PRECONDITIONED CONJUGATE GRADIENT IAA SPECTRAL ESTIMATION

George-Othon Glentis[1] *and Andreas Jakobsson*[2]

[1] Dept. of Science and Technology of Telecommunications, University of Peloponnese, Greece
gglentis@uop.gr
[2] Dept. of Mathematical Statistics, Lund University, Sweden
aj@maths.lth.se

ABSTRACT

In this paper, we develop superfast approximative algorithms for the computationally efficient implementation of the recent Iterative Adaptive Approach (IAA) spectral estimate. The proposed methods are based on rewriting the IAA algorithm using suitable Gohberg-Semencul representations, solving the resulting linear systems of equations using the preconditioned conjugate gradient method, where a novel preconditioning is applied using an incomplete factorization of the Toeplitz matrix. Numerical simulations illustrate the efficiency of the proposed algorithm.

1. INTRODUCTION

Computationally efficient high-resolution spectral estimation algorithms are of great importance in numerous applications. Typically, to achieve improved resolution, higher than the periodogram, one has to resort to using parametric or data-adaptive non-parametric estimation techniques. Due to their inherent robustness to model assumptions, the data-adaptive approaches are often of particular interest, and the topic has attracted increasing interest during the last decade. One such promising technique that is currently widely studied is the so-called iterative adaptive approach (IAA), recently proposed in [1], and there shown to outperform the well-known Capon and APES spectral estimation techniques (see, e.g., [2]) for data with a sparse spectrum. As a result, the technique has attracted significant interest in a variety of topics [3–9]. As noted in these papers, the IAA-based estimation techniques allow for accurate high-resolution estimates even when only a few data snapshots are available. However, this improved performance comes at the cost of a notably high computational complexity, suggesting the need for computationally efficient implementations of such estimates. In [8], we presented several computationally efficient *exact* implementations for various IAA-based estimators, exploiting efficient formulations of data dependent trigonometric polynomials as well as suitable Gohberg-Semencul (GS) representations of the estimated inverse covariance matrix.This approach allows for substantial computational savings, but given the requirement of an exact implementation, one is still limited to the original formulation of the methods. In [9], we extended on this work allowing also for non-stationary signals, formulating both exact and approximative time-recursive sliding-window implementations of the IAA estimate. As shown there, the approximative solutions, being based on variations of GS factorizations as well as a steepest descent formulation of the algorithm, allow for significant additional computational savings, while still resulting in estimates close to the exact solution. In this work, we instead formulate computationally efficient *approximate* solutions of the IAA spectral estimate. Extending the GS representations formulated in [8], we propose to solve the resulting linear set of equations using a novel Quasi-Newton (QN) preconditioned conjugate gradient (PCG) algorithm. The proposed preconditioner is motivated by the QN algorithm formulated in [10, 11], and approximate the resulting Toeplitz covariance matrix as resulting from a low-order autoregressive (AR) process. Using a GS factorization of the inverse of this approximative covariance matrix as a preconditioning to the CG algorithm will gather the eigenvalues of the matrix in a narrow range, substantially improving the conditioning of the system. Combined with the use of data dependent trigonometric polynomials, the resulting QN-PCG-IAA algorithm will, using a sufficient number of iterations, yield an exact implementation of the IAA estimate. However, the resulting algorithm will then be computationally more demanding as compared to our earlier fast IAA (FIAA) implementation proposed in [8]. Here, we are primarily interested in using the proposed reformulation to allow for approximative solutions using only a few PCG iterations, and as a result allowing for substantial computational reductions as compared to the FIAA algorithm, without more than a marginal loss of accuracy in the resulting estimates.

This work was supported in part by the Swedish Research Council and Carl Trygger's foundation.

2. AN OVERVIEW OF IAA AND FIAA

Let $\{y_n\}_{n=0}^N \in \mathbb{C}$ denote a uniformly sampled sequence of observations for which one wish to compute a spectral estimate. Form the data and frequency vectors

$$\mathbf{y}_{N+1} = \begin{bmatrix} y_0 & \dots & y_N \end{bmatrix}^T, \quad (1)$$

$$\mathbf{f}_{N+1}(\omega_k) = \begin{bmatrix} 1 & e^{j\omega_k} & \dots & e^{j\omega_k N} \end{bmatrix}^T \quad (2)$$

where $(\cdot)^T$ denotes the transpose, and where $\omega_k = 2\pi\frac{k}{K}$, $k = 0,1,\dots,K-1$, typically with $K > N+1$. Denote the power of the signal $\Phi_s(\omega_k) = |\alpha(\omega_k)|^2$, where $\alpha(\omega_k)$ is the complex-valued spectral amplitude at frequency ω_k, and let $\mathbf{R}_{N+1}$ denote an estimate of the sample covariance matrix. Then, for all frequencies of interest, the IAA spectral estimate is formed by iteratively estimating $\alpha(\omega_k)$ and $\mathbf{R}_{N+1}$, until practical convergence, as (see [1,4] for details)

$$\alpha(\omega_k) = \frac{\mathbf{f}_{N+1}^H(\omega_k)\mathbf{R}_{N+1}^{-1}\mathbf{y}_{N+1}}{\mathbf{f}_{N+1}^H(\omega_k)\mathbf{R}_{N+1}^{-1}\mathbf{f}_{N+1}(\omega_k)}, \quad (3)$$

