Support Estimation of Analytic Eigenvectors of Parahermitian Matrices

Faizan Khattak, Ian K. Proudler, and Stephan Weiss
Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, Scotland
{faizan.khattak,ian.proudler,stephan.weiss} @strath.ac.uk

Abstract—Extracting analytic eigenvectors from parahermitian matrices relies on phase smoothing in the discrete Fourier transform (DFT) domain as its most expensive algorithmic component. Some algorithms require an a priori estimate of the eigenvector support and therefore the DFT length, while others iteratively increase the DFT. Thus in this document, we aim to complement the former and to reduce the computational load of the latter by estimating the time-domain support of eigenvectors. The proposed approach is validated via an ensemble of eigenvectors of known support, which the estimated support accurately matches.

I. INTRODUCTION

The polynomial eigenvalue decomposition (PEVD) provides solutions to numerous broadband problems [1]–[14] by diagonalising a space-time covariance matrix \( \mathbf{R}[\tau] = \mathbb{E}\{\mathbf{x}[n]\mathbf{x}^H[n-\tau]\} \), where \( \mathbb{E}\{\cdot\} \) is the expectation operator, \( \mathbf{x}[n] \in \mathbb{C} \) is a vector of measurements, and \( \{\cdot\}^H \) denotes Hermitian transposition. PEVD algorithms can be divided into either time domain or DFT-based methods. All time domain approaches such as [15], [16] and their variants [17]–[21] are techniques which iteratively transfer off-diagonal energy of the space-time covariance matrix onto its diagonal. Time domain algorithms tend to yield spectrally majorised eigenvalues [22], even if the ground truth eigenvalues are not majorised. In this case, algorithms attempt to approximate non-differentiable eigenvalues and discontinuous eigenvectors [23]. This results in much higher approximation orders than their DFT domain counterparts which are capable to extracting the analytic solution [23], [24], which therefore is maximally smooth [25]–[27] and can be well-approximated by lower order polynomials [28]–[31].

While DFT-based methods are superior to time-domain approaches in terms of order of the extracted factors, the algorithms themselves do not scale well with the spatial dimension — in case of the eigenvalues [28], [32] — and the temporal dimension — in case of the eigenvectors [30], [33] — of the input para-hermitian matrix. For the eigenvectors [30], [33], the computational bottleneck is a phase-smoothing operation to establish phase coherence between eigenvectors across frequency bins [28], [30], [31], which is non-convex in nature [30] and NP-hard to solve [33]. The DFT size, which is directly related to the time-domain support of the analytic eigenvectors, therefore determines the algorithm complexity. The approaches in [30], [31], [33] iteratively increase the DFT size eliminating the need of support knowledge, but potentially overestimate the necessary length and thereby exaggerate the complexity. The method in [28] requires an a priori estimate of the support but no such method is provided. Also, if different from [30], [33] the phase-retrieval concepts described in [34]–[36] were to be used for the eigenvector functions, their support would have to be known. This motivates the need to estimate the support of the eigenvector prior to phase smoothing, in order to reduce the algorithmic complexity.

Therefore, this paper presents a method to determine a sufficient DFT size, which in combination with the approaches in [28], [30], [33] allows to extract analytic eigenvectors in close approximation. The proposed approach utilises a smoothness criterion on the auto-correlation of the eigenvectors, which will satisfy real-valuedness in the DFT domain and therefore bypasses phase smoothing. The support can then be determined from this auto-correlation sequence. The accuracy of this proposed method is assessed via an ensemble of para-unitary matrices of known support, thus defining ground-truth eigenvectors in their columns.

