<table>
<thead>
<tr>
<th>Simple Effective Computation of Principal Eigenvectors</th>
<th>Rhode Island Univ. Kingston Dept. of Electrical Engineering D W Tufts et al. Oct 85</th>
</tr>
</thead>
</table>

UNCLASSIFIED N00014-83-K-0664
FINAL REPORT

SIMPLE, EFFECTIVE COMPUTATION OF PRINCIPAL EIGEN-VECTORS AND THEIR EIGENVALUES AND APPLICATION TO HIGH-RESOLUTION ESTIMATION OF FREQUENCIES

Donald W. Tufts
Costas D. Melissinos

Department of Electrical Engineering
University of Rhode Island
Kingston, Rhode Island 02881

October 1985

Probability and Statistics Program
Electronics Program
Office of Naval Research
Arlington, Virginia 22217
under Contracts N00014-83-K-0664 & N00014-84-K-0445
D.W. Tufts, Principal Investigator

&

partially supported
under Contract ECS-8408997, NSF

Approved for public release; distribution unlimited
**Title:** Simple, Effective Computation of Principal Eigenvectors and their Eigenvalues and Application to High-Resolution Estimation of Frequencies

**Authors:**
- D.W. Tufts
- C.D. Melissinos

**Performing Organization:**
- Department of Electrical Engineering
  - University of Rhode Island
  - Kingston, RI 02881

**Controlling Office:**
- Probability and Statistics Program; Electronics Program
  - Office of Naval Research (Code 411SP)
  - Dept. of the Navy, Arlington, VA 22217

**Report Date:**
- October 1985

**Number of Pages:**
- 30

**Security Classification:**
- Unclassified

---

**Abstract:**

We present the results of an investigation of the Prony-Lanczos (P-L) method and the power method for simple computation of approximations to a few eigenvectors and eigenvalues of a Hermitian matrix. We are motivated by realization of high-resolution signal processing in an integrated circuit. The computational speeds of the above methods are analyzed. They are completely dependent on the speed of a matrix-vector product operation. If only a few eigenvalues or eigenvectors are needed, the suggested methods can substitute for the slower methods of the LINPACK or EISPACK subroutine.
libraries. The accuracies of the suggested methods are evaluated using matrices formed from simulated data consisting of two sinusoids plus Gaussian noise. Comparisons are made with the corresponding eigenvalues and eigenvectors obtained using LINPACK. Also the accuracies of frequency estimates obtained from the eigenvectors are compared.
Simple, Effective Computation of Principal Eigenvectors and their Eigenvalues and Application to High-Resolution Estimation of Frequencies

D.W. Tufts and C.D. Melissinos

Department of Electrical Engineering
University of Rhode Island
Kingston, RI 02881

Abstract

We present the results of an investigation of the Prony-Lanczos (P-L) method [14,38] and the power method [39] for simple computation of approximations to a few eigenvectors and eigenvalues of a Hermitian matrix. We are motivated by realization of high-resolution signal processing in an integrated circuit. The computational speeds of the above methods are analyzed. They are completely dependent on the speed of a matrix-vector product operation. If only a few eigenvalues or eigenvectors are needed, the suggested methods can substitute for the slower methods of the LINPACK or EISPACK subroutine libraries. The accuracies of the suggested methods are evaluated using matrices formed from simulated data consisting of two sinusoids plus gaussian noise. Comparisons are made with the corresponding eigenvalues and eigenvectors obtained using LINPACK. Also the accuracies of frequency estimates obtained from the eigenvectors are compared.

This work was supported in part by the Electronics Program of the Office of Naval Research.
I. **Introduction**

We are motivated by the use of eigenvector decompositions of data matrices or estimated covariance matrices for detection of signals in noise and for estimation of signal parameters. This has evolved from early work of Liggett [1] and Owsley [2], to adaptive-array-detection improvements of Tufts and Kirsteins [3,33] and high-resolution parameter estimators of Cantoni and Godara [4], Bienvenu and Kopp [5], Owsley [6], Schmidt [21] and Tufts and Kumaresan [7,32].

Principal component analysis, using principal eigenvalues and eigenvectors of a matrix, was initiated by Karl Pearson (1901) [8], and Frisch (1929) [9] in the problem of fitting a line, a plane or in general a subspace to a scatter of points in a higher dimensional space. Eckart and Young [34] presented the use of singular value decomposition for finding low-rank approximations to rectangular matrices. C.R. Rao examined the applications of principal component analysis [10]. Eigenvector analysis is also used in image processing to provide efficient representations of pictures [11]. Recently, principal component analysis has been coupled with the Wigner mixed time-frequency signal representation to perform a variety of signal processing operations [28,30,31].

Linear Prediction techniques for estimation of signal parameters, which are modern variants of Prony’s method, can be improved using eigenvector decomposition [7]. Prony’s method is a simple procedure for determining the values of parameters of a linear combination of exponential functions. Now "Prony’s method" is usually taken to mean the least squares extension of the method as presented by Hildebrand [13]. The errors in signal parameters which are estimated by Prony’s method can be very large [14]. If the data is composed of undamped sinusoids, the forward and backward prediction
equations and a prediction order larger than the number of signal components can be used simultaneously as advocated by Nutall [22], Ulrych and Clayton [23], and Lang and McClellan [24]. Tufts and Kumaresan have shown how one can improve such methods of parameter estimation by going through a preprocessing step before application of Prony's method [7,15,16,17]. The measured data matrix or the matrix of estimated covariances is replaced by a matrix of rank $M$, which is the best least squares approximation to the given matrix. If there is no prior information about the value of $M$, it is estimated from the data using singular value decomposition (SVD).