$$\mathbf{R}_{N+1} = \sum_{k=0}^{K-1} |\alpha(\omega_k)|^2 \mathbf{f}_{N+1}(\omega_k)\mathbf{f}_{N+1}^H(\omega_k) \quad (4)$$

where $(\cdot)^H$ denotes the conjugate transpose, with $\mathbf{R}_{N+1}$ initialized to the identity matrix $\mathbf{I}_{N+1}$. Typically, no more than $m = 10-15$ iterations are needed to allow for convergence [1]. To form an efficient implementation, one may note that $\mathbf{R}_{N+1}$ is the upper left part of the circulant matrix [8]

$$\mathbf{C}_K = \mathbf{W}_K^H \mathrm{diag}\left\{ |\alpha(\omega_0)|^2, \dots, |\alpha(\omega_{K-1})|^2 \right\} \mathbf{W}_K, \quad (5)$$

where $\mathbf{W}_K$ is the Discrete Fourier Transform (DFT) matrix of size $K \times K$. Since $\mathbf{C}_K$ is a circulant matrix, with its first column denoted $\mathbf{c}_K$, it can be computed using the Inverse DFT (IDFT) as $\mathbf{c}_K = \mathbf{W}_K^H \alpha_K$, where $\alpha_K = \begin{bmatrix} |\alpha(\omega_0)|^2 & \dots & |\alpha(\omega_{K-1})|^2 \end{bmatrix}^T$. As a consequence of the embedding of $\mathbf{R}_{N+1}$ in $\mathbf{C}_K$, the first column of $\mathbf{R}_{N+1}$, denoted by $\mathbf{r}_{N+1}$, can be extracted as the $N+1$ initial

elements of $\mathbf{c}_K$. Once $\mathbf{r}_{N+1}$ has been computed, the solution of the linear system of equations that appears in (3), i.e.,

$$\mathbf{R}_{N+1}\mathbf{d}_{N+1} \triangleq \mathbf{y}_{N+1} \tag{6}$$

can be solved using the celebrated Levinson-Durbin (LD) algorithm. To do so, partition $\mathbf{R}_{N+1}$ as

$$\mathbf{R}_{N+1} = \begin{bmatrix} r_0 & \mathbf{r}_N^{fH} \\ \mathbf{r}_N^f & \mathbf{R}_N \end{bmatrix} = \begin{bmatrix} \mathbf{R}_N & \mathbf{J}_N\mathbf{r}_N^{f*} \\ \mathbf{r}_N^{fT}\mathbf{J}_N & r_0 \end{bmatrix}, \tag{7}$$

with r_0 and $\mathbf{r}_N^f$ defined accordingly, where $\mathbf{J}_N$ is the exchange matrix, which, by using the matrix inversion lemma (see, e.g., [2]), yields

$$\mathbf{R}_{N+1}^{-1} = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}_N^{-1} \end{bmatrix} + \bar{\mathbf{a}}_{N+1}\bar{\mathbf{a}}_{N+1}^H \tag{8}$$

$$= \begin{bmatrix} \mathbf{R}_N^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \mathbf{J}_{N+1}\bar{\mathbf{a}}_{N+1}^*\bar{\mathbf{a}}_{N+1}^T\mathbf{J}_{N+1} \tag{9}$$

where

$$\bar{\mathbf{a}}_{N+1} = \begin{bmatrix} 1 \\ \mathbf{a}_N \end{bmatrix} / \sqrt{\alpha_N^f}, \tag{10}$$

and

$$\mathbf{a}_N = -\mathbf{R}_N^{-1}\mathbf{r}_N^f \tag{11}$$

$$\alpha_N^f = r_0 + \mathbf{r}_N^{fH}\mathbf{a}_N \tag{12}$$

The resulting LD algorithm allows for a solution of (6) at a cost of approximately $2N^2$ operations. This complexity can be halved by instead using the GS factorization of $\mathbf{R}_{N+1}^{-1}$ for the computation of the matrix vector product that is involved in solving (6). To see this, we define the down-shifting operator

$$\mathbf{Z}_{N+1}(\nu) = \begin{bmatrix} \mathbf{0}^T & \nu \\ \mathbf{I}_N & \mathbf{0} \end{bmatrix} \tag{13}$$

and $\mathbf{C}(\xi_{N+1},\nu)$ as the ν-*circulant* matrix having ξ_{N+1} along its first column, given by

$$\mathbf{C}(\xi_{N+1},\nu) = \begin{bmatrix} \xi_0 & \nu\xi_N & \dots & \dots & \nu\xi_1 \\ \xi_1 & \xi_0 & \nu\xi_N & \dots & \nu\xi_2 \\ \vdots & \xi_1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \nu\xi_N \\ \xi_N & \dots & \dots & \xi_1 & \xi_0 \end{bmatrix}. \tag{14}$$