II. APPROXIMATE ANALYTIC EIGENVALUE DECOMPOSITION

Consider a matrix \( \mathbf{R}(z) = \sum_\tau \mathbf{R}[\tau]z^{-\tau} \), or short \( \mathbf{R}(z) \leftrightarrow \mathbf{R}[\tau] \), that is analytic in \( z \in \mathbb{C} \) and parahermitian, such that \( \mathbf{R}(z) = \mathbf{R}^P(z) \). This involves the parahermitian operator \( \{\cdot\}^P \) which applies a Hermitian transpose and time reversal, \( \mathbf{R}^P(z) = \mathbf{R}^H(1/z^*) \). For such a matrix \( \mathbf{R}(z) \), generally an analytic eigenvalue decomposition

\[
\mathbf{R}(z) = \mathbf{Q}(z)\Lambda(z)\mathbf{Q}^P(z)
\]

exists [23], [24]. In (1), \( \mathbf{Q}(z) \) contains the analytic eigenvectors and \( \Lambda(z) \) is a diagonal parahermitian matrix that contains analytic eigenvalues.

Being analytic, both \( \mathbf{Q}[n] \leftrightarrow \mathbf{Q}(z) \) and \( \Lambda[n] \leftrightarrow \Lambda(z) \) are absolutely convergent, but potentially infinite Laurent series. Even for an \( \mathbf{R}[\tau] \) with limited time domain support, its analytic decomposition factors are not necessarily of finite order and can even be transcendental [23]. Analyticity implies that by truncation, any convergent Laurent series can be sufficiently well approximated by Laurent polynomial [29]. Specifically with reference to \( \mathbf{Q}[n] \), since we want to obtain a causal solution for the eigenvectors, we can further apply a suitable time delay. The latter is covered by the ambiguity

The work of Faizan Khattak is supported by a Commonwealth Scholarship.
of eigenvectors with respect to allpass functions [29] which includes a delay [37], [38].

Therefore, if $q_m(z) \rightarrow q_m[\tau]$ is the $m$th analytic eigenvector, then its $\Lambda$th order approximation $\hat{q}_m^{(N)}[n]$ is,
\[
\hat{q}_m^{(N)}[n] = p_{N+1}[n]q_m[n + \Delta_m],
\]
where $p_{N+1}[n]$ is rectangular window of length $(N+1)$, such that $p_{N+1}[n] = 1$ for $0 \leq n \leq N$ and $p_{N+1}[n] = 0$ otherwise. The shift $\Delta_m$ influences the truncation error $\zeta_m$.
\[
\zeta_m = \sum_n ||q_m[n] - \hat{q}_m^{(N)}[n - \Delta_m]||^2_2.
\]

Thus if $\zeta$ can be minimized using $\Delta_m$, we can approximate the eigenvector $q_m(z)$ using an $\Lambda$th order approximation. For illustration, we provide the following example:

**Example 1:** A paraunitary matrix $Q(z) \in \mathbb{C}^{2 \times 2}$ of order 100 is generated through a concatenation of elementary paraunitary elements [39]. The norms of $q_m[n] \rightarrow q_m(z)$ are shown in Fig. 1. To approximate $q_m(z)$ by $N = 50$th order polynomials, a minimum for $\zeta_m$ is found for $\Delta_m = 25$ for both $m = 1, 2$. The resulting total normalized truncation error i.e. $\sum \zeta_m / \sum_n ||q_m[n]||^2_2$ is $3.4 \times 10^{-8}$.

