On the Relationship Between Multi-Stage Nested Wiener Filter and Conjugate-Gradient Based Optimization

Michael D. Zoltowski[1], Guido Dietl[1], and J. Scott Goldstein[2]

[1]School of Electrical Engineering
Purdue University, West Lafayette, IN 47907-1285
e-mail: {mikedz,dietl}@ecn.purdue.edu

[2] Adaptive Signal Exploitation, SAIC
4001 Fairfax Drive, Arlington, VA 22203
e-mail: JAY.S.GOLDSTEIN@saic.com

ABSTRACT

The *Multi-Stage Nested Wiener Filter* (MSNWF) and the *Conjugate Gradient* (CG) *method* yield the solution of the *Wiener-Hopf equation* in the *Krylov subspace* of the covariance matrix of the observation and the crosscorrelation vector between the observation and the desired signal. Using the *Lanczos algorithm* instead of the *Arnoldi algorithm* for the MSNWF simplifies the computation of the Krylov subspace basis.

In this paper, we show the relationship between the CG method and the Lanczos based MSNWF and finally derive that the MSNWF may be mathematically transformed into the CG algorithm. Consequently, we present a new implementation of the MSNWF where the weight vector and the *Mean Square Error* (MSE) is directly updated as each new stage is added.

The new algorithm is applied to an *Enhanced Data rates for GSM Evolution* (EDGE) system where it linearily equalizes the received signal. Simulation results demonstrate the ability of the MSNWF to reduce the receiver complexity while maintaining the same level of system performance.

Keywords: adaptive filtering, conjugate gradients, reduced-rank equalization, space-time processing, wireless communications, EDGE, multipath propagation.

1 Introduction

One general problem in estimation theory is to regain the unknown signal $d_0[n]$ from an observation signal $\boldsymbol{x}_0[n]$. The *Wiener filter* (WF) solves this problem in the *Minimum Mean Square Error* (MMSE) sense exploiting only second order statistics. Thus, the WF is easy to implement and therefore used in many applications. However, the resulting filter needs the inverse of the covariance matrix of the observation. This means observations $\boldsymbol{x}_0[n]$ of high dimensionality imply high computational complexity.

The *Principal Component* (PC) *method* [1] was the first approach to reduce the estimation problem. A pre-filter matrix composed of the eigenvectors belonging to the principal eigenvalues of the covariance matrix of the observation is applied to the observation signal to get a transformed signal of lower dimensionality. An alternative approach is the *Cross-Spectral* (CS) *metric* introduced by Goldstein et. al. [2], where the columns of the pre-filter matrix are the eigenvectors which belong to the largest CS metric. Thus, it considers not only the statistics of the observation signal but also the relation to the desired signal. More recently, Goldstein et. al. developed the *Multi-Stage Nested Wiener Filter* (MSNWF) [3] which approximates the WF in reduced space without the computation of eigenvectors. The MSNWF shows that dimensionality reduction of the observation signal based on eigenvectors is generally suboptimal.

Honig et. al. [4] observed that the MSNWF can be seen as the solution of the *Wiener-Hopf equation* in the *Krylov subspace* of the covariance matrix of the observation and the crosscorrelation vector between the observation and the desired signal. Finally, in [5] it is shown that the *Arnoldi algorithm* can be replaced by the *Lanczos algorithm* to generate the basis vectors for the Krylov subspace since the covariance matrix is Hermitian. The resulting algorithm is an order-recursive version of the MSNWF which recursively updates the filter and the *Mean Square Error* (MSE) at each stage.

Our contribution is to derive the relationship between the *Conjugate Gradient* (CG) *method* and the Lanczos based implementation of the MSNWF. The CG method was originally introduced by Hestenes and Stiefel [6] in the year 1952 to solve a system of linear equations. It searchs for an approximate solution in the Krylov subspace similar to the Lanczos based MSNWF. We transform the equations of the Lanczos based MSNWF algorithm to yield a formulation of the CG algorithm, and finally present a new implementation of the MSNWF where the weight vector and the *Mean Square Error* (MSE) between the estimated and desired signal is directly updated as each new stage is added.

In the next section, we briefly derive the Lanczos based MSNWF. Before we show the relationship between the considered algorithms in Section 4, we review the basics of the CG algorithm in Section 2. Finally, we present a new formulation of the MSNWF algorithm in Section 5 and apply it to an EDGE system in Section 6.

Throughout the paper the covariance matrix of a vector $\boldsymbol{x}[n]$ is denoted by $\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}[n]\boldsymbol{x}^{\mathrm{H}}[n]\right\}$, the cross-correlation of a vector $\boldsymbol{x}[n]$ and a scalar $d[n]$ is $\boldsymbol{r}_{\boldsymbol{x},d} = \mathrm{E}\{\boldsymbol{x}[n]d^*[n]\}$, and the variance of a scalar $d[n]$ is $\sigma_d^2 = \mathrm{E}\left\{|d[n]|^2\right\}$.