The eigenvalue problem [37] is one area where extensive research has been done and well established algorithms are available in highly optimized mathematical libraries such as LINPACK and EISPACK [40]. The computational complexity of these algorithms is of order $O(N^3)$ where $N$ is the size of the matrix. They solve for the complete set of eigenvalues and eigenvectors of the matrix even if the problem requires only a small subset of them to be computed. For the above applications, only a few principal eigenvectors and eigenvalues are needed. Hence, we would like to use a method which uses this specialization to reduce the computations.

Tufts and Kumaresan [29,32,33] have suggested procedures for improving Prony's method without computation of eigenvectors. These appear to perform about the same as the more complicated approaches which use eigenvalue and eigenvector decomposition. The approach in [29] is based on the results of Hocking and Leslie for efficient selection of a best subset [25]. The approach of [32] and [33] is based on the simple computations which result from using the longest possible prediction interval.
Here we investigate two different approaches to achieving SVD-like improvement to Prony’s method without the computational cost of actually computing the SVD or computing all eigenvectors and eigenvalues. The idea is to calculate the few, necessary eigenvalues and eigenvectors using the power method [39] and a method of Lanczos [14]. Our derivation of Lanczos' method stresses the connection with Prony's method. The methods are analyzed and their amounts of computation are calculated. Simulations are performed and results are compared to the singular value decomposition method in LINPACK.

II. The Prony-Lanczos Method

Let us assume that we start with a given square, Hermitian matrix $A$ for which we want to compute the principal eigenvectors and eigenvalues. For examples, this could be either the true underlying, population covariance matrix or the estimated covariance matrix [36] from spatial or temporal data. Let us also define the eigenvectors and eigenvalues associated with the matrix $A$ (dimension $A=n$).

$$ Au_i = \lambda_i u_i , \ i = 1,2,...,n $$

(1)

where $u_i^* u_j = 0 , \ i \neq j$

$$ u_i^* u_j = 1 , \ i = j , \ \text{that is} \ u_i \ \text{are orthonormal vectors.} $$

(2)

The asterisk is used to denote a complex conjugate transpose.

The characteristic polynomial associated with the matrix $A$ is given by

$$ \det(A - \lambda I) = 0 $$

(3)

Expanding the determinant we have the polynomial equation

$$ \lambda^n + p_{n-1} \lambda^{n-1} + \ldots + p_0 = 0 $$

(4)

and the roots of this polynomial will give us the eigenvalues $\lambda_i$ of the matrix. We briefly summarize the procedure for obtaining the eigenvalues $\lambda_i$
based on the Lanczos "power sums" as presented in [14]. We shall show that the eigenvalues can then be obtained from the power sums by Prony's method [13].

Let us select a starting vector \( \mathbf{b}_0 \). We assume that the starting vector \( \mathbf{b}_0 \) has a non-zero projection on the eigenvectors of the matrix \( A \) corresponding to the eigenvalues that we want to compute.

We then analyze the vector \( \mathbf{b}_0 \) in the reference system of the vectors \( \{ \mathbf{u}_i \} \), which are the set of orthonormal eigenvectors of the matrix \( A \):

\[
\mathbf{b}_0 = \tau_1 \mathbf{u}_1 + \tau_2 \mathbf{u}_2 + \cdots + \tau_n \mathbf{u}_n
\]

where

\[
\tau_i = \mathbf{u}_i^* \mathbf{b}_0
\]

Hence, using equation (1), successive vectors formed by premultiplications of \( \mathbf{b}_0 \) by powers of the matrix \( A \) can be represented as follows:

\[
\begin{align*}
\mathbf{b}_1 &= A \mathbf{b}_0 = \tau_1 \lambda_1 \mathbf{u}_1 + \tau_2 \lambda_2 \mathbf{u}_2 + \cdots + \tau_n \lambda_n \mathbf{u}_n \\
\mathbf{b}_2 &= A^2 \mathbf{b}_0 = A \mathbf{b}_1 = A^2 \mathbf{b}_0 = \tau_1 \lambda_1^2 \mathbf{u}_1 + \tau_2 \lambda_2^2 \mathbf{u}_2 + \cdots + \tau_n \lambda_n^2 \mathbf{u}_n
\end{align*}
\]

Let us form the set of basic scalars:

\[
\mathbf{c}_k = \mathbf{b}_k^* \mathbf{b}_k = \mathbf{b}_k^* \mathbf{b}_k
\]

Then we shall have:

\[
\mathbf{c}_k = |\tau_1|^2 \lambda_1^k + |\tau_2|^2 \lambda_2^k + \cdots + |\tau_n|^2 \lambda_n^k = \mathbf{b}_0^* A^k \mathbf{b}_0
\]

which were called by Lanczos the "weighted power sums" [14].

The problem of obtaining \( \lambda_i \)'s from the \( \mathbf{c}_i \)'s is the "problem of weighted moments" [14]. That is the problem of Prony [12] and the old and modern versions of Prony's method can be used to estimate the \( \lambda_i \)'s.