### III. EIGENVECTOR SUPPORT ESTIMATION

#### A. EVD on Unit Circle and Ground Truth

All DFT-based PEVD methods [28]–[31], [33] perform EVDs on $R_k = R(z) |_{z = e^{j\Omega_k}}$, $k = 0, ..., K - 1$, where $R_k$ are the $K$ sample points, by evaluating $R(z)$ along the unit circle. Because of $R_k = R_k^H$, for the $k$th bin this EVD is
\[
R_k = Q_k' \Lambda_k' Q_k'^H.
\]
The eigenvalues in $\Lambda_k'$ are assumed to be arranged in descending order for each $k$, which is also known as spectral majorisation. Similarly evaluating (1) at $\Omega_k = 2\pi \frac{k}{K}$ gives us
\[
R(e^{j\Omega_k}) = Q(e^{j\Omega_k}) \Lambda(e^{j\Omega_k}) Q^H(e^{j\Omega_k}).
\]
The relation between sample points of analytic functions in (5) and the bin-wise EVD in (4) can be stated as
\[
\Lambda_k' = P_k \Lambda(e^{j\Omega_k}) P_k^T
\]
\[
Q_k' = \Phi_k Q(e^{j\Omega_k}) V_k^H P_k^T,
\]
where $P_k$ represents a permutation matrix, $V_k$ a block-diagonal unitary matrix [29] and $\Phi_k$ is a diagonal matrix of phase shifts.

In (6) and (7), the permutation matrix $P_k$ re-orders the eigenvalues in each $\Lambda_k'$ to match the order of the sample points of the analytic eigenvalues in $\Lambda(e^{j\Omega_k})$. The block-diagonal unitary matrix $V_k$ is required to adjust the eigenvector basis in case of an algebraic multiplicity of eigenvalues in $\Lambda_k'$ — see [33]; it is an identity matrix if there are no repeated eigenvalues in $Q_k'$. The phase ambiguity of the eigenvectors is expressed by $\Phi_k$, which needs to be adjusted via a phase smoothing operation [30], [33]. The phase smoothing operation is NP-hard and ideally is performed only once if the support of the eigenvectors is known. If the support is unknown, phase smoothing has to be applied to iteratively larger DFT-sizes, as suggested in [33].

#### B. Autocorrelation and DFT Magnitude

For a time domain quantity $u[n] \in \mathbb{C}$ of limited support $N + 1$, its triangularly-weighted, unnormalised autocorrelation sequence $a[\nu]$ is
\[
a[\nu] = \sum_n u[n] u^*[\nu - n],
\]
where $a[\nu] = a^*[-\nu]$ has conjugate symmetry and a support limited to $|\nu| \leq N$. For a DFT $A[k]$ of $a[\nu]$ with a DFT length $K > 2N$, based on the DFT $U[k]$ of $u[n]$, we have [40]
\[
A[k] = |U[k]|^2, \quad \text{for} \quad k = 1, ..., K.
\]
The condition $K > 2N$ ensures that the cyclic convolution implemented in the time domain by (9) is equivalent to the linear convolution in (8) [41]. Note that if only the magnitude $|X[k]|$ is known, this therefore suffices to determine $A[k]$ and thereby $a[\nu]$ and thus the support of $u[n]$.

#### C. Proposed Support Estimation Approach

We intend to apply the idea of Sec. III-B to our case, where we would like to determine the support of $Q_k = \{q_1[k], ..., q_M[k]\}$. Each eigenvector $q_1[k]$ has $M$ components, $q_m[k] = [q_{1,m}[k], ..., q_{M,m}[k]]^T$. To utilize (9), we need to know the unknown support $N$, whereafter $K$ can be decided. Although the support $N$ of the eigenvectors is unknown, we do have access to their DFT magnitude for any $K$ through knowing $P_k$ from the eigenvalues extraction [28], [29] and $V_k$ via [30]. Once eigenvectors are ordered, even if phase smoothing is not yet performed, we can utilize the concept in III-B as the DFT magnitude $|q_m[k]|$ is available. Thus, we express (9) as
\[
A_{i,m}[k] = |q_{i,m}[k]|^2, \quad i, m = 1, \ldots, M, \quad k = 1, \ldots, K
\]
in terms of the eigenvectors components.