2 Lanczos Based MSNWF

Applying the linear filter $\boldsymbol{w} \in \mathbb{C}^N$ to the observation signal $\boldsymbol{x}_0[n] \in \mathbb{C}^N$ leads to the estimate $\hat{d}_0[n] = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}_0[n]$ of the desired signal $d_0[n] \in \mathbb{C}$. The power of the estimation error $\varepsilon_0[n] = d_0[n] - \hat{d}_0[n] = d_0[n] - \boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}_0[n]$ is the *mean square error*

$$\mathrm{MSE}_0 = \mathrm{E}\left\{|\varepsilon_0[n]|^2\right\} = \sigma_{d_0}^2 - \boldsymbol{w}^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}^{\mathrm{H}}\boldsymbol{w} + \boldsymbol{w}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}. \tag{1}$$

The *Wiener filter* (WF) minimizes MSE_0, thus, we get the design criterion

$$\boldsymbol{w}_0 = \arg\min_{\boldsymbol{w}} \mathrm{MSE}_0, \tag{2}$$

which leads to the *Wiener-Hopf equation*

$$\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}_0 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0}. \tag{3}$$

Its solution, the WF $\boldsymbol{w}_0$, achieves the *minimum mean square error*

$$\mathrm{MMSE}_0 = \sigma_{d_0}^2 - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}^{-1}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}. \tag{4}$$

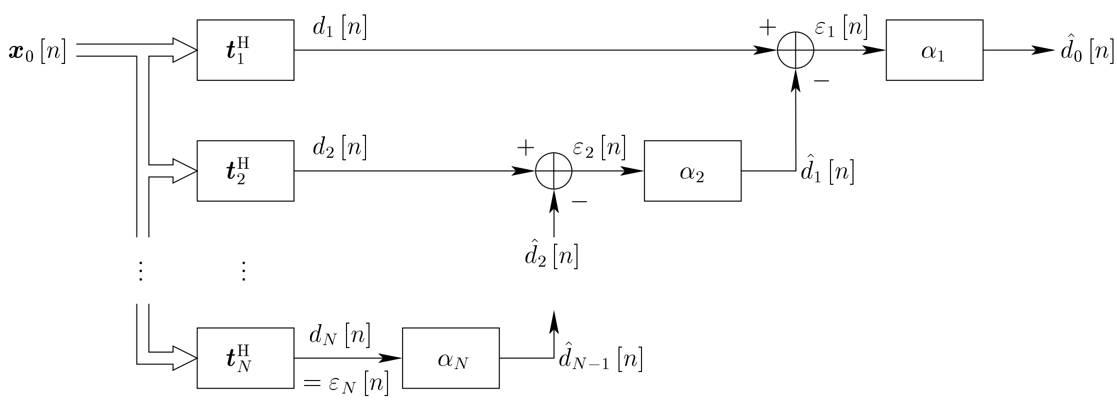

Figure 1: MSNWF as a Filter Bank

Figure 1 shows the block circuit diagram of the *Multi-Stage Nested Wiener Filter* (MSNWF) as a filter bank that solves Equation (3). Its derivation [3, 5] is not shown in this paper due to space limitations. The basis vectors for the observation space, $\boldsymbol{t}_i \in \mathbb{C}^N$, $i \in \{1, \ldots, N\}$, are defined as

$$\boldsymbol{t}_i = \left(\prod_{k=1}^{i-1} \boldsymbol{B}_k^{\mathrm{H}}\right)\boldsymbol{h}_i. \tag{5}$$

$\boldsymbol{h}_i \in \mathbb{C}^{N-(i-1)}$ is the normalized matched filter

$$\boldsymbol{h}_i = \frac{\boldsymbol{r}_{\boldsymbol{x}_{i-1},d_{i-1}}}{\left\|\boldsymbol{r}_{\boldsymbol{x}_{i-1},d_{i-1}}\right\|_2}, \tag{6}$$

which maximizes the real part of the crosscorrelation between the new desired signal $d_i[n] = \boldsymbol{h}_i^{\mathrm{H}}\boldsymbol{x}_{i-1}[n] \in \mathbb{C}$ at stage i and the desired signal $d_{i-1}[n]$ at the previous stage $i-1$ of the MSNWF. $\boldsymbol{B}_i \in \mathbb{C}^{(N-i)\times(N-(i-1))}$ for $i \in \{1,\ldots,N-1\}$ are the blocking matrices satisfying the following equation

$$\boldsymbol{B}_i\boldsymbol{h}_i = 0 \quad \Leftrightarrow \quad \operatorname{span}\left(\boldsymbol{B}_i^{\mathrm{H}}\right) = \operatorname{null}\left(\boldsymbol{h}_i^{\mathrm{H}}\right). \tag{7}$$

Thus, the rows of $\boldsymbol{B}_i$ are chosen to be orthogonal to the matched filter $\boldsymbol{h}_i$ and the new observation signal $\boldsymbol{x}_i[n] = \boldsymbol{B}_i\boldsymbol{x}_{i-1}[n] \in \mathbb{C}^{N-i}$ at stage i is minimum correlated to the desired signal $d_{i-1}[n]$ at stage $i-1$. Finally, $\alpha_i \in \mathbb{R}$ are scalar WF which estimate the desired signal $d_{i-1}[n]$ from the error $\varepsilon_i[n]$ of the new desired signal $d_i[n]$ to its estimate $\hat{d}_i[n]$, i.e. $\varepsilon_i[n] = d_i[n] - \hat{d}_i[n]$. Thus, it holds for $i \in \{1,\ldots,N\}$

$$\alpha_i = \sigma_{\varepsilon_i}^{-2} r_{\varepsilon_i,d_{i-1}}, \tag{8}$$

where $r_{\varepsilon_i,d_{i-1}} = \mathrm{E}\left\{\varepsilon_i[n]\, d_{i-1}^*[n]\right\} = \left\|\boldsymbol{r}_{\boldsymbol{x}_{i-1},d_{i-1}}\right\|_2$ [3] and $\varepsilon_N[n] = d_N[n]$.