The prediction-error-filter equations of Prony's method can be written as follows:
\[ \begin{align*}
  & c_0 g_0 + c_1 g_1 + \ldots + c_{n-1} g_{n-1} + c_n = 0 \\
  & c_1 g_0 + c_2 g_1 + \ldots + c_{n} g_{n-1} + c_{n+1} = 0 \\
  & \quad \vdots \\
  & c_n g_0 + c_{n+1} g_1 + \ldots + c_{2n-1} g_{n-1} + c_{2n} = 0 
\end{align*} \]

or in matrix form,

\[ C \cdot g = 0 \]  

A non-zero solution is possible if the determinant of \( C \) is zero.

From the theory of Prony's method [13]

\[ g(\lambda_i) = \lambda_i^n + g_{n-1} \lambda_i^{n-1} + \ldots + g_1 \lambda_i + g_0 = 0 \]  

hence the polynomial coefficient vector \( g \) is also orthogonal to the vector \( \{1, \lambda_i, \lambda_i^2, \ldots, \lambda_i^n\}^T \) where \( \lambda_i \)'s are the eigenvalues of the matrix \( A \).

Lanczos noticed that Prony's method can be simplified if we substitute the sequence \( \{1, \lambda_i, \lambda_i^2, \ldots, \lambda_i^n\} \) for a row of the matrix \( C \) to form a matrix \( C' \). If we replace the matrix \( C \) by \( C' \) in (10b), the non-zero vector \( g \) is still a solution, because of (11). Hence the determinant of \( C' \) must be zero.

\[ \begin{vmatrix}
  1 & 1 & \lambda_i & \lambda_i^2 & \ldots & \lambda_i^n \\
  c_0 & c_1 & c_2 & \ldots & c_n \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  c_{n-1} & c_n & \ldots & \ldots & c_{2n-1} 
\end{vmatrix} = p'(\lambda_i) = 0 \]  

Hence, the \( \lambda_i \)'s can be obtained directly by finding the zeros of the polynomial \( p'(z) \). That is, Lanczos showed that it is not necessary to first
solve equations (10) for the prediction-error-filter coefficients.

Thus, in the absence of noise, we know that entering the weighted power sums \( c_k \) of (8) in equation (12) and finding the roots of the resulting polynomial will provide us with accurate estimates of the true eigenvalues \( \lambda_1 \) of the covariance matrix \( A \). Note also, that equation (12) can be reduced to a 2nd order equation involving only \( c_0, c_1, c_2, c_3 \) and still provide us with accurate solutions for our problem of estimating one or two sinusoids.

Now, if our data is composed of one or two complex sinusoids, then the \((LxL)\) covariance matrix elements will be also one sinusoid or a sum of two sinusoids, hence the rank of the matrix will be one or two, respectively. The eigen-decomposition of the matrix will show that it has only one or two non-zero eigenvalues and hence it can be characterized by a linear combination of one or two eigenvectors, corresponding to the principal non-zero eigenvalues. In Appendix A it is shown that these eigenvectors can be expressed as a linear combination of complex sinusoids which have frequencies equal to these of the sinusoids composing the data.

Now, suppose that we have accurately determined a few eigenvalues, say two, \( \lambda_1 \) and \( \lambda_2 \), from the \((nxn)\) matrix \( A \). We wish to determine the corresponding eigenvectors. Two concepts are used: (a) premultiplication of a vector by the matrix \((A-\lambda I)\) removes the \( i \)th eigenvector component of that vector and (b) if a vector, to a good approximation, consists only of \( M \) eigenvector components, then removing \((M-1)\) of these components leaves one isolated eigenvector component.

Let us consider the special case of a rank two matrix:

\[
A = \lambda_1 u_1 u_1^* + \lambda_2 u_2 u_2^* \quad (13)
\]
From equations (5) and (13) we have:

\[ Ab_0 = \tau_1\lambda_1 u_1 + \tau_2\lambda_2 u_2 \]  

(14)

Then, our preliminary, unnormalized estimates of the two principal eigenvectors are:

\[ u_1' = (A-\lambda_2I)Ab_0 = (A-\lambda_2I)(\tau_1\lambda_1 u_1 + \tau_2\lambda_2 u_2) = \]
\[ = \tau_1\lambda_1^2 u_1 + \tau_2\lambda_2^2 u_2 - \tau_1\lambda_1\lambda_2 u_1 - \tau_2\lambda_2 u_2 = \]
\[ = \tau_1\lambda_1(\lambda_1 - \lambda_2)u_1 \]  

(15)

And similarly for the second eigenvector estimate we have:

\[ u_2' = \tau_2\lambda_2(\lambda_2 - \lambda_1)u_2 \]  

(16)

Normalizing the eigenvectors \( u_1' \) (i=1,2) we can write (15) and (16) as

\[ u_1' = e^{j\theta_1}u_1 \ ; \ \theta_1 = \text{angle of } \tau_1\lambda_1(\lambda_1 - \lambda_2) \]  

(17)

\[ u_2' = e^{j\theta_2}u_2 \ ; \ \theta_2 = \text{angle of } \tau_2\lambda_2(\lambda_2 - \lambda_1) \]  

(18)

In general, given the required eigenvalues from the earlier Prony calculation, we estimate an unnormalized \( k \)th eigenvector from the formula:

\[ u_k' = (A-\lambda_kI)Ab_0 \ ; \ i \neq k \]  

(19)
where the number of factors in the product depends on the number of significant eigenvector components in $A b_0$.

