
\mathbf{a}_{112234} & = & -0.0237, & \mathbf{a}_{112244} & = & 0.0308, & \mathbf{a}_{112333} & = & 0.0142, & \mathbf{a}_{112334} & = & 0.0006, \\
\mathbf{a}_{112344} & = & -0.0044, & \mathbf{a}_{112444} & = & 0.0353, & \mathbf{a}_{113333} & = & 0.0947, & \mathbf{a}_{113334} & = & -0.0610, \\
\mathbf{a}_{113344} & = & -0.0293, & \mathbf{a}_{113444} & = & 0.0638, & \mathbf{a}_{114444} & = & 0.2326, & \mathbf{a}_{122222} & = & -0.2574, \\
\mathbf{a}_{122223} & = & 0.1018, & \mathbf{a}_{122224} & = & 0.0044, & \mathbf{a}_{122233} & = & 0.0248, & \mathbf{a}_{122234} & = & 0.0562, \\
\mathbf{a}_{122244} & = & 0.0221, & \mathbf{a}_{122333} & = & 0.0612, & \mathbf{a}_{122334} & = & 0.0184, & \mathbf{a}_{122344} & = & 0.0226, \\
\mathbf{a}_{122444} & = & 0.0247, & \mathbf{a}_{123333} & = & 0.0847, & \mathbf{a}_{123334} & = & -0.0209, & \mathbf{a}_{123344} & = & -0.0795, \\
\mathbf{a}_{123444} & = & -0.0323, & \mathbf{a}_{124444} & = & -0.0819, & \mathbf{a}_{133333} & = & 0.5486, & \mathbf{a}_{133334} & = & -0.0311, \\
\mathbf{a}_{133344} & = & -0.0592, & \mathbf{a}_{133444} & = & 0.0386, & \mathbf{a}_{134444} & = & -0.0138, & \mathbf{a}_{144444} & = & 0.0246, \\
\mathbf{a}_{222222} & = & 0.9207, & \mathbf{a}_{222223} & = & -0.0908, & \mathbf{a}_{222224} & = & 0.0633, & \mathbf{a}_{222233} & = & 0.1116, \\
\mathbf{a}_{222234} & = & -0.0318, & \mathbf{a}_{222244} & = & 0.1629, & \mathbf{a}_{222333} & = & 0.1797, & \mathbf{a}_{222334} & = & -0.0348, \\
\mathbf{a}_{222344} & = & -0.0058, & \mathbf{a}_{222444} & = & 0.1359, & \mathbf{a}_{223333} & = & 0.0584, & \mathbf{a}_{223334} & = & -0.0299, \\
\mathbf{a}_{223344} & = & -0.0110, & \mathbf{a}_{223444} & = & 0.1375, & \mathbf{a}_{224444} & = & -0.1405, & \mathbf{a}_{233333} & = & 0.3613, \\
\mathbf{a}_{233334} & = & 0.0809, & \mathbf{a}_{233344} & = & 0.0205, & \mathbf{a}_{233444} & = & 0.0196, & \mathbf{a}_{234444} & = & 0.0226, \\
\mathbf{a}_{244444} & = & -0.2487, & \mathbf{a}_{333333} & = & 0.6007, & \mathbf{a}_{333334} & = & -0.0272, & \mathbf{a}_{333344} & = & -0.1343, \\
\mathbf{a}_{333444} & = & -0.0233, & \mathbf{a}_{334444} & = & -0.0227, & \mathbf{a}_{344444} & = & -0.3555, & \mathbf{a}_{444444} & = & -0.5037.
\end{array}
\]
Fig. C.2: $\mathcal{B}$ from Example 5.4

\begin{align*}
b_{11111} &= 0.2678, & b_{11112} &= -0.0044, & b_{11113} &= -0.0326, & b_{11114} &= -0.0081, \\
b_{11112} &= 0.0591, & b_{11113} &= -0.0009, & b_{11114} &= 0.0045, & b_{11121} &= 0.0533, \\
b_{11113} &= -0.0059, & b_{11114} &= 0.0511, & b_{11122} &= -0.0029, & b_{11123} &= -0.0072, \\
b_{11122} &= -0.0016, & b_{11123} &= -0.0005, & b_{11124} &= 0.0007, & b_{11133} &= -0.0006, \\
b_{11133} &= -0.0185, & b_{11134} &= 0.0001, & b_{11144} &= -0.0058, & b_{11144} &= -0.0046, \\
b_{11222} &= 0.0651, & b_{11223} &= -0.0013, & b_{11224} &= -0.0050, & b_{11233} &= 0.0190, \\
b_{11234} &= -0.0023, & b_{11244} &= 0.0190, & b_{11333} &= -0.0011, & b_{11334} &= -0.0014, \\
b_{11334} &= 0.0006, & b_{11344} &= -0.0043, & b_{11333} &= 0.0498, & b_{11334} &= -0.0061, \\
b_{11344} &= 0.0169, & b_{11344} &= -0.0060, & b_{11444} &= 0.0486, & b_{12222} &= -0.0054, \\
b_{12222} &= -0.0078, & b_{12224} &= -0.0016, & b_{12223} &= -0.0006, & b_{12224} &= 0.0008, \\
b_{12224} &= -0.0006, & b_{12233} &= -0.0067, & b_{12234} &= 0.0001, & b_{12344} &= -0.0022, \\
b_{12244} &= -0.0016, & b_{12333} &= -0.0002, & b_{12334} &= 0.0006, & b_{12344} &= -0.0002, \\
b_{12344} &= 0.0006, & b_{12444} &= -0.0003, & b_{13333} &= -0.0286, & b_{13334} &= 0.0017, \\
b_{13334} &= -0.0056, & b_{13344} &= 0.0001, & b_{13444} &= -0.0051, & b_{14444} &= -0.0073, \\
b_{22222} &= 0.3585, & b_{22222} &= -0.0082, & b_{22224} &= -0.0279, & b_{22223} &= 0.0610, \\
b_{22224} &= -0.0076, & b_{22224} &= 0.0636, & b_{22233} &= -0.0042, & b_{22334} &= -0.0044, \\
b_{22334} &= -0.0002, & b_{22444} &= -0.0145, & b_{22333} &= 0.0518, & b_{22334} &= -0.0067, \\
b_{22334} &= 0.0184, & b_{22444} &= -0.0069, & b_{22444} &= 0.0549, & b_{23333} &= -0.0059, \\
b_{23333} &= -0.0034, & b_{23334} &= -0.0002, & b_{23344} &= -0.0039, & b_{24444} &= 0.0010, \\
b_{24444} &= -0.0208, & b_{33333} &= 0.2192, & b_{33334} &= -0.0294, & b_{33344} &= 0.0477, \\
b_{33344} &= -0.0181, & b_{33444} &= 0.0485, & b_{34444} &= -0.0304, & b_{44444} &= 0.2305. \\
\end{align*}