In contrast to [40], which assumes the support of $u[n]$ to be limited as described in III-B, the support of $Q_k[\tau] \rightarrow Q_k(z)$ is not necessarily finite [29]. It is therefore challenging to set the appropriate $K$, as otherwise the autocorrelation $a_{i,m}[n]$...
obtained by an IDFT of $A_{(i,m)}[k]$ will incur aliasing or wrap-around due to (9) implementing a cyclic convolution. To resolve this issue, we can use time-domain aliasing as a suitable criterion to identify a sufficiently large $K$, which in turn provide a good estimate for the eigenvector support $N$. Even with an infinite support, since $q_{m}[n]$ is absolutely converging, time-domain aliasing will decrease in $a_{(i,m)}[n]$ with increasing $K$. Thus $K$ can be iteratively increased until time-domain aliasing in $a_{(i,m)}[n]$ becomes negligible. To quantify the time-domain aliasing in $a_{(i,m)}[n]$, its conjugate symmetry property permits us to adopt the strategy of [29], which we outline in the following section.

IV. SUPPORT ESTIMATION VIA TIME-DOMAIN ALIASING

Assume that $A_{(i,m)}[k]$ are the sample points of a $2\pi$-periodic function $A_{(i,m)}(e^{j\Omega}) \mapsto a_{i,m}[\nu]$. If $N$ and therefore $K$ are sufficiently large, then $A_{(i,m)}(e^{j\Omega})$ can be determined from $A_{(i,m)}[k]$ via a Dirichlet interpolation. We exploit this in the following by comparing the interpolations based on $K/2$ and $K$ sample points.

A. Dirichlet Interpolation

Based on the $K$ sample points $A_{(i,m)}[k]$ (or $|q_{i,m}[k]|^2$), a $2\pi$-periodic Dirichlet interpolation $A_{(i,m)}(e^{j\Omega})$ can be denoted as

$$A^{(K)}_{(i,m)}(e^{j\Omega}) = \frac{1}{K} \sum_{k=0}^{K-1} A_{(i,m)}[k] \sum_{t=-L_K}^{L_K} p'_K[t] e^{-j(\Omega-\frac{2\pi k}{K})t},$$

(10)

where $L_K = \frac{K}{2}$ and $\frac{K-1}{2}$ for even and odd $K$, respectively. The interpolant $p'_K[t]$ is, different from the earlier $p_N[n]$, a centred window [29]

$$p'_K[t] = \left\{ \begin{array}{ll}
p_K[t-L_K], & K \text{ odd}, \\
\frac{1}{2}(p_K[t-L_K] + p_K[t-L_K+1]), & K \text{ even}.
\end{array} \right.$$

(11)

The interpolation (10) can be brought into vectorial form

$$A^{(K)}_{(i,m)}(e^{j\Omega}) = \frac{1}{\sqrt{K}} e^{H}_{K}(e^{j\Omega}) O_K T_K A^{(K)}_{(i,m)},$$

(12)

where

$$e^{H}_{K}(e^{j\Omega}) = [e^{j\Omega L_K}, e^{j\Omega-1}, \ldots, e^{-j\Omega L_K}],$$

$$O_K = \text{diag}(p'[−L_K], p'[−L_K+1], \ldots, p'[L_K]),$$

$$T_K = \begin{bmatrix}
0_{L_K \times (K-L_K)} & I_{L_K} \\
0_{L_K+1 \times (K-L_K)} & 0_{(L_K+1) \times (K-L_K-1)}
\end{bmatrix},$$

$$A^{(K)}_{(i,m)} = [A_{(i,m)}[1], \ldots, A_{(i,m)}[K]]^{T},$$

with $W_K$ denoting a unitary DFT matrix.

B. Detecting Time Domain Aliasing

Based on the above formulations, we can now assess the difference $\Theta^{(K)}_{(i,m)}(e^{j\Omega})$ between interpolations over $K$ and $K/2$ sample points,

$$\Theta^{(K)}_{(i,m)}(e^{j\Omega}) = A^{(K)}_{(i,m)}(e^{j\Omega}) - A^{(K/2)}_{(i,m)}(e^{j\Omega}).$$

(13)

This difference should tend to zero for increasing $K$ as $a_{i,m}[\nu]$ is analytic and therefore absolutely convergent. Thus for sufficiently large $K$, time-domain aliasing will be negligible. Note that $K$ should be even in (13) as other wise $K/2 \notin \mathbb{Z}$.