Finally, a few comments should be made on the selection of the starting vector $b_0$. Our sole assumption until now has been that $b_0$ has a non-zero projection on some eigenvector of $A$ that we want to compute. A good $b_0$ vector would have to be biased in favor of the principal eigenvectors. We have found that the Fourier vector provides a very good selection for $b_0$. This vector will have its fundamental frequency computed from the maximum peak of the DFT data spectrum. Very frequently in signal processing applications the data is preprocessed through a DFT step for a coarse analysis. This is a valuable bonus for our method to use the available information for further processing.

III. The Power Method

Suppose $A$ is a Hermitian $(nxn)$ matrix. The SVD theorem [37] states that $A$ can be written as:

\[ A = U \cdot S \cdot U^T \]  

(20)

where $U$ is a unitary matrix and $S$ is a matrix consisting of real only diagonal elements [37].

The power method computes the dominating singular vectors one at a time and is based on solving the equation:

\[ s u = A u \]  

(21)
for the singular vector $u$ and the singular value $s$. The power method uses an iterative scheme to solve (21). We instead suggest a two-step solution using an appropriate starting vector $b_0$:

$$ u_1 = \frac{A b_0}{\| A b_0 \|} \quad (22) $$

The singular value is chosen to be:

$$ s_1 = \| A b_0 \| \quad (23) $$

In order to obtain the next singular vector, the estimated singular plane $(u_1 u_1^T)$ is removed from $A$ using the following deflation procedure [37]:

$$ A' = A - s_1 u_1 u_1^T \quad (24) $$

and the procedure is repeated with matrix $A$ to yield $s_2, u_2$.

The selection of $b_0$ is very important and the Fourier vector provides a very good estimate. This preprocessing step can be implemented in VLSI very efficiently using summation-by-parts [28] or the Fast Hartley Transform [42,43] methods. A necessary thing required to implement the power method is a circuit capable of computing matrix vector products of the form $A u$. But the rounding errors associated with it are always worrisome limiting the usefulness of the power method. For this reason we propose to use the permuted difference coefficients (PDC) algorithm [26,27] coupled with the known Fourier vector to perform the above operation with high accuracy and no round-off errors. A VLSI implementation for the PDC algorithm can be easily realized using a random access memory (RAM) together with a read-
only-memory (ROM) where the original Fourier coefficients and the subsequent reordered coefficients addresses are stored.

IV. Operation count

In this section we calculate the total operations needed for the singular value decomposition (LINPACK), the Prony-Lanczos method and the Power method.

1. The matrix eigenvalue problem has been solved in both LINPACK and EISPACK mathematical libraries. The LINPACK SVD routine is presented here. The solution can be divided in three steps: reduction to bidiagonal form, initialization of the right and left unitary matrices \( U \) and \( V \) and the iterative reduction to diagonal form.

The reduction to bidiagonal form has the following floating point multiplication count (for a square \( N \times N \) matrix):

\[
2\left[ N^3 - \frac{N^3}{3} \right]
\]

Approximately the same number of additions are required.

In the second step the amount of work involved when only the right-hand side matrix \( V \) is computed, is:

\[
\frac{2N^3}{3}
\]

floating point multiplies and approximately the same number of additions.

In the last step rotations are used to reduce the bidiagonal matrix to diagonal form. Thus the amount of work depends on the total number of
rotations needed. If this number is \( r \), then we have the following multiplication counts:

\[
4Nr
\]

The number \( r \) is quite difficult to estimate. There exists an upper bound for \( r \),

\[
r \leq sN^2/2
\]

where \( s \) is the maximum number of iterations required to reduce a superdiagonal element as to be considered zero by the convergence criterion.

Hence the total operation count for the LINPACK SVD solution is:

\[
2N^3+4Nr \leq 2N^3(s+1) \text{ flops}
\]

where by the term "flop" we denote a floating point multiply-add operation.

(2). The Prony-Lanczos method is entirely dependent on the speed of a matrix-vector product operation. For a rank two square matrix of size \( N \) we shall have:

The matrix-vector multiplications to determine the vectors \( b_i \) involve \( N \) floating point multiplications and \( (N-1) \) floating point additions per row for a total of:

\[
N^2 \text{ flops}
\]

(\( 2N^2 \) flops for the two vectors \( b_1, b_2 \)). The scalar weights \( c_i, i=0,1,2,3 \)
require vector–vector inner products for a count of N multiplications and (N-1) additions per weight. Therefore the total is:

4N flops

The computation of the eigenvalues from the (second order) determinant condition involves 12 flops and one square root calculation. Finally, the eigenvector computation requires N flops for each vector for a total of 2N flops.

Hence the total operation count for the Prony–Lanczos procedure requires:

(2N^2 + 6N + 12) flops + 1 square root

The above computations do not include the work required to select the starting vector b_0 using a DFT analysis. In this case, assuming a data sequence zero padded to M points, we shall have:

Mlog_2M flops

plus (M-1) additions for the determination of the maximum spectral peak.

(3) The power method computes the dominating eigenvalues and eigenvectors one pair at a time. The second pair will be computed following a deflation of A. In general, the number of iteration steps depend on the convergence criterion severity. We instead claim that two-steps are generally enough to provide sufficient accuracy. The Fourier vector is again selected as the starting vector b_0.
The first eigenvalue/eigenvector pair requires $2N^2 + 2N$ flops. The deflation step requires $N^2$ flops and $N^2$ floating point additions.

Hence (for a rank two matrix) the power method requires a total of

$$5N^2 + 4N \text{ flops}$$

plus $N^2$ floating point additions.

V. Simulation results

Let us assume that we have a data sequence which is composed of uniformly spaced samples of two closely spaced complex sinusoids in white noise. We shall follow the methods described earlier in section II & III to calculate the principal eigenvalues and eigenvectors.