Note, that the pre-filtered observation vector

$$\boldsymbol{d}[n] = \begin{bmatrix} d_1[n] & \cdots & d_N[n] \end{bmatrix}^{\mathrm{T}} \tag{9}$$

has a real tri-diagonal covariance matrix $\boldsymbol{R}_{\boldsymbol{d}}$ [3, 5] because the matched filters $\boldsymbol{h}_i$ and $\boldsymbol{h}_{i+1}$ of two arbitrary adjoining stages of the MSNWF are designed to maximize the real part of the correlation between $d_i[n]$ and $d_{i-1}[n]$, and between $d_{i+1}[n]$ and $d_i[n]$, respectively, whereas the blocking matrix $\boldsymbol{B}_i$ ensures that $d_{i+1}[n]$ is uncorrelated with $d_{i-1}[n]$.

To obtain a reduced rank MSNWF we only use the first D basis vectors $\boldsymbol{t}_i$, $i \in \{1,\ldots,D\}$, to build the pre-filter matrix

$$\boldsymbol{T}^{(D)} = \begin{bmatrix} \boldsymbol{t}_1 & \ldots & \boldsymbol{t}_D \end{bmatrix} \in \mathbb{C}^{N\times D}, \tag{10}$$

which yields the observation vector $\boldsymbol{d}^{(D)}[n] \in \mathbb{C}^D$ of reduced length

$$\boldsymbol{d}^{(D)}[n] = \boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{x}_0[n]. \tag{11}$$

Its tri-diagonal covariance matrix $\boldsymbol{R}_{\boldsymbol{d}}^{(D)} \in \mathbb{R}^{D\times D}$ and its crosscorrelation vector with the desired signal $d_0[n]$, $\boldsymbol{r}_{\boldsymbol{d},d_0}^{(D)} \in \mathbb{R}^D$, may be written as

$$\boldsymbol{R}_{\boldsymbol{d}}^{(D)} = \boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)} = \left[\begin{array}{cc|c} \boldsymbol{T}^{(D-1),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D-1)} & & \begin{matrix}\boldsymbol{0}\\ r_{D-1,D}\end{matrix} \\ \hline \boldsymbol{0}^{\mathrm{T}} & r_{D-1,D} & r_{D,D} \end{array}\right], \tag{12}$$

$$\boldsymbol{r}_{\boldsymbol{d},d_0}^{(D)} = \boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \begin{bmatrix} \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2 \\ \boldsymbol{0} \end{bmatrix}. \tag{13}$$

The new entries of $\boldsymbol{R}_{\boldsymbol{d}}^{(D)}$ are simply

$$r_{D-1,D} = \boldsymbol{t}_{D-1}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_D \quad \text{and} \quad r_{D,D} = \boldsymbol{t}_D^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_D \tag{14}$$

if we use the knowledge of $\boldsymbol{R}_{\boldsymbol{d}}^{(D-1)}$. It can be proved that the elements of $\boldsymbol{R}_{\boldsymbol{d}}^{(D)}$ are real-valued but the proof is omitted due to space limitations. Next, we have to compute the WF $\boldsymbol{w}_{\boldsymbol{d}}^{(D)} \in \mathbb{R}^D$ which estimates $d_0[n]$ from $\boldsymbol{d}^{(D)}[n]$, i.e.

$$\boldsymbol{w}_{\boldsymbol{d}}^{(D)} = \left(\boldsymbol{R}_{\boldsymbol{d}}^{(D)}\right)^{-1}\boldsymbol{r}_{\boldsymbol{d},d_0}^{(D)} = \left(\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)}\right)^{-1}\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}, \tag{15}$$

to get finally the rank D approximation of the WF

$$\boldsymbol{w}_0^{(D)} = \boldsymbol{T}^{(D)}\boldsymbol{w}_{\boldsymbol{d}}^{(D)} = \boldsymbol{T}^{(D)}\left(\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)}\right)^{-1}\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}, \tag{16}$$

which yields the mean square error

$$\mathrm{MSE}^{(D)} = \sigma_{d_0}^2 - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}^{\mathrm{H}}\boldsymbol{T}^{(D)}\left(\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)}\right)^{-1}\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}. \tag{17}$$

Note that the rank D MSNWF is equivalent [4, 5] to solving the Wiener-Hopf equation in the D-dimensional *Krylov subspace* $\mathcal{K}^{(D)} = \operatorname{span}\left(\left[\begin{array}{cccc} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} & \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} & \cdots & \boldsymbol{R}_{\boldsymbol{x}_0}^{(D-1)}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\end{array}\right]\right)$ if the filters $\boldsymbol{t}_i$ for $i \in \{1,\ldots,D\}$ are mutually orthogonal. Consequently, the columns of the pre-filter matrix $\boldsymbol{T}^{(D)}$ are basis vectors for $\mathcal{K}^{(D)}$ and may be computed by the *Lanczos algorithm* since we assume that $\|\boldsymbol{t}_i\|_2 = 1$ for all $i \in \{1,\ldots,D\}$ and remember that the covariance matrix $\boldsymbol{R}_{\boldsymbol{x}_0}$ is Hermitian. Thus, we get the recursion formula

$$\begin{aligned}\boldsymbol{t}_i &= \frac{\boldsymbol{P}_{i-1}\boldsymbol{P}_{i-2}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}}{\|\boldsymbol{P}_{i-1}\boldsymbol{P}_{i-2}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\|_2} \\ &= \frac{\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1} - \boldsymbol{t}_{i-2}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\boldsymbol{t}_{i-2} - \boldsymbol{t}_{i-1}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\boldsymbol{t}_{i-1}}{\left\|\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1} - \boldsymbol{t}_{i-2}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\boldsymbol{t}_{i-2} - \boldsymbol{t}_{i-1}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\boldsymbol{t}_{i-1}\right\|_2}\end{aligned} \tag{18}$$

with the projectors $\boldsymbol{P}_k = \mathbf{1} - \boldsymbol{t}_k\boldsymbol{t}_k^{\mathrm{H}}$, $k \in \{i-1, i-2\}$, onto the space orthogonal to $\boldsymbol{t}_k$. $\mathbf{1}$ denotes the $N \times N$ identity matrix.