Since $A^{(K)}_{(i,m)}(e^{j\Omega})$ and $A^{(K/2)}_{(i,m)}(e^{j\Omega})$ will be identical in $K/2$ sample points but differ in their derivatives there, the difference can be exaggerated by assessing the power in the $p$th derivative of (13). Thus, a suitable decision metric $\xi^{p}_{m}$ similar to the procedure in [29] is

$$\xi^{p}_{m} = \frac{1}{2\pi} \sum_{i=1}^{M} \int_{\pi}^{\pi} \left| \frac{d^{p}}{d\Omega^{p}} \Theta^{(K)}_{(i,m)}(e^{j\Omega}) \right|^{2}.$$  

(14)

While $\xi^{p}_{m} \rightarrow 0$ is necessary for convergence, it can also be shown to be a sufficient criterion since $A_{(i,m)}(\Omega)$ is analytic [29].

For the evaluation of the derivatives of $\Theta^{(K)}_{(i,m)}(e^{j\Omega})$, we can utilise (12), and with some minor restructuring write

$$\frac{d^{p}}{d\Omega^{p}} \Theta^{(K)}_{(i,m)}(e^{j\Omega}) = e^{H}_{K}(e^{j\Omega}) \left( \frac{1}{\sqrt{K}} D^{H}_{K} T_{K} - \sqrt{\frac{2}{K}} M^{2}_{K} D^{H}_{K/2} T_{K/2} M_{1} \right) A^{(K)}_{(i,m)}$$

(15)

where for $K$ even,

$$D_{K} = \text{diag}\{ 1/2(-jL_{K})^{p}, (-j(L_{K}-1))^{p}, \ldots, 1/2(jL_{K})^{p} \}$$

and for $K$ odd,

$$D_{K} = \text{diag}\{ (-jL_{K})^{p}, (-j(L_{K}-1))^{p}, \ldots, (jL_{K})^{p} \}.$$

Further, we have

$$M_{1} = I_{K/2} \otimes [1 0],$$

(16)

$$M_{2} = [0_{(K/2+1) \times L_{K/2}} I_{K/2+1} 0_{(K/2+1) \times L_{K/2}}].$$

(17)

In (16), $\otimes$ denotes the Kronecker product.

Using (15) and Parseval’s theorem, we can write (14) as

$$\xi^{p}_{m} = \frac{1}{2\pi} \sum_{m=1}^{M} \| B_{K} A^{(K)}_{m} \|_{F}^{2},$$

(18)

where $A^{(K)}_{m} = [A^{(K)}_{(1,m)}, \ldots, A^{(K)}_{(M,m)}] \in \mathbb{R}^{K \times M}$. Once for some $K$ we reach $\xi^{p}_{m} \leq \epsilon$, with $\epsilon$ a suitably small threshold, it means that $K/2$ is sufficient to obtain $a_{(i,m)}[n], i, m = 1, \ldots, M$ from an IDFT of $A^{(K/2)}_{(i,m)}[k]$ with a sufficiently small aliasing term. The sufficiency of $K/2$ as DFT size for the autocorrelation term also implies that, following the rationale of Sec. III-B, $K/4$ is comparable or greater than the support of the eigenvectors, $N$. Based on $N = K/4$ as an initial crude estimate, trimming can be applied to further refine the support to the required accuracy, as will be discussed below.
Algorithm 1: Sufficient DFT Size Estimation