The data sequence is given by the equation

$$x(n) = \exp(j2\pi f_1 n + \phi_1) + \exp(j2\pi f_2 n + \phi_2) + w(n) \quad (25)$$

with $f_1 = 0.52\text{Hz}$, $f_2 = 0.5\text{Hz}$ and for $n=1,2,\ldots,25$

Here, 25 data samples are used and the phase difference is $\Delta \phi = \pi/2$ computed at the middle of the data set, effectively reducing the signal-to-noise ratio in that region, thereby representing the worst case that can be encountered. The frequency separation is less than the reciprocal of the observation time. The data is zero padded to $m=128$ points and then the maximum peak of the DFT is computed to yield the frequency of the Fourier vector. This vector will be used as a starting eigenvector for the P-L and Power methods later.

We construct the forward plus backward augmented covariance matrix $A$ of
size (21x21). Its effective rank is two. The SVD routine, the P-L method and the Power method are employed to solve for the eigenvalues and eigenvectors (eigenpairs) of the matrix. The P-L method and the Power method compute only the two principal eigenpairs. The mean values and standard deviations of the eigenvalue estimates are given in Table I for an ensemble of 500 experiments. The performance of the P-L and Power methods is almost identical to the SVD (LINPACK) method for the first eigenvalue estimates. At high SNR the second eigenvalue mean and standard deviation estimate obtained from the P-L method is biased with respect to the noiseless SVD results. However, at low SNR the eigenvalue statistics are closer to the noiseless SVD results than the other two methods.

Table II presents the statistics of the distances of the P-L and Power methods eigenvectors from those of the SVD method. The distance is the inverse cosine of the angle between the subspaces spanned by the estimated eigenvectors [41]. The results show that for the first eigenvector the P-L estimate of the mean is less biased (about one order of magnitude) than the Power method, whereas for the second eigenvector estimates they perform the same. This shows that these vectors span virtually the same subspace as the vectors computed from the SVD method. The eigenvector estimates were also compared to the signal eigenvectors and the distances were computed as above. The results show that at high SNR the eigenvector spanned subspaces have a greater distance from the signal subspace than the SVD subspace. At low SNR the distance is reduced and the second eigenvector statistics are closer to the signal eigenvector than the SVD eigenvector.

Table III shows the CPU time required to compute the eigenvalues/eigenvectors pairs for these methods. The P-L method is faster than the SVD by the order of the size of the covariance matrix, which here
is 21. This roughly agrees with the theoretical operation count we presented in section IV. It is almost twice as fast as the Power method. Inclusion of the FFT computation in these two methods will offset some of their speed advantage over the SVD. Nevertheless, the P-L method is again about one order of magnitude faster than the SVD method and the Power method a little more than half that (6 times faster).

The frequencies $f_i$ are then obtained from the eigenvectors of the estimated covariance matrix by the T-K method [7]. For both estimates of the mean and standard deviation, as presented in Table IV, all three methods perform similarly down to 15 db. At 0 db the P-L method yields slightly better statistics than the other two methods.

VI. Conclusion

Two methods, the Prony-Lanczos method and the Power method are proposed for simple computation of approximations to a few eigenvectors and eigenvalues of a Hermitian matrix. The computational speeds of these methods were analyzed. The accuracies of the proposed methods were evaluated using covariance matrices from data consisting of two sinusoids in a gaussian noise environment. Comparisons were made with the corresponding eigenvectors and eigenvalues obtained using the LINPACK mathematical library. The suggested methods can substitute for the slower method of LINPACK if a few eigenvalues or eigenvectors are needed.
Appendix A:

In this appendix we derive the eigenvalues and eigenvectors of the covariance matrix $R$ for the case of one and two sinusoids.

One Complex Sinusoid Case:

The data sequence is modelled by:

$$ y(n) = a_1 e^{j\omega_1 n}, \quad n = 1, 2, \ldots, N $$

The covariance values of $y(n)$ are:

$$ r_{yy}(i,j) = \frac{1}{N-L} \sum_{n=L+1}^{N} y(n-i)y(n-j) \quad i, j = 1, 2, \ldots, L \quad (A.1) $$

Writing the covariance matrix $R$ explicitly in terms of the signal, we have:

$$ R = \begin{bmatrix} |a_1|^2 & |a_1|^2 e^{-j\omega_1} & |a_1|^2 e^{-j\omega_1(L-1)} \\ |a_1|^2 e^{j\omega_1} & |a_1|^2 & |a_1|^2 e^{-j\omega_1(L-2)} \\ \vdots & \vdots & \vdots \\ |a_1|^2 e^{j\omega_1(L-1)} & |a_1|^2 \end{bmatrix} \quad (A.2) $$

We can diagonalize $R$ by an orthogonal matrix $U$ resulting in the following
equation:

\[ U^* R U = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots & 0 \\ & & & \lambda_n \end{bmatrix} \quad (A.3) \]

The eigenvalues of \( R \) which occur along the diagonal elements of the above equation, satisfy the following equation:

\[ \sum_{i=1}^{L} \lambda_i = \text{tr}(R) = L |a_1|^2 \quad (A.4) \]

But the covariance matrix \( R \) is of rank = 1, since it has only one linearly independent row (or column). The rest are obtained by multiplying by a constant number \( (e^{j\omega_1 k}) \).