Clearly, $\mathbf{C}(\xi,1)$ and $\mathbf{C}(\xi,-1)$ define a *circulant* matrix and *skew-circulant* matrix, respectively. Moreover, $\mathbf{C}(\xi,0)$ coincides with the lower triangular Toeplitz matrix $\mathbf{L}(\xi_{N+1})$, formed with ξ_{N+1} along its first column. Let μ and ν be two constants. Then, it can be shown that the following lemmas hold [12]:

Lemma 1 *The inverse covariance matrix $\mathbf{R}_{N+1}^{-1}$ can be computed by the following GS-type factorization, provided that $\mu\nu \neq 1$,*

$$\mathbf{R}_{N+1}^{-1} = \frac{1}{1-\nu\mu}\sum_{i=1}^{2}\sigma_i\mathbf{C}(\mathbf{t}_{N+1}^i,\nu)\mathbf{C}^H(\mathbf{s}_{N+1}^i,\mu) \tag{15}$$

where $\sigma_1 = 1$ and $\sigma_2 = -1$, and $\mathbf{t}_{N+1}^1 \triangleq \bar{\mathbf{a}}_{N+1}$, $\mathbf{s}_{N+1}^1 = \mathbf{t}_{N+1}^1$ $\mathbf{t}_{N+1}^2 \triangleq \mathbf{Z}_{N+1}(\nu)\mathbf{J}_{N+1}\bar{\mathbf{a}}_{N+1}^*$, and $\mathbf{s}_{N+1}^2 \triangleq \mathbf{Z}_{N+1}(\mu)\mathbf{J}_{N+1}\bar{\mathbf{a}}_{N+1}^*$.

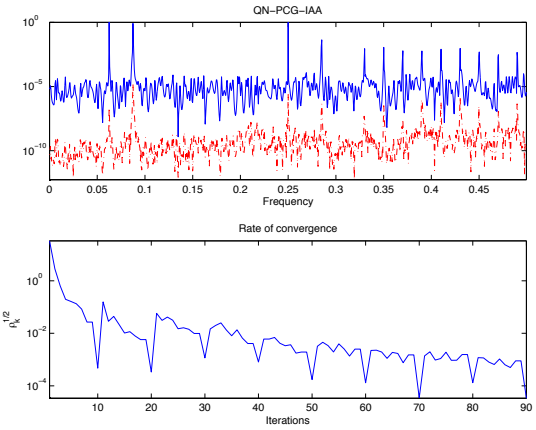

Figure 1: Performance of the QN-PCG-IAA algorithm using only $k = 8$ QN-PCG iterations followed by 2 Newton refinement iterations for $N = 511$, $M = 64$, $m = 10$, and $K = 2048$: a) estimated spectrum (solid line) and absolute error (dashed line), and b) rate of convergence.

Lemma 2 *The lower order matrix $\mathbf{R}_N^{-1}$ allows for a GS factorization of the form, provided that $\nu\mu \neq 1$,*

$$\begin{bmatrix} \mathbf{R}_N^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} = \frac{1}{1-\nu\mu}\sum_{i=1}^{2}\sigma_i\mathbf{C}(\bar{\mathbf{t}}_{N+1}^i,\nu)\mathbf{C}^H(\bar{\mathbf{s}}_{N+1}^i,\mu) \tag{16}$$

where $\bar{\mathbf{t}}_{N+1}^1 \triangleq \bar{\mathbf{a}}_{N+1}$, $\bar{\mathbf{s}}_{N+1}^1 = \bar{\mathbf{t}}_{N+1}^1$, $\bar{\mathbf{t}}_{N+1}^2 \triangleq \mathbf{J}_{N+1}\bar{\mathbf{a}}_{N+1}^*$, and $\bar{\mathbf{s}}_{N+1}^2 \triangleq \mathbf{J}_{N+1}\bar{\mathbf{a}}_{N+1}^*$.

As shown in [13], by using $\nu = \mu = 0$, one obtains the standard lower and upper triangular Toeplitz GS decomposition. If instead using $\nu = 0$ and $\mu = 1$, one obtains a lower triangular and circulant GS decomposition variant [14]. Finally, if $\nu = 1$ and $\mu = -1$, a circulant and skew-circulant GS decomposition is obtained [15]. Using (15), one may reduce the cost of solving (6) to about $\phi(N+1)$ operations, where $\phi(N)$ denotes the cost of performing an FFT of length N, providing that $\bar{\mathbf{a}}_{N+1}$, given by (10), is available. Here, the circulant and skew-circulant GS decomposition is adopted [15], i.e.

$$\mathbf{d}_{N+1} = \left(\frac{1}{2}\sum_{i=1}^{2}\sigma_i\mathbf{C}(\mathbf{t}_{N+1}^i)\mathbf{S}^H(\mathbf{s}_{N+1}^i)\right)\mathbf{y}_{N+1}. \tag{17}$$