Input: $R(z)$, $\epsilon$, $p$
Output: $K$

obtain eigenvalues $A(z)$ of $R(z)$ using [29] or [28]
$K \leftarrow 2^{\left\lceil \log_2 \text{Ord}(R(z)) \right\rceil - 1}$

while $\xi^p > \epsilon$ do
  $K \leftarrow 2K$
  determine $Q_k^*$ from EVD of $R_k$, $k = 1, \ldots, K$
  obtain $P_k$ from $A(z)$ and $V_k$ from [30]
  $Q_k' \leftarrow Q_k^*P_kV_k$
  calculate $\xi^p_K$ for $Q_k'$ via (18)
end

C. Iterative Procedure and Final Adjustment

The comparison between aliasing for DFT sizes of $K/2$ and $K$ suggests an iterative scheme that doubles the DFT length in every iteration step until a required precision is reached. This process is outlined in Algorithm 1. The doubling of the DFT size also has computational advantages. A DFT does not need to be entirely recomputed at every step, since half of the bins have already been determined in the previous iteration. Likewise, when increasing the DFT length from $K/2$ to $K$, only half of the EVDs — the ones in odd-indexed bins — need to be calculated. To further exploit efficient fast Fourier transform structures for DFT, Algorithm 1 selects powers of 2 for $K$.

Once the suggested threshold for $\xi^p_K$ is passed, we know that $K/2$ is a suitable support of $a_{i,m}[\nu]$ and that therefore for eigenvector support $\hat{N}$, we have $K/8 < \hat{N} \leq K/4$. Based on these bounds, the support estimate $\hat{N}$ can be further refined, for which two approaches are outlined below.

1) Truncation of Autocorrelation Sequence: After Algorithm 1 returns a sufficient DFT size, the autocorrelation $a_{i,m}[\nu]$ can be determined from $A_{i,m}^{K/2}$ with a small error due to aliasing that depends on $\epsilon$ via $K/2$-point IDFT. The operation

$$a_{i,m}^{K/2} = \sqrt{\frac{2}{K}} W_{K/2}^H A_{i,m}^{K/2}$$

returns the coefficients $a_{i,m}[\nu]$ in the vector $a_{i,m}^{K/2} \in \mathbb{C}^{K/2}$. Since the autocorrelation is conjugate symmetric, we can apply a symmetric truncation to enhance the support estimate $\hat{N}$. The truncation error can be given as

$$\xi_{i,m} = \frac{\sum_{n=J+1}^{K/4} |a_{i,m}[n]|^2}{|a_{i,m}^{K/2}|^2},$$

where $J \in \mathbb{Z}$ is varied from $K/8$ to $K/4$ until the truncation error decreases below some threshold $\epsilon'$. The resulting $J$ can be considered as an estimate for $\hat{N}$. Such estimates can be obtained for all $M$ components across the $M$ eigenvectors, and the largest $\hat{N}$ can be used for $Q_N^H[n]$.  

2) Extra Iterations: Knowing via Algorithm 1 that $K/8 < \hat{N} \leq K/4$, the DFT size can be iteratively refined between $K/2$ and $K$. This approach is more systematic and expensive as compared to truncation, as additional iterations are required.

Algorithm 2 outlines the procedure where $\Delta = K/4$ is decreased iteratively until it reaches $\Delta_{\text{min}}$.

V. SIMULATIONS AND RESULTS

A. Simulation Scenario

An ensemble of 200 instances of paraunitary matrices $Q(z) \in \mathbb{C}^{2 \times 2}$ is generated by concatenating elementary paraunitary operations [39], where the support of the columns — here taken as the ground truth eigenvectors — varies from 20 to 200 in steps of 20. This ensemble is used to test the accuracy of the proposed methods. As shown in Example 1, the eigenvector coefficients can decay below machine precision, and the ground-truth support becomes meaningless. To avoid this issue, the ensemble is created such that absolute values of eigenvector coefficients do not fall below $10^{-6}$ i.e.