Then the eigenvector corresponding to the eigenvalue \( \lambda_1 = L |a_1|^2 \) is:

\[ u_1 = c_1(1 \ e^{j\omega_1} \ e^{2j\omega_1} \ e^{j(L-1)\omega_1})^T \]
since it annihilates every row of the matrix \((R - \lambda_1 I)\). The constant \(c_1\) can be determined from the fact that the matrix \(U\) is orthonormal, hence:

\[ u_1^* \cdot u_1 = 1 \]

which yields:

\[ c_1 = \frac{1}{\sqrt{L}} \]

Hence finally:

\[ \mathbf{u}_1 = \frac{1}{\sqrt{L}} \left( 1, e^{j\omega}, e^{2j\omega}, \ldots, e^{j(L-1)\omega} \right) \]

and this is a Fourier vector with fundamental frequency \(\omega_1\).

Two Complex Sinusoids Case:

The data sequences is modelled by:
\[ y(n) = a_1 e^{j\omega_1 n} + a_2 e^{j\omega_2 n} \quad n = 1,2,\ldots,N \]  
\hspace{0.5cm} (A.6)

The covariance estimates are given by the expression:

\[ r_{yy}(k,m) = \frac{1}{N-L} \sum_{n=L+1}^{N} y(n-k) y(n-m) = |a_1|^2 e^{-j\omega_1 (m-k)} + a_2 a_2^* v_1 + a_1 a_1^* v_2 \quad , k,m = 1,2,\ldots,L \]
\hspace{0.5cm} (A.7)

where:

\[ v_1 = \frac{1}{N-L} \sum_{n=L+1}^{N} e^{j\omega_1 (n-m) - j\omega_2 (n-k)} \]

\[ v_2 = \frac{1}{N-2} \sum_{n=L+1}^{N} e^{j\omega_2 (n-m) - j\omega_1 (n-k)} \]

Rewriting the matrix \( R \), we have:

\[ R = M_1 M_2 \]
\hspace{0.5cm} (A.8)
\[ (Lx2)(2xL) \]

where:
\[ M_1 = \begin{bmatrix} |a_1|^2 e_1 + x e_2 \\ |a_2|^2 e_2 + x^* e_1 \end{bmatrix} \]

\[ M_2 = \begin{bmatrix} e_1^* \\ e_2^* \end{bmatrix}^T \]

\[ e_1 = [1 \\ e^{j\omega_1} \\ e^{2j\omega_1} \\ e^{j(L-1)\omega_1}]^T \]

\[ e_2 = [1 \\ e^{j\omega_2} \\ e^{2j\omega_2} \\ e^{j(L-1)\omega_2}]^T \]

and

\[
x = \frac{a_1 a_2^*}{N-L} e^{j(\omega_2 - \omega_1)} \sum_{n=L+1}^{N} e^{-j(\omega_2 - \omega_1)n} \]

If \( u_1 \) is an eigenvector of \( R \) corresponding to eigenvalue \( \lambda_1 \), then:

\[ M_1 M_2 u_1 = \lambda_1 u_1 \tag{A.9} \]

Premultiplying by \( M_2 \), we have:

\[ M_2 M_1 M_2 u_1 = \lambda_1 M_2 u_1 \tag{A.10} \]

Thus \( \lambda_1 \) is also the eigenvalue of \( M_2 M_1 \) and the corresponding eigenvector is:

\[ v_1 = M_2 u_1 \tag{A.11} \]

Premultiplying (A.10) again by \( M_1 \)

\[ M_1 M_2 M_1 v_1 = \lambda_1 M_1 v_1 \tag{A.12} \]

and comparing (A.10) with (A.12)

\[ u_1 = M_1 v_1 \tag{A.13} \]

Thus we can find the eigenvalues and eigenvectors of \( R \) by working with the matrix \( M_2 M_1 \) which is of order 2. Hence:
\[ M_2M_1 = \begin{bmatrix} |a_1|^2 + x & |a_2|^2 + Lx^* \\ |a_1|^2 + Lx & |a_2|^2 + x^* \\ \end{bmatrix} \]  \hspace{1cm} (A.14)

where

\[ g = \sum_{n=0}^{L-1} e^{j(\omega_2 - \omega_1)n} = \sum_{n=0}^{L-1} e^{j \Delta \omega n} \]  \hspace{1cm} (A.15)

The eigenvalues \( \lambda_1 \) and \( \lambda_2 \) are found to be:

\[ \lambda_1 = \frac{1}{2}(|a_1|^2 + L|a_2|^2 + 2\text{Re}(xg)) + ((|a_1|^2 + L|a_2|^2 + 2\text{Re}(xg))^2 - 4(|a_1|^2|a_2|^2 - |x|^2))^{1/2} \]  \hspace{1cm} (A.16)

\[ \lambda_2 = \frac{1}{2}(|a_1|^2 + L|a_2|^2 + 2\text{Re}(xg)) - ((|a_1|^2 + L|a_2|^2 + 2\text{Re}(xg))^2 - 4(|a_1|^2|a_2|^2 - |x|^2))^{1/2} \]

where

\[ \text{Re}(xg) = \text{Re} \left( \frac{|a_1|^2 a_2^* \cos(-\frac{N}{2} - 2)\Delta \omega \cdot \sin \frac{N-L}{2} \Delta \omega \sin \frac{L\Delta \omega}{2}}{2(N-L) \sin^2 \frac{\Delta \omega}{2}} \right) \]  \hspace{1cm} (A.17)