where, using (14), $\mathbf{C}(\xi_{N+1}) \triangleq \mathbf{C}(\xi_{N+1},1)$ and $\mathbf{S}(\xi_{N+1}) \triangleq \mathbf{C}(\xi_{N+1},-1)$ are circulant and skew circulant matrices. This particular implementation of the matrix-vector product required to form, results in a lower computational complexity compared to the standard approach, bounded by $10\phi(N+1)$, where $4\phi(N+1)$ accounts for the proprocessing and $6\phi(N+1)$ accounts for the actual matrix vector multiplication [15], a fact that can be usuful in the case when the products of the same matrix with several vectors are required. Thus, the numeranator of (3), $\mathbf{f}_{N+1}^H(\omega_k)\mathbf{d}_{N+1}$, can be computed by means of fast Toeplitz vector multiplication methods at a cost proportional to $\phi(N+1)$. The denominator of (3) can similarly be handled using (see also [16])

$$\varphi(\omega_k) = \mathbf{f}_N^H(\omega_k)\mathbf{R}_{N+1}^{-1}\mathbf{f}_N(\omega_k) = \sum_{i=-N}^{N}\varphi_i e^{j\frac{2\pi}{K}ki}. \tag{18}$$

The coefficients of the polynomial defined by (18) can efficiently be computed using the GS representation of $\mathbf{R}_{N+1}^{-1}$, at a cost of

$5\phi(2(N+1))$ [17]. Finally, $\varphi(\omega_k)$, for $k=0,1\ldots,K-1$, is computed using an FFT of size K.

3. THE PROPOSED PCG-IAA IMPLEMENTATION

The CG method can be used to form an iterative solution of a set of equations such as (6). In theory, the CG algorithm provides the exact solution of (6) after $N+1$ iterations. In practice, however, this may not be so, due to the round-off errors caused by the finite precision arithmetic implementation. Moreover, the convergence rate of the algorithm may seriously be affected when $\mathbf{R}_{N+1}$ is an ill-conditioned matrix. In this case, an increased number of iterations is required to obtain an accurate solution. One way to handle this effect is to apply a preconditioning to speed up the convergence rate of the algorithm, i.e., instead of solving (6), one solves the preconditioned system (see, e.g., [18, 19])

$$\mathbf{P}_{N+1}^{-1}\mathbf{R}_{N+1}\mathbf{d}_{N+1}=\mathbf{P}_{N+1}^{-1}\mathbf{y}_{N+1}, \tag{19}$$

where $\mathbf{P}_{N+1}$ is the so-called the *preconditioning matrix*, which should be easy to construct and to invert, and have the property that the condition number $\kappa(\mathbf{P}_{N+1}^{-1}\mathbf{R}_{N+1})$ should be close to one, or, at least, it should be much lower than the condition number of the original matrix $\mathbf{R}_{N+1}$. As we are here not only interested in estimation the solution of (6), but also in computing the displacement representation of $\mathbf{R}_{N+1}^{-1}$, we present a slightly different version of the standard PCG algorithm, first computing the generators of $\mathbf{R}_{N+1}^{-1}$, then applying the PCG on (11), and, finally, estimating $\mathbf{d}_{N+1}$ using (17). The computational complexity of PCG methods depends on the type of preconditioning matrix $\mathbf{P}_N$ utilized. Using a circulant preconditioning matrix is perhaps the most popular approach adopted, since the inverse of a circulant matrix is easily computed and a variety of circulant matrices have been proposed and have been studied in terms of convergence efficiency. Regrettably, we have using numerical simulations found that the available circulant preconditioning matrices do not work well, or at all, when used in combination with IAA due to the inherent assumptions these make on the spectrum of the Toeplitz matrix, which are not fulfilled in the here examined case of data with sparse spectra. As a result, we will instead propose a novel QN based preconditioner, constructed from an incomplete factorization of $\mathbf{R}_{N+1}^{-1}$. The QN adaptive algorithm, originally proposed in [10] (see also [11, 20, 21]), provides an efficient and low complexity implementation scheme of approximate recursive least squares algorithms, by imposing a low order autoregressive approximation on the input signal of the adaptive algorithm. Motivated by [10], we are here applying an QN methodology when forming the iterative solution of Toeplitz systems using the PCG method. Given a Toeplitz matrix $\mathbf{R}_N$, the new preconditioning matrix is constructed from an incomplete factorization of the inverse $\mathbf{R}_N^{-1}$. Consider the case when the signal can be well modeled as an AR process of order $M \ll N$.

Step 1. Assuming that $\mathbf{R}_\ell$ is positive definite, at least for $\ell=M$, $\mathbf{R}_M^{-1}$ is computed using (15) as $\mathbf{R}_M^{-1}=1/2\sum_{i=1}^{2}\sigma_i\mathbf{C}(\mathbf{t}_M^i)\mathbf{S}^H(\mathbf{s}_M^i)$, where, using (10)-(12),

$$\mathbf{a}_{M-1} = -\mathbf{R}_{M-1}^{-1}\mathbf{r}_{M-1}^f \tag{20}$$

$$\alpha_{M-1}^f = r_0+\mathbf{r}_{M-1}^{fH}\mathbf{a}_{M-1} \tag{21}$$

$$\bar{\mathbf{a}}_M = \begin{bmatrix}1\\ \mathbf{a}_{M-1}\end{bmatrix}/\sqrt{\alpha_{M-1}^f} \tag{22}$$

with $\mathbf{a}_{M-1}$ and α_{M-1}^f computed either using the LD algorithm, provided that all principal minors up to size $M-1$ are positive definite, or by using another standard linear systems solution method, such as Cholesky's algorithm. This step requires $\mathcal{O}(M^2)$ operations, when the LD algorithm is utilized.