$|q_{i,m}[n]| \geq 10^{-6}, n = 0, \ldots, N-1$. Simulations are carried out with a derivative order $p = 2$, a threshold $\epsilon = 10^{-10}$, increments of $\Delta_{\text{min}} = 2$ in the refinement procedure, and truncation threshold of $\epsilon' = \{10^{-15}, 10^{-20}, 10^{-30}\}$.

B. Ensemble Results

Ensemble results for both methods are shown in Fig. 2 and Fig. 3 illustrating the accuracy of the proposed estimation methods. For a low cost estimation, truncation can be performed directly on the auto-correlation sequence $a_{i,m}^{K/2}$ as mentioned in Sec. IV-C1. It can be seen that different thresholds for the truncation error (19) produce slight variation in the results for high order polynomials. However, for all three threshold values, the standard deviation of the results is
negligible even for large $N$. The standard deviation is zero for $\hat{\mu}' = 10^{-30}$ and $\hat{N}$ is equal to $N$ for the entire ensemble.

For the refined iteration using Algorithm 2, results are shown in Fig. 3. For the ensemble over which the simulations were performed, this method was able to extract the exact support with zero standard deviation, although the small gain in precision for large $N$ comes at a higher computations cost compared to the truncation approach shown in Fig. 2.

VI. APPLICATION EXAMPLE

As an example for the impact of the proposed approach, we want to demonstrate the impact on support estimation for the extraction of analytic eigenvectors. The state-of-the-art approach in [33] is based on an iterative increase of the DFT length by factors of two; the correct order is then obtained by checking against various criteria, followed by a potential truncation in the time domain akin to trimming applied to the original SBR2 and SMD algorithms [37], [38], [42], [43].

We assume that a ground truth space-time covariance matrix is constructed via $\mathbf{R}(z) = \mathbf{Q}(z)\Lambda(z)\mathbf{Q}^H(z)$, where $\Lambda(z)$ is generated via a source model [16], and $\mathbf{Q}(z)$ akin to Sec. V-A via concatenated elementary paraunitary operations [39] of a specified order $O\{\mathbf{Q}(z)\} \in [1, 12]$. Results below are based on an ensemble of 1000 instance of $\mathbf{Q}(z)$ for each order.

The extraction in [33] assumes that the eigenvalues have already been determined e.g. via [29]; its main bottleneck is a phase smoothing algorithm, that in [33] has to be determined for several different DFT sizes until a convergence is found. This results in a relatively high computational load, with the execution time shown in Fig. 4, showing curves for the median and the bounds into which 90% of the ensemble probes fall. Note that since the DFT-size is increased in powers of two, the execution time appears in steps. In contrast, if the eigenvector extraction algorithm takes into account the correct support of the eigenvectors using the proposed approach described in this paper, the phase smoothing algorithm only has to be calculated for a single DFT size, resulting in Fig. 4 in a considerably lower cost that increases strictly monotonically with $O\{\mathbf{Q}(z)\}$.

Therefore, the correct time domain support estimation directly impacts on the complexity of any subsequent applications [1]–[14], [44], [45], and complements the significant efforts that have been dedicated to the efficient implementation of polynomial matrix factorisations, see e.g. [46]–[48].

VII. CONCLUSION

Based on the fact that the autocorrelation function of any function with limited time-domain support can be obtained if the function’s DFT magnitude is known, we have applied this to estimate the support of analytic eigenvectors of parahermitian matrices. The proposed method utilises an iterative approach to find a range estimate for the support based on measuring residual aliasing, which is then refined either by truncation or by further and finer iteration. The algorithms are validated via an ensemble of eigenvectors of different support. The estimated support closely matches the ground truth provided that the dynamic range of the coefficients of the eigenvectors are sensibly limited.

The solution presented in this paper can provide a starting point for the algorithms in [28], [30], [31] that aim to extract analytic eigenvectors from parahermitian matrices. This alleviates a previous bottleneck, as now the estimate can be formed before a costly phase smoothing step is invoked, thus presenting a step forward towards solutions that scale well with the matrix dimension.