To finish the derivation of the Lanczos implementation of the MSNWF, define

$$\boldsymbol{C}^{(D)} = \boldsymbol{R}_{\boldsymbol{d}}^{(D),-1} = \left[\begin{array}{ccc}\boldsymbol{c}_1^{(D)} & \cdots & \boldsymbol{c}_D^{(D)}\end{array}\right] \tag{19}$$

and use the inversion lemma for partitioned matrices to get

$$\boldsymbol{C}^{(D)} = \left[\begin{array}{cc}\boldsymbol{C}^{(D-1)} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0\end{array}\right] + \beta_D^{-1}\boldsymbol{b}^{(D)}\boldsymbol{b}^{(D),\mathrm{H}}, \tag{20}$$

where

$$\boldsymbol{b}^{(D)} = \left[\begin{array}{c}-\boldsymbol{C}^{(D-1)}\left[\begin{array}{c}\mathbf{0} \\ r_{D-1,D}\end{array}\right] \\ 1\end{array}\right] = \left[\begin{array}{c}-r_{D-1,D}\boldsymbol{c}_{D-1}^{(D-1)} \\ 1\end{array}\right] \tag{21}$$

and

$$\beta_D = r_{D,D} - \left[\begin{array}{cc}\mathbf{0}^{\mathrm{T}} & r_{D-1,D}\end{array}\right]\boldsymbol{C}^{(D-1)}\left[\begin{array}{c}\mathbf{0} \\ r_{D-1,D}\end{array}\right] = r_{D,D} - r_{D-1,D}^2 c_{D-1,D-1}^{(D-1)}. \tag{22}$$

The variable $c_{D-1,D-1}^{(D-1)}$ denotes the last element of the last column $\boldsymbol{c}_{D-1}^{(D-1)}$ of $\boldsymbol{C}^{(D-1)}$ at the previous step. Recall that only the first element of $\boldsymbol{r}_{\boldsymbol{d},d_0}^{(D)}$ is unequal 0 (cf. Equation 13), thus, only the first column $\boldsymbol{c}_1^{(D)}$ of the inverse $\boldsymbol{C}^{(D)}$

$$\boldsymbol{c}_1^{(D)} = \left[\begin{array}{c}\boldsymbol{c}_1^{(D-1)} \\ 0\end{array}\right] + \beta_D^{-1}c_{1,D-1}^{(D-1)}\left[\begin{array}{c}r_{D-1,D}^2\boldsymbol{c}_{D-1}^{(D-1)} \\ -r_{D-1,D}\end{array}\right], \tag{23}$$

where $c_{1,D-1}^{(D-1)}$ denotes the first element of $\boldsymbol{c}_{D-1}^{(D-1)}$, is needed for the computation of the WF $\boldsymbol{w}_{\boldsymbol{d}}^{(D)}$ using Equation (15). We observe that the filter at step D depends upon the first column $\boldsymbol{c}_1^{(D-1)}$ at step $D-1$, the new entries of the covariance matrix $r_{D-1,D}$ and $r_{D,D}$, and the previous last column $\boldsymbol{c}_{D-1}^{(D-1)}$. Using again Equation (20), we get the following recursion formula for the new last column

$$\boldsymbol{c}_D^{(D)} = \beta_D^{-1}\left[\begin{array}{c}-r_{D-1,D}\boldsymbol{c}_{D-1}^{(D-1)} \\ 1\end{array}\right], \tag{24}$$

which only depends on the previous last column and the new entries of $\boldsymbol{R}_{\boldsymbol{d}}^{(D)}$. So, we only have to update the vectors $\boldsymbol{c}_1^{(D)}$ and $\boldsymbol{c}_D^{(D)}$ at each iteration step. Moreover, the mean square error at step D can be updated using the first entry $c_{1,1}^{(D)}$ of $\boldsymbol{c}_1^{(D)}$ as follows (cf. Equation 17)

$$\mathrm{MSE}^{(D)} = \sigma_{d_0}^2 - \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2^2 c_{1,1}^{(D)}. \tag{25}$$

Finally, substituting $\boldsymbol{c}_1^{(i)}$ and $\boldsymbol{c}_i^{(i)}$ by $\boldsymbol{c}_{\text{first}}^{(i)}$ and $\boldsymbol{c}_{\text{last}}^{(i)}$, repectively, and using the last elements of the vectors in Equation (24), i.e. $c_{D,D}^{(D)} = \beta_D^{-1}$, to replace $c_{D-1,D-1}^{(D-1)}$ in Equation (22), yield the resulting Lanczos based implementation of the MSNWF given by Algorithm 1 which is used in Section 4 to show its relationship to the CG method.