Step 2. Given $\mathbf{a}_M$, the novel QN preconditioner is constructed as the Toeplitz matrix $\mathbf{Q}_N$, whose first column is formed from the autocorrelation sequence q_ℓ, $\ell=0,1,\ldots N-1$, defined as [10]

$$q_\ell \triangleq \begin{cases} r_\ell, & \ell=0,1,\ldots,M-1\\ -\sum_{i=1}^{M}\mathbf{a}_M^{[i]}q_{\ell-i+1}, & \ell=M,\ldots,N-1\end{cases} \tag{23}$$

The special structure of the Toeplitz matrix $\mathbf{Q}_N$, being formed as the Toeplitz matrix constructed using q_ℓ, allows for the computation of the inverse matrix $\mathbf{Q}_N^{-1}$ using (8) as

$$\mathbf{Q}_N^{-1} = \begin{bmatrix}\mathbf{0} & \mathbf{0}^T\\ \mathbf{0} & \mathbf{R}_M^{-1}\end{bmatrix}+\mathbf{A}_{N,N-M-1}\mathbf{A}_{N,N-M-1}^H \tag{24}$$

where

$$\mathbf{A}_{N,N-M-1}=\underbrace{\left.\begin{bmatrix}\bar{\mathbf{a}}_M & 0 & \ldots & 0\\ 0 & \bar{\mathbf{a}}_M & \ddots & \vdots\\ \vdots & 0 & \ddots & \vdots\\ \vdots & \vdots & \ddots & 0\\ \vdots & \vdots & \ddots & \bar{\mathbf{a}}_M\\ 0 & 0 & \ddots & 0\end{bmatrix}\right\}N}_{N-M-1} \tag{25}$$

Step 3. We proceed to use $\mathbf{P}_N=\mathbf{Q}_N$ as the preconditioning matrix in the PCG scheme. In this case, $\mathbf{P}_N^{-1}=\mathbf{Q}_N^{-1}$ can efficiently be implemented using (24). Indeed, due to the special structure of $\mathbf{Q}_N^{-1}$, the required matrix vector products can be computed using FFT based schemes, at a cost of no more that $\mathcal{O}(\phi(N))+\mathcal{O}(\phi(M))$ operations. The computational complexity of the proposed PCG scheme for the computation of the generator of $\mathbf{R}_{N+1}^{-1}$ is thus $C^{[QN-PCG]}=M^2+9\phi(N)+3N+k\Big(8\phi(N)+9N+6\phi(M)+6M\Big)$, where the first term corresponds to the initialization and the second term corresponds to the repetitive computations imposed by the method. We proceed to use the above QN-PCG algorithm to estimate the displacement representation of $\mathbf{R}_{N+1}^{-1}$, i.e., the estimation of $\bar{\mathbf{a}}_{N+1}$, as defined in (10), by means of computing $\mathbf{a}_N$ in (11) using the PCG algorithm tabulated along with the QN preconditioner in (24). We term the resulting scheme the *Quasi-Netwon PCG IAA* (QN-PCG-IAA) algorithm and it is tabulated in Table 1. The initialization of the QN-PCG algorithm can be done either by setting

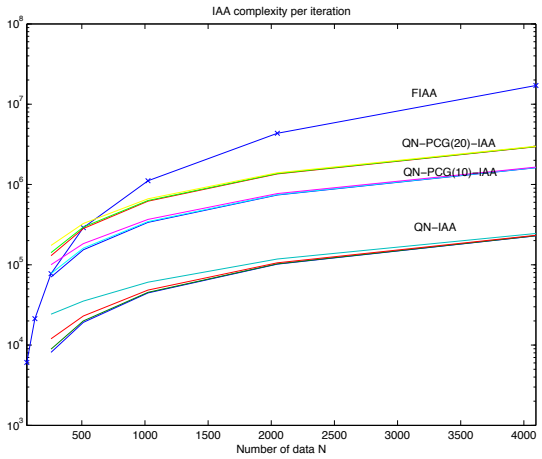

Figure 2: The computational complexity of the discussed algorithms for different values predictor sizes, M.

$\mathbf{a}_N^{\text{INI}} = \mathbf{0}_N$, or more efficiently, by using the estimate of $\mathbf{a}_N$ obtained in the previous IAA iteration. This stems from the fact that, upon convergence, $\mathbf{R}_{N+1}$ will not change too much in between successive IAA iterations. Finally, better performance can be achieved by introducing a Newton-based refinement scheme following the QN-PCG solution, with (see also [22])

$$\mathbf{e}_N^f = -\mathbf{r}_N^f - \mathbf{R}_N \mathbf{a}_N \quad (26)$$

$$\mathbf{a}_N = \mathbf{a}_N + \mathbf{R}_N^{-1} \mathbf{e}_N^f \quad (27)$$

iterated until practical convergence. Usually, only very few steps are required for the convergence of the algorithm, provided that a good guess for $\mathbf{a}_N$ is used for initialization, and an accurate estimate of $\mathbf{R}_N^{-1}$ is available. Here, the output of the QN-PCG is used for initialization and $\mathbf{R}_N^{-1}$ is re-estimated at each step from its displacement representation, using (16). Thus, we obtain