Note that a column of the adjoint of \((M_2M_1 - \lambda_1 I)\) gives the eigenvector \( v_1 \) of \( M_2 M_1 \).
\[ \text{Adj}(M_2 M_1 - \lambda_1 I) = \begin{bmatrix} (L|a_1|^2 - x_2) - \lambda_1 & -|a_2|^2 g - Lx^* \\ -|a_1|^2 g - Lx & (L|a_2|^2 + x_2^* g^*) - \lambda_1 \end{bmatrix} \] (A.18)

Therefore the eigenvector \( v_1 \) is:
\[ v_1 = [v_{11} \ v_{21}]^T \]
or
\[ v_1 = [(L|a_1|^2 + x_2) - \lambda_1 - |a_1|^2 g - Lx]^T \] (A.19)

Now the eigenvector \( u_1 \) of \( R \) corresponding to \( \lambda_1 \) is:
\[ u_1 = M_1 v_1 \]
and hence,
\[ u_1 = v_{11} (|a_1|^2 e_1 + x e_2) + v_{21} (|a_2|^2 e_2 + x_2^* e_1) \] (A.20)
a linear combination of the Fourier vectors \( e_1 \) and \( e_2 \).

Similarly, the eigenvector \( u_2 \) of \( R \) corresponding to \( \lambda_2 \) is:
\[ u_2 = v_{11}' (|a_1|^2 e_1 + x e_2) + v_{21}' (|a_2|^2 e_2 + x^* e_1) \] (A.21)
where
\[ v_{11}' = [v_{11}' \ v_{21}')^T \]
and
\[ v_{11}' = (L|a_2|^2 + x_2^* g^*) - \lambda_2 \] (A.22)
\[ v_{21}' = -|a_2|^2 g - Lx^* \]

The rest of the eigenvalues of \( R \) are zero and the corresponding eigenvectors are not unique.
<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD</th>
<th>P-L</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean= 22.0357</td>
<td>22.0126</td>
<td>22.0174</td>
</tr>
<tr>
<td></td>
<td>st.dev= 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>22.0636</td>
<td>22.0423</td>
<td>22.0353</td>
</tr>
<tr>
<td></td>
<td>0.2652</td>
<td>0.2655</td>
<td>0.2642</td>
</tr>
<tr>
<td>15</td>
<td>22.0341</td>
<td>22.3182</td>
<td>22.2957</td>
</tr>
<tr>
<td></td>
<td>1.4927</td>
<td>1.4936</td>
<td>1.4892</td>
</tr>
<tr>
<td>0</td>
<td>28.8561</td>
<td>28.5285</td>
<td>28.5425</td>
</tr>
<tr>
<td></td>
<td>8.6489</td>
<td>8.7576</td>
<td>8.7477</td>
</tr>
</tbody>
</table>

Eigenvalue estimate $\lambda_1$

<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD</th>
<th>P-L</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.7107</td>
<td>0.5741</td>
<td>1.7131</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>1.7162</td>
<td>0.7504</td>
<td>1.7199</td>
</tr>
<tr>
<td></td>
<td>0.0497</td>
<td>0.3777</td>
<td>0.0498</td>
</tr>
<tr>
<td>15</td>
<td>1.8634</td>
<td>1.0327</td>
<td>1.8677</td>
</tr>
<tr>
<td></td>
<td>0.2856</td>
<td>0.5357</td>
<td>0.2884</td>
</tr>
<tr>
<td>0</td>
<td>10.5379</td>
<td>1.6797</td>
<td>7.3859</td>
</tr>
<tr>
<td></td>
<td>2.7981</td>
<td>1.2625</td>
<td>2.6301</td>
</tr>
</tbody>
</table>

Eigenvalue estimate $\lambda_2$

**TABLE I**
<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD, P-L</th>
<th>SVD, PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean = 0</td>
<td>0.4770</td>
</tr>
<tr>
<td></td>
<td>st.dev = 0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.6980 −5</td>
<td>0.5819 −4</td>
</tr>
<tr>
<td></td>
<td>0.1938 −4</td>
<td>0.7575 −4</td>
</tr>
<tr>
<td>15</td>
<td>0.2775 −4</td>
<td>0.1991 −3</td>
</tr>
<tr>
<td></td>
<td>0.5746 −4</td>
<td>0.1606 −3</td>
</tr>
<tr>
<td>0</td>
<td>0.5305 −2</td>
<td>0.5932 −2</td>
</tr>
<tr>
<td></td>
<td>0.1019 −1</td>
<td>0.1035 −1</td>
</tr>
</tbody>
</table>

First Eigenvector Distances

<table>
<thead>
<tr>
<th></th>
<th>0.3917 −4</th>
<th>0.6169 −4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.1744 −3</td>
<td>0.3283 −3</td>
</tr>
<tr>
<td></td>
<td>0.1162 −3</td>
<td>0.2850 −3</td>
</tr>
<tr>
<td>15</td>
<td>0.4618 −2</td>
<td>0.2055 −2</td>
</tr>
<tr>
<td></td>
<td>0.3682 −2</td>
<td>0.1110 −2</td>
</tr>
<tr>
<td>0</td>
<td>0.8243 −1</td>
<td>0.7146 −1</td>
</tr>
<tr>
<td></td>
<td>0.2614 −1</td>
<td>0.2938 −1</td>
</tr>
</tbody>
</table>