Algorithm 1 Lanczos Based MSNWF

$\boldsymbol{t}_0 = \boldsymbol{0}$
$\boldsymbol{t}_1 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0} / \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2$
$r_{0,1} = 0, \quad r_{1,1} = \beta_1 = \boldsymbol{t}_1^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_1$
$c_{\text{first}}^{(1)} = c_{\text{last}}^{(1)} = r_{1,1}^{-1}$
$\mathrm{MSE}^{(1)} = \sigma_{d_0}^2 - \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2^2 c_{\text{first}}^{(1)}$
for $i = 2$ to D **do**
  $\boldsymbol{v} = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_{i-1} - r_{i-1,i-1} \boldsymbol{t}_{i-1} - r_{i-2,i-1} \boldsymbol{t}_{i-2}$
  $r_{i-1,i} = \left\|\boldsymbol{v}\right\|_2$
  $\boldsymbol{t}_i = \boldsymbol{v} / r_{i-1,i}$
  $r_{i,i} = \boldsymbol{t}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_i$
  $\beta_i = r_{i,i} - r_{i-1,i}^2 \beta_{i-1}^{-1}$
  $\boldsymbol{c}_{\text{first}}^{(i)} = \begin{bmatrix} \boldsymbol{c}_{\text{first}}^{(i-1)} \\ 0 \end{bmatrix} + \beta_i^{-1} c_{\text{last},1}^{(i-1)} \begin{bmatrix} r_{i-1,i}^2 \boldsymbol{c}_{\text{last}}^{(i-1)} \\ -r_{i-1,i} \end{bmatrix}$
  $\boldsymbol{c}_{\text{last}}^{(i)} = \beta_i^{-1} \begin{bmatrix} -r_{i-1,i} \boldsymbol{c}_{\text{last}}^{(i-1)} \\ 1 \end{bmatrix}$
  $\mathrm{MSE}^{(i)} = \sigma_{d_0}^2 - \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2^2 c_{\text{first},1}^{(i)}$
end for
$\boldsymbol{T}^{(D)} = \begin{bmatrix} \boldsymbol{t}_1 & \cdots & \boldsymbol{t}_D \end{bmatrix}$
$\boldsymbol{w}_0^{(D)} = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \boldsymbol{T}^{(D)} \boldsymbol{c}_{\text{first}}^{(D)}$

3 Conjugate Gradient Algorithm

The iterative *Conjugate Gradient* (CG) *algorithm* was developed by Hestenes and Stiefel [6] for solving a system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ of N linear equations in N unknowns. The solution is given in N steps. If we stop the algorithm after D steps, we get an approximate solution. One possible realization is given by Algorithm 2, where the matrix $\boldsymbol{A} \in \mathbb{C}^{N \times N}$ has to be symmetric and positive definite. Other implementations may be obtained [6] by replacing Lines 4 and/or 7 of Algorithm 2 by

$$\gamma_i = \left(\boldsymbol{r}_i^{\mathrm{H}} \boldsymbol{r}_i\right) / \left(\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{A} \boldsymbol{p}_i\right) \quad \text{and/or} \tag{26}$$

$$\delta_i = \left(\boldsymbol{r}_{i+1}^{\mathrm{H}} \boldsymbol{r}_{i+1}\right) / \left(\boldsymbol{r}_i^{\mathrm{H}} \boldsymbol{r}_i\right), \tag{27}$$

respectively. These formulas reduce computational complexity as can be seen in Section 5.

The fundamental recursion formula which updates the approximate solution of the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is Line 5 of Algorithm 2, where $\boldsymbol{p}_i$ is the *search direction* at iteration step i and γ_i is its *weight factor*. To understand Line 4 of Algorithm 2, we introduce the *error function* as the $\boldsymbol{A}$-norm of the error of the approximate solution $\boldsymbol{x}$ to the exact solution $\boldsymbol{x}^{(N)}$, i.e.

$$f\left(\boldsymbol{x}\right) = \left\|\boldsymbol{x}^{(N)} - \boldsymbol{x}\right\|_{\boldsymbol{A}} = \left(\boldsymbol{x}^{(N)} - \boldsymbol{x}\right)^{\mathrm{H}} \boldsymbol{A} \left(\boldsymbol{x}^{(N)} - \boldsymbol{x}\right) = \boldsymbol{x}^{\mathrm{H}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{x}^{\mathrm{H}} \boldsymbol{b} - \boldsymbol{b}^{\mathrm{H}} \boldsymbol{x} + \boldsymbol{x}^{(N),\mathrm{H}} \boldsymbol{b}. \tag{28}$$

Algorithm 2 Conjugate Gradient Method

$\boldsymbol{x}^{(0)} = \boldsymbol{0}$
$\boldsymbol{p}_1 = -\boldsymbol{r}_1 = \boldsymbol{b}$
for $i = 1$ to D **do**
  $\gamma_i = -\left(\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{r}_i\right) / \left(\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{A} \boldsymbol{p}_i\right)$
5: $\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)} + \gamma_i \boldsymbol{p}_i$
  $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \gamma_i \boldsymbol{A} \boldsymbol{p}_i$
  $\delta_i = \left(\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{A} \boldsymbol{r}_{i+1}\right) / \left(\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{A} \boldsymbol{p}_i\right)$
  $\boldsymbol{p}_{i+1} = -\boldsymbol{r}_{i+1} + \delta_i \boldsymbol{p}_i$
end for

The choice of γ_i ensures that the approximate solution $\boldsymbol{x}^{(i)}$ minimizes the error function $f(\boldsymbol{x})$ on the line $\boldsymbol{x} = \boldsymbol{x}^{(i-1)} + \gamma \boldsymbol{p}_i$. It holds

$$\gamma_i = \arg\min_{\gamma} f\left(\boldsymbol{x}^{(i-1)} + \gamma \boldsymbol{p}_i\right). \tag{29}$$

Line 6 of Algorithm 2 updates the *residual* $\boldsymbol{r}_i = \boldsymbol{A}\boldsymbol{x}^{(i-1)} - \boldsymbol{b}$. Note that the residual $\boldsymbol{r}_i$ belongs to the approximation $\boldsymbol{x}^{(i-1)}$. The index mismatch is useful for the derivations we make in Section 4. The recursion formula for the residuals can easily be derived as follows

$$\boldsymbol{r}_{i+1} = \boldsymbol{A}\boldsymbol{x}^{(i)} - \boldsymbol{b} = \boldsymbol{A}\left(\boldsymbol{x}^{(i-1)} + \gamma_i \boldsymbol{p}_i\right) - \boldsymbol{b} = \boldsymbol{r}_i + \gamma_i \boldsymbol{A} \boldsymbol{p}_i \tag{30}$$

by using Line 5 of Algorithm 2.