$$\begin{bmatrix} \mathbf{e}_N^f \\ \times \end{bmatrix} = -\begin{bmatrix} \mathbf{r}_N^f \\ \times \end{bmatrix} - \mathbf{R}_{N+1} \begin{bmatrix} \mathbf{a}_N \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{a}_N \\ \times \end{bmatrix} = \begin{bmatrix} \mathbf{a}_N \\ \times \end{bmatrix} + \frac{1}{2} \sum_{i=1}^{2} \sigma_i \mathbf{C}(\bar{\mathbf{t}}_{N+1}^i) \mathbf{S}^H(\bar{\mathbf{s}}_{N+1}^i) \begin{bmatrix} \mathbf{e}_N^f \\ 0 \end{bmatrix}.$$

As an alternative, one may form an approximative IAA algorithm by instead of computing $\mathbf{R}_{N+1}^{-1}$ use the proposed matrix $\mathbf{Q}_{N+1}^{-1}$, as defined in (24), in place of $\mathbf{R}_{N+1}^{-1}$ directly in the IAA algorithm[1]. In this way, an approximate IAA algorithm is formed by iteratively estimating $\alpha(\omega_k)$ and $\mathbf{Q}_{N+1}$,

$$\alpha(\omega_k) = \frac{\mathbf{f}_{N+1}^H(\omega_k) \mathbf{Q}_{N+1}^{-1} \mathbf{y}_{N+1}}{\mathbf{f}_{N+1}^H(\omega_k) \mathbf{Q}_{N+1}^{-1} \mathbf{f}_{N+1}(\omega_k)}, \quad (28)$$

$$\mathbf{R}_M = \sum_{k=0}^{K-1} |\alpha(\omega_k)|^2 \mathbf{f}_M(\omega_k) \mathbf{f}_M^H(\omega_k) \quad (29)$$

$$\mathbf{Q}_{N+1}^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}_M^{-1} \end{bmatrix} + \mathbf{A}_{N+1,N-M} \mathbf{A}_{N+1,N-M}^H \quad (30)$$

until practical convergence, for $k = 0, 1, \ldots, K-1$. Using $M \ll N$, a significant computation reduction can then be achieved, at the expense of a possible degradation in the quality of the spectrum estimate. Since $M \ll N$, the Levinson-Durbin algorithm is employed for the computation of the generators of the inverse matrix $\mathbf{R}_M$. Moreover, the denominator of (28) can be computed efficiently, since $\varphi(\omega_k) \triangleq \mathbf{f}_{N+1}^H(\omega_k) \mathbf{Q}_{N+1}^{-1} \mathbf{f}_{N+1}(\omega_k)$ equals to $\varphi(\omega_k) = \hat{\varphi}(\omega_k) + (N-M)|\mathbf{f}_M^H \bar{\mathbf{a}}_M|^2$, where $\hat{\varphi}(\omega_k) \triangleq \mathbf{f}_M^H(\omega_k) \mathbf{R}_M^{-1} \mathbf{f}_M(\omega_k)$. We denote the resulting approximative algorithm the QN-IAA algorithm.

4. PERFORMANCE EVALUATION

To be competitive with the FIAA algorithm, the proposed QN-PCG-IAA method should be able to provide fairly accurate results using only a few PCG iterations. For simplicity, the data set described in [24] is used. The performance of the proposed scheme when $M = 64$, the number of QN-PCG iterations is $k = 8$, followed by $k_1 = 2$ Newton refinement iterations, is illustrated in Figure 1, where the norm error, over all frequencies, in this case equals 1.210^{-5}, together with the evolution of the convergence indicator of the PCG algorithm, ρ_K. As can be seen from the figure, the algorithm offers an almost as good estimate in this case. The QN-IAA algorithm will also perform relatively well, even for small values of M (see [12] for details). The computational complexity of the proposed methods is illustrated in Figure 2. The curves designated by QN-PCG(k)-IAA, correspond to the cost of the proposed QN-PCG-IAA algorithm, using k PCG iterations, or more precisely $k-2$ PCG and 2 Newton refinement iterations, for $M = 32$, $M = 64$, and $M = 128$. Clearly, the algorithm allows for up to an order of magnitude improvement as compared to the complexity of the FIAA algorithm, where QN-PCG(10)-IAA becomes more efficient than the FIAA for $N > 256$ and QN-PCG(20)-IAA for $N > 512$, and where the influence of the size of M on the overall complexity is noticeable only for relatively small values of N. Finally, as is also illustrated in the figure, the proposed approximate QN-IAA algorithm is seen to offers a significant further cost reduction (up to two orders of magnitude) for $M = 16$, $M = 32$, $M = 64$ and $M = 128$.