Second Eigenvector Distances

**TABLE II**
<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD</th>
<th>FFT</th>
<th>P−L</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞</td>
<td>0.30472 +5</td>
<td>0.14050 +4</td>
<td>0.15835 +4</td>
<td>0.27610 +4</td>
</tr>
<tr>
<td>30</td>
<td>0.24391 +5</td>
<td>0.14119 +4</td>
<td>0.15859 +4</td>
<td>0.27793 +4</td>
</tr>
<tr>
<td>15</td>
<td>0.24538 +5</td>
<td>0.14065 +4</td>
<td>0.15877 +4</td>
<td>0.27506 +4</td>
</tr>
<tr>
<td>0</td>
<td>0.25819 +5</td>
<td>0.14029 +4</td>
<td>0.15874 +4</td>
<td>0.27568 +4</td>
</tr>
</tbody>
</table>

Computational Cost
(measured in time units $t_s$, where $1 ~ t_s = 26.04166$ μsec)

**TABLE III**
<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD</th>
<th>P-L</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean= 0.5000</td>
<td>0.5000</td>
<td>0.5000</td>
</tr>
<tr>
<td></td>
<td>st.dev= 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.4999</td>
<td>0.4999</td>
<td>0.4999</td>
</tr>
<tr>
<td></td>
<td>0.0013</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
<tr>
<td>15</td>
<td>0.4961</td>
<td>0.4952</td>
<td>0.4962</td>
</tr>
<tr>
<td></td>
<td>0.0157</td>
<td>0.0137</td>
<td>0.0154</td>
</tr>
<tr>
<td>0</td>
<td>0.4331</td>
<td>0.4620</td>
<td>0.4551</td>
</tr>
<tr>
<td></td>
<td>0.1334</td>
<td>0.0898</td>
<td>0.1082</td>
</tr>
</tbody>
</table>

**Frequency Estimate** $f_1$

<table>
<thead>
<tr>
<th>SNR</th>
<th>SVD</th>
<th>P-L</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5200</td>
<td>0.5200</td>
<td>0.5200</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0.5201</td>
<td>0.5201</td>
<td>0.5201</td>
</tr>
<tr>
<td></td>
<td>0.0013</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
<tr>
<td>15</td>
<td>0.5251</td>
<td>0.5249</td>
<td>0.5251</td>
</tr>
<tr>
<td></td>
<td>0.0190</td>
<td>0.0141</td>
<td>0.0190</td>
</tr>
<tr>
<td>0</td>
<td>0.5717</td>
<td>0.5613</td>
<td>0.5642</td>
</tr>
<tr>
<td></td>
<td>0.1184</td>
<td>0.0893</td>
<td>0.0980</td>
</tr>
</tbody>
</table>

**Frequency Estimate** $f_2$

**TABLE IV**

26
References


<table>
<thead>
<tr>
<th>Copies</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics and Probability Program (Code 411(SP)) Office of Naval Research Arlington, VA 22217 3</td>
<td></td>
</tr>
<tr>
<td>Defense Technical Information Center Cameron Station Alexandria, VA 22314 12</td>
<td></td>
</tr>
<tr>
<td>Commanding Officer Office of Naval Research Eastern/Central Regional Office Attn: Director for Science Barnes Building 495 Summer Street Boston, MA 02210 1</td>
<td></td>
</tr>
<tr>
<td>Commanding Officer Office of Naval Research Western Regional Office Attn: Dr. Richard Lau 1030 East Green Street Pasadena, CA 91101 1</td>
<td></td>
</tr>
<tr>
<td>U.S. ONR Liaison Office - Far East Attn: Scientific Director APO San Francisco 96503 1</td>
<td></td>
</tr>
<tr>
<td>Applied Mathematics Laboratory David Taylor Naval Ship Research and Development Center Attn: Mr. G. H. Gleissner Bethesda, Maryland 20084 1</td>
<td></td>
</tr>
<tr>
<td>Commandant of the Marine Corps (Code AX) Attn: Dr. A. L. Slafkosky Scientific Advisor Washington, DC 20380 1</td>
<td></td>
</tr>
<tr>
<td>Navy Library National Space Technology Laboratory Attn: Navy Librarian Bay St. Louis, MS 39522 1</td>
<td></td>
</tr>
<tr>
<td>U.S. Army Research Office P.O. Box 12211 Attn: Dr. J. Chandra Research Triangle Park, NC 27706 1</td>
<td></td>
</tr>
<tr>
<td>Director National Security Agency Attn: RS1, Dr. Maar Fort Meade, MD 20755 1</td>
<td></td>
</tr>
<tr>
<td>ATAA-SL, Library U.S. Army TRADOC Systems Analysis Activity Department of the Army White Sands Missile Range, NM 88002 1</td>
<td></td>
</tr>
<tr>
<td>ARI Field Unit-USAREUR Attn: Library c/o OD/CSPER HQ USAEREUR &amp; 7th Army APO New York 09403 1</td>
<td></td>
</tr>
<tr>
<td>Library, Code 1424 Naval Postgraduate School Monterey, CA 93940 1</td>
<td></td>
</tr>
<tr>
<td>Technical Information Division Naval Research Laboratory Washington, DC 20375 1</td>
<td></td>
</tr>
<tr>
<td>OASD (I&amp;L), Pentagon Attn: Mr. Charles S. Smith Washington, DC 20301 1</td>
<td></td>
</tr>
</tbody>
</table>
Mr. F. R. Del Priori
Code 224
Operational Test and Evaluation
Force (OPTEVFOR)
Norfolk, VA 23511 1
END

FILMED

24-86

DTIC