The CG algorithm belongs to the *Conjugate Directions* (CD) *methods* because the search directions are mutually $\boldsymbol{A}$-conjugate, i.e.

$$\boldsymbol{p}_j^{\mathrm{H}} \boldsymbol{A} \boldsymbol{p}_i = 0 \qquad \forall j < i \leq D, \tag{31}$$

which follows from Lines 6, 7, and 8 of Algorithm 2. Moreover, the CG method is a special family within the CD methods because the residuals are mutually orthogonal. It holds

$$\boldsymbol{r}_j^{\mathrm{H}} \boldsymbol{r}_i = 0 \qquad \forall j < i \leq D. \tag{32}$$

The proofs of Equations (31) and (32) [6] are not shown in this paper due to space limitations.

4 Relationship Between CG and MSNWF

In numerous papers and books [5, 7] it was mentioned that the Lanczos algorithm which is used by our special implementation of the MSNWF is only a version of the CG algorithm. If we compare the optimization function in Equations (1) and (28), we see that they are the same except for a constant which does not change the minimum computed by Equations (2) and (29), respectively. Moreover, the solution at each step i is searched in the same Krylov subspace $\mathcal{K}^{(D)}$ spanned by the column vectors of the matrix $\boldsymbol{T} = \left[\begin{array}{ccc} \boldsymbol{t}_1 & \cdots & \boldsymbol{t}_D \end{array}\right]$ as shown in [7]. To establish the equivalence of both algorithms, we transform the formulas of the Lanczos based MSNWF to those of the CG algorithm.

Assume that $D \geq i \geq 2$. Using Lines 17 and 12 of Algorithm 1, and by setting $\boldsymbol{T}^{(i)} = \left[\begin{array}{cc} \boldsymbol{T}^{(i-1)} & \boldsymbol{t}_i \end{array}\right]$, it holds for $\boldsymbol{w}_0^{(i)}$ that

$$\boldsymbol{w}_0^{(i)} = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \boldsymbol{T}^{(i)} \boldsymbol{c}_{\text{first}}^{(i)} = \boldsymbol{w}_0^{(i-1)} + \eta_i \boldsymbol{u}_i, \quad \boldsymbol{w}_0^{(1)} = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1} \boldsymbol{t}_1, \tag{33}$$

where η_i and $\boldsymbol{u}_i$ are defined as

$$\eta_i = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \rho_i^{-1} \beta_i^{-1} r_{i-1,i} c_{\text{last},1}^{(i-1)}, \tag{34}$$

$$\boldsymbol{u}_i = \rho_i \left(r_{i-1,i} \boldsymbol{T}^{(i-1)} \boldsymbol{c}_{\text{last}}^{(i-1)} - \boldsymbol{t}_i\right), \tag{35}$$

and where $\rho_i \in \mathbb{R} \setminus \{0\}$ is so far an arbitrary factor which meaning is established later. Multiplying Line 13 of Algorithm 1 on the left side by $\boldsymbol{T}^{(i)}$ and using Equation (35) leads to

$$\boldsymbol{T}^{(i)}\boldsymbol{c}_{\text{last}}^{(i)} = -\beta_i^{-1}\left(r_{i-1,i}\boldsymbol{T}^{(i-1)}\boldsymbol{c}_{\text{last}}^{(i-1)} - \boldsymbol{t}_i\right) = -\rho_i^{-1}\beta_i^{-1}\boldsymbol{u}_i. \tag{36}$$

Substituting i by $i+1$ in Equation (35) and replacing $\boldsymbol{T}^{(i)}\boldsymbol{c}_{\text{last}}^{(i)}$ by using Equation (36) yields a recursion formula for $\boldsymbol{u}_{i+1}$ and $D-1 \geq i \geq 1$

$$\boldsymbol{u}_{i+1} = \psi_i\boldsymbol{u}_i - \boldsymbol{g}_{i+1}, \quad \boldsymbol{u}_1 = -\rho_1\boldsymbol{t}_1, \tag{37}$$

where

$$\psi_i = -\rho_{i+1}\rho_i^{-1}\beta_i^{-1}r_{i,i+1}, \tag{38}$$

$$\boldsymbol{g}_{i+1} = \rho_{i+1}\boldsymbol{t}_{i+1}. \tag{39}$$

In Equations (33) and (37) we observe the analogous terms in Lines 5 and 8 of Algorithm 2. The vectors $\boldsymbol{u}_i$ and $\boldsymbol{g}_i$ are seen to be equivalent to the search direction $\boldsymbol{p}_i$ and the residual $\boldsymbol{r}_i$ of the CG algorithm, respectively. In the Lanczos based MSNWF, the linear system to be solved is the Wiener-Hopf equation (cf. Equation 3) where the solution is the WF $\boldsymbol{w}_0$. In the sequel, we prove that the MSNWF can be transformed into the CG algorithm.