It is worth noting that the proposed preconditioning method can also be used for the solution of Hermitian positive definite Toeplitz equations. Let the elements of the first column of a Toeplitz matrix, $\mathbf{T}$, be the Fourier coefficients of the generating function $f(\theta)$, i.e., (see also [18, 19])

$$t_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-jn\theta} d\theta, \quad n = 0, 1, \ldots, N-1 \quad (31)$$

The effectiveness of the proposed preconditioner can then be examined in the context of the iterative Toeplitz solvers, using the generating functions $f_1(\theta) = \theta^4 + 1$, $f_2(\theta) = |\theta|^3 + 1$, $f_3(\theta) = \theta^4$, and $f_4(\theta) = \theta^4(\pi^2 - \theta^2)$ to generate Toeplitz matrices of size $N = 1024$, here designated by $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$, and $\mathbf{T}_4$, respectively. Moreover, a fifth matrix, $\mathbf{T}_5$, is considered, as resulting from the $m = 10$ iteration of the standard FIAA algorithm, $\mathbf{R}_N$. The right hand side vector of the Toeplitz linear system to be solved is set equal to ones, i.e., $\mathbf{b} = [1 \ldots 1]^T$. The PCG algorithms are all initialized by setting the initial guess of the solution sought equal to zero, while iterations are performed until a desired threshold of the solution error norm, $\sqrt{\rho}$, is reached, usually expressed as $tol \cdot \sqrt{\mathbf{b}^T \mathbf{b}}$, where $tol = 10^{-7}$. The number of iterations required by each PCG algorithm to reach the desired error threshold are tabulated in Table 2, where † indicates that the PCG iterations did not converge to the given tol after a maximum number of iterations, here set to $\kappa_{max} = 4000$. Apart form the standard CG algorithm (without preconditioning), here denoted I, we examine PCG using various circulant preconditioners are examined, namely T. Chan's preconditioner, denoted C_F, and the preconditioners resulting from the use of the r-th order generalized Jackson kernels, denoted $K_{\kappa,2r}$, $r = 2, 3, 4$, which are the preconditioners considered to be the more efficient in the case of ill-conditioned Toeplitz systems [18, 19]. Finally, the proposed preconditioning method is considered, denoted $QN(M)$, for several values of M. As is clear from the table, the proposed preconditioner offers substantial improvements as compared to these typically used preconditioners, especially for large values of M.

[1] We note that a similar extrapolation of the data covariance matrix using time domain data processing has been used in [23].

REFERENCES

[1] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source Localization and Sensing: A Nonparametric Iterative Approach Based on Weighted Least Squares," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 425–443, January 2010.

[2] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

[3] W. Roberts, P. Stoica, J. Li, T. Yardibi, and F. A. Sadjadi, "Iterative Adaptive Approaches to MIMO Radar Imaging," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 5–20, Feb 2010.

[4] P. Stoica, Jian Li, and Jun Ling, "Missing Data Recovery Via a Nonparametric Iterative Adaptive Approach," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 241–244, April 2009.

[5] P. Stoica, Jian Li, and Hao He, "Spectral Analysis of Nonuniformly Sampled Data: A New Approach Versus the Peri-

Table 1: The Fast QN-PCG-IAA algorithm

For iteration $\ell = 1,\ldots,m$ % BEGIN IAA ITERATIONS

$$\begin{aligned}
\mathbf{R}_M\mathbf{a}_M &= -\mathbf{r}_M^f \\
\mathbf{R}_{M+1}^{-1} &= \frac{1}{2}\sum_{i=1}^{2}\sigma_i\mathbf{C}(\mathbf{t}_{M+1}^i)\mathbf{S}^H(\mathbf{s}_{M+1}^i) \\
\mathbf{Q}_N^{-1} &= \begin{bmatrix}\mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}_M^{-1}\end{bmatrix} + \mathbf{A}_{N,N-M-1}\mathbf{A}_{N,N-M-1}^H
\end{aligned}$$

For iteration $\kappa = 1,\ldots k$ % BEGIN QN-PCG ITERATION

$$\begin{aligned}
\mathbf{z}_N &= \mathbf{Q}_N^{-1}\mathbf{e}_N^f \\
\tau_{k-1} &= \mathbf{z}_N^H\mathbf{e}_N^f \\
\beta &= 0, \quad (\text{if } \kappa = 1) \\
\mathbf{q}_N &= \mathbf{0}, \quad (\text{if } \kappa = 1) \\
\beta &= \tau_{\kappa-1}/\tau_{\kappa-2}, \quad (\text{if } \kappa \neq 1) \\
\mathbf{q}_N &= \mathbf{z}_N + \beta\mathbf{q}_N, \quad (\text{if } \kappa \neq 1) \\
\mathbf{w}_N &= \mathbf{R}_N\mathbf{q}_N \\
\alpha &= \tau_{\kappa-1}/(\mathbf{q}_N^H\mathbf{w}_N) \\
\mathbf{a}_N &= \mathbf{a}_N + \alpha\mathbf{q}_N \\
\mathbf{e}_N^f &= \mathbf{e}_N^f - \alpha\mathbf{w}_N \\
\rho_\kappa &= \mathbf{e}_N^{fH}\mathbf{e}_N^f
\end{aligned}$$

End κ % END QN-PCG ITERATIONS

$$\begin{aligned}
\mathbf{R}_{N+1}^{-1} &= \frac{1}{2}\sum_{i=1}^{2}\sigma_i\mathbf{C}(\mathbf{t}_{N+1}^i)\mathbf{S}^H(\mathbf{s}_{N+1}^i) \\
\mathbf{d}_{N+1} &= [\mathbf{R}_{N+1}]^{-1}\mathbf{y}_{N+1} \\
\psi(\omega_k) &= \mathbf{f}_L^H(\omega_k)\mathbf{d}_{N+1} \\
\varphi(\omega_k) &= \sum_{i=-N}^{N}\varphi_i e^{j\frac{2\pi k}{K}i} \\
\alpha(\omega_k) &= \frac{\psi(\omega_k)}{\varphi(\omega_k)} \\
\begin{bmatrix} r_0 \\ \mathbf{r}_N^f \\ \times \end{bmatrix}^T &= \mathbf{W}_K^H\left[|\alpha(\omega_0)|^2\ldots|\alpha(\omega_{K-1})|^2\right]^T
\end{aligned}$$

End ℓ % END IAA ITERATIONS

Table 2: Number of PCG iterations for various Toeplitz matrices and preconditioning matrices, for $N = 1024$ and $tol = 10^{-7}$

	$\mathbf{T}_1$	$\mathbf{T_2}$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$
I	71	430	†	†	730
C_F	5	10	587	350	546
$K_{\kappa,4}$	5	6	24	25	903
$K_{\kappa,6}$	5	6	23	24	937
$K_{\kappa,8}$	5	6	22	25	1022
$QN(8)$	3	17	902	817	3170
$QN(16)$	3	5	280	290	1755
$QN(32)$	2	3	94	95	553
$QN(64)$	2	2	38	40	167
$QN(128)$	1	2	21	23	111
$QN(256)$	1	1	13	15	61

odogram," *IEEE Trans. on Signal Processing*, vol. 57, no. 3, pp. 843–858, March 2009.

[6] N. R. Butt and A. Jakobsson, "Coherence Spectrum Estimation From Nonuniformly Sampled Sequences," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 339–342, April 2010.

[7] E. Gudmundson, P. Stoica, J. Li, A. Jakobsson, M. D. Rowe, J. A. S. Smith, and J. Ling, "Spectral Estimation of Irregularly Sampled Exponentially Decaying Signals with Applications to RF Spectroscopy," *Journal of Magnetic Resonance*, vol. 203, no. 1, pp. 167–176, March 2010.

[8] G.-O. Glentis and A. Jakobsson, "Efficient Implementation of Iterative Adaptive Approach Spectral Estimation Techniques," *IEEE Trans. Signal Processing*, to appear.

[9] G.-O. Glentis and A. Jakobsson, "Time-Recursive IAA Spectral Estimation," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 111–114, Feb. 2011.

[10] G. V. Moustakides and S. Theodoridis, "Fast Newton Transversal Filters - A New Class of Adaptive Estimation Algorithms," *IEEE Trans. Signal Processing*, vol. 39, no. 10, pp. 2184–2193, Oct. 1991.

[11] B. Farhang-Boroujeny, "Fast LMS/Newton algorithms based on autoregressive modeling and their application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 45, no. 8, pp. 1987–2000, August 1997.

[12] G.-O. Glentis and A. Jakobsson, "Superfast Approximative Implementation of the IAA Spectral Estimate," *Submitted to IEEE Trans. Signal Processing*.

[13] I. C. Gohberg and A. A. Semencul, "On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs," *Mat. Issled.*, vol. 2, pp. 201–233, 1972, (in Russian).

[14] G. Ammar and P. Gader, "A Variant of the Gohberg–Semencul Formula Involving Circulant Matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 12, no. 3, pp. 534–540, July 1991.

[15] I. Gohberg and V. Olshevksy, "Complexity of multiplication with vectors for structured matrices," *Linear Algebra Appl.*, vol. 202, pp. 163–192, 1994.

[16] H. Li, P. Stoica, and J. Li, "Capon Estimation of Covariance Sequences," *Circuits, Systems, and Signal Processing*, vol. 17, no. 1, pp. 29–49, 1998.

[17] G.-O. Glentis, "A Fast Algorithm for APES and Capon Spectral Estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4207–4220, Sept. 2008.

[18] M. Ng, *Iterative methods for Toeplitz systems*, Oxford University Press, 2004.

[19] R. Chan and X. Jin, *An introduction to iterative Toeplitz solvers*, SIAM, 2007.

[20] A. Gilloire, T. Petillon, and S. Theodoridis, "The fast Newton transversal filter: an efficient scheme for acoustic echo cancellation in mobile radio," *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 509–518, March 1994.

[21] G. Moustakides, S. Theodoridis, and K. Berberidis, "A fast Newton multichannel algorithm for decision feedback equalization," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 327–331, Jan. 1995.

[22] V. Pan, *Structured matrices and polynomials: Unified superfast algorithms*, Birkhauser Boston, 2001.

[23] C. Xu and S. Kay, "A Wide Dynamic Range Multichannel Spectral Estimator," unpublished work.

[24] P. Stoica, A. Jakobsson, and J. Li, "Matched-Filterbank Interpretation of Some Spectral Estimators," *Signal Processing*, vol. 66, no. 1, pp. 45–59, April 1998.