Proposition 1. *The vectors $\boldsymbol{g}_i$ for $D \geq i \geq 1$ can be updated by the recursion formula*

$$\boldsymbol{g}_{i+1} = \boldsymbol{g}_i + \eta_i\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i, \quad \boldsymbol{g}_1 = \rho_1\boldsymbol{t}_1, \quad \rho_1 = -\left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2, \tag{40}$$

where the definition of η_i is given in Equation (34) and where

$$\eta_1 = r_{1,1}^{-1}. \tag{41}$$

Proof. We prove by induction. First, set $i=1$. Recall that $\boldsymbol{u}_1 = -\rho_1\boldsymbol{t}_1 = -\boldsymbol{g}_1$ and we get by using the initialization of Equations (40) and (41), and Lines 7 to 9 of Algorithm 1

$$\boldsymbol{g}_1 + \eta_1\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_1 = \rho_1\eta_1\left(\eta_1^{-1}\boldsymbol{t}_1 - \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_1\right) = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1}r_{1,2}\boldsymbol{t}_2 = \rho_2\boldsymbol{t}_2 = \boldsymbol{g}_2.$$

The last equality holds if we define ρ_2 as

$$\rho_2 = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1}r_{1,2}. \tag{42}$$

Now assume that Equation (40) holds for $i = n-1$. This leads to

$$\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_{n-1} = \eta_{n-1}^{-1}\left(\boldsymbol{g}_n - \boldsymbol{g}_{n-1}\right). \tag{43}$$

To prove Equation (40) for $i = n$ if it holds for $i = n-1$, we need a transformation of Equation (37). Substituting $i+1$ by n and multiplying the equation on the left side by $\boldsymbol{R}_{\boldsymbol{x}_0}$ yields

$$\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_n = \psi_{n-1}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_{n-1} - \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_n. \tag{44}$$

Finally, we get by using Equations (44) and (43)

$$\begin{aligned}
\boldsymbol{g}_n + \eta_n\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_n &= \boldsymbol{g}_n - \eta_n\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_n + \eta_n\psi_{n-1}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_{n-1}\\
&= \boldsymbol{g}_n - \eta_n\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_n + \eta_{n-1}^{-1}\eta_n\psi_{n-1}\left(\boldsymbol{g}_n - \boldsymbol{g}_{n-1}\right)\\
&= -\rho_n\eta_n\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_n + \rho_n\left(1 + \eta_{n-1}^{-1}\eta_n\psi_{n-1}\right)\boldsymbol{t}_n - \rho_{n-1}\eta_{n-1}^{-1}\eta_n\psi_{n-1}\boldsymbol{t}_{n-1}.
\end{aligned} \tag{45}$$

On the other hand, multiplying the recursion formula in Lines 7 to 9 of Algorithm 1 by ρ_{n+1} on both sides yields

$$\rho_{n+1}\boldsymbol{t}_{n+1} = \rho_{n+1}r_{n,n+1}^{-1}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_n - \rho_{n+1}r_{n,n+1}^{-1}r_{n,n}\boldsymbol{t}_n - \rho_{n+1}r_{n,n+1}^{-1}r_{n-1,n}\boldsymbol{t}_{n-1}. \tag{46}$$

Recalling Equation (39) and comparing Equation (45) with (46), Proposition 1 is proved if the following three equations are true:

$$-\rho_n \eta_n = \rho_{n+1} r_{n,n+1}^{-1}, \tag{47}$$

$$\rho_n \left(1 + \eta_{n-1}^{-1} \eta_n \psi_{n-1}\right) = -\rho_{n+1} r_{n,n+1}^{-1} r_{n,n}, \tag{48}$$

$$\rho_{n-1} \eta_{n-1}^{-1} \eta_n \psi_{n-1} = \rho_{n+1} r_{n,n+1}^{-1} r_{n-1,n}. \tag{49}$$

Using Equations (47), (34), and Line 13 of Algorithm 1 to calculate ρ_{n+1} yields for $D-1 \geq n \geq 2$

$$\rho_{n+1} = -\rho_n \eta_n r_{n,n+1} = -\left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \beta_n^{-1} r_{n,n+1} r_{n-1,n} c_{\text{last},1}^{(n-1)} = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{n,n+1} c_{\text{last},1}^{(n)}.$$

If $n+1 = i$ and we include Equation (42) for $i = 2$, ρ_i for $D \geq i \geq 2$ is given by the formulas

$$\rho_i = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{i-1,i} c_{\text{last},1}^{(i-1)}, \tag{50}$$

$$\rho_i = -\rho_{i-1} \eta_{i-1} r_{i-1,i}. \tag{51}$$

If we plug Equations (50) and (51) in Equations (34) and (38), respectively, we get simpler formulas for η_i and ψ_i

$$\eta_i = \beta_i^{-1}, \qquad D \geq i \geq 1, \tag{52}$$

$$\psi_i = \beta_i^{-2} r_{i,i+1}^2, \qquad D-1 \geq i \geq 1. \tag{53}$$

To prove Equation (48) use Equations (51), (52), (53), and Line 11 of Algorithm 1. It follows

$$\begin{aligned}\rho_n \left(1 + \eta_{n-1}^{-1} \eta_n \psi_{n-1}\right) &= -\rho_{n+1} \eta_n^{-1} r_{n,n+1}^{-1} \left(1 + \beta_n^{-1} \beta_{n-1}^{-1} r_{n-1,n}^2\right) \\ &= -\rho_{n+1} r_{n,n+1}^{-1} \left(\beta_n + r_{n-1,n}^2 \beta_{n-1}^{-1}\right) = -\rho_{n+1} r_{n,n+1}^{-1} r_{n,n}.\end{aligned}$$

With the same equations, we get

$$\rho_{n-1} \eta_{n-1}^{-1} \eta_n \psi_{n-1} = \rho_{n+1} \eta_{n-1}^{-2} \psi_{n-1} r_{n-1,n}^{-1} r_{n,n+1}^{-1} = \rho_{n+1} r_{n,n+1}^{-1} r_{n-1,n}.$$

Thus, Equations (47), (48), and (49) hold and Proposition 1 is proved. □

Note that with the definition of η_1 by Equation (41), Equation (33) holds for $i = 1$, where $\boldsymbol{w}_0^{(0)} = 0$, as well as for $D \geq i > 1$.

Proposition 2. *The vectors $\boldsymbol{g}_i$ are residual vectors for $D+1 \geq i \geq 1$. Thus,*

$$\boldsymbol{g}_i = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(i-1)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}. \tag{54}$$

Proof. Again, this proposition can be proved by induction. Set $i = 1$. It holds that

$$\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(0)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = -\left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \boldsymbol{t}_1 = \boldsymbol{g}_1.$$

Assume that Equation (54) holds for $i = n-1$. Hence, we get for $i = n$ (cf. Equation 33)

$$\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(n-1)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(n-2)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} + \eta_{n-1} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{n-1} = \boldsymbol{g}_{n-1} + \eta_{n-1} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{n-1} = \boldsymbol{g}_n$$

The last equality holds because of Proposition 1. □

Up to now we showed, in addition to Equations (33) and (37), that the residuals $\boldsymbol{g}_i$ are updated by a similar formula as the residuals $\boldsymbol{r}_i$. To derive the remaining problem, we need to show that the computation of the weight factors ψ_i and η_i is the same as for those in the CG algorithm. First, we prove the following proposition.

Proposition 3. *The vectors $\boldsymbol{u}_j$ and $\boldsymbol{g}_i$ satisfy the relation*

$$\boldsymbol{u}_j^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = 0, \qquad \forall \left(j \leq i-2 \wedge i \leq D\right). \tag{55}$$

Proof. First, let $j = 1$ and $D \geq i \geq 3$, but fixed. With Equations (37) and (39) we get

$$\boldsymbol{u}_1^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = -\rho_1 \rho_i \boldsymbol{t}_1^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_i = -\rho_1 \rho_i r_{1,i} = 0.$$

The last equality holds, because $r_{j,i}$ are the elements of the tridiagonal matrix $\boldsymbol{R}_{\boldsymbol{d}}^{(D)}$ (cf. Section 2) and thus, $r_{j,i} = 0$ for all $j \leq i - 2$ and $i \leq D$. Assume that Equation (55) holds for $j = n - 1$. Then, we get for $j = n$

$$\boldsymbol{u}_n^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = \psi_{n-1} \boldsymbol{u}_{n-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i - \boldsymbol{g}_n^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = -\rho_n \rho_i \boldsymbol{t}_n^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_i = -\rho_n \rho_i r_{n,i} = 0.$$

□

Proposition 4. *The factor ψ_i in Equation (37) satisfies the relation*

$$\psi_i = \frac{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_{i+1}}{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i} \tag{56}$$

and thus, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D\}$ is a set of $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate vectors, i.e.

$$\boldsymbol{u}_j^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i = 0, \qquad \forall j < i \leq D. \tag{57}$$

Proof. Use Equation (37) to get

$$\boldsymbol{u}_{i-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = \psi_{i-2} \boldsymbol{u}_{i-2}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i - \boldsymbol{g}_{i-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i = -\rho_{i-1} \rho_i r_{i-1,i}. \tag{58}$$

The last equality holds because of Proposition 3.

To optain a similar expression for the denominator on the right side of Equation (56), substitute $\boldsymbol{u}_i$ given by Equation (37) and use Equations (39), (53), (58), and Line 10 of Algorithm 1.

$$\begin{aligned}
\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i &= \boldsymbol{g}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i - \psi_{i-1} \boldsymbol{g}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{i-1} - \psi_{i-1} \boldsymbol{u}_{i-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_i + \psi_{i-1}^2 \boldsymbol{u}_{i-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{i-1} \\
&= \rho_i^2 \left(r_{i,i} - 2 r_{i-1,i}^2 \beta_{i-1}^{-1} + r_{i-1,i}^2 \beta_{i-1}^{-1} \frac{\boldsymbol{u}_{i-1}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{i-1}}{\rho_{i-1}^2 \beta_{i-1}} \right)
\end{aligned}$$

Comparing the equation above with Line 11 of Algorithm 1 leads to

$$\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i = \rho_i^2 \beta_i. \tag{59}$$

Substituting i by $i+1$ in Equation (58) and dividing it by the left side of Equation (59) yields Equation (56). Thus, similar to δ_i in Algorithm 2, ψ_i, Equations (37), and (40) ensure that the vectors $\boldsymbol{u}_i$ are mutually $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate. □

Proposition 5. *It holds for $D \geq i \geq 1$ that*

$$\eta_i = -\frac{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i}. \tag{60}$$

Thus, with $\boldsymbol{w} = \boldsymbol{w}_0^{(i-1)} + \eta_i \boldsymbol{u}_i$ the value of η_i above minimizes the error function

$$f(\boldsymbol{w}) = \boldsymbol{w}^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w} - \boldsymbol{w}^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}^{\mathrm{H}} \boldsymbol{w}. \tag{61}$$

Proof. Replacing $\boldsymbol{g}_i$ by Equation (54), $\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i$ by Equation (59) and using recursively Equation (33) yields

$$\begin{aligned}
-\frac{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i} &= -\rho_i^{-2} \beta_i^{-1} \boldsymbol{u}_i^{\mathrm{H}} \left(\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(i-1)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} \right) \\
&= \rho_i^{-2} \beta_i^{-1} \boldsymbol{u}_i^{\mathrm{H}} \left(\boldsymbol{r}_{\boldsymbol{x}_0,d_0} - \boldsymbol{R}_{\boldsymbol{x}_0} \sum_{k=1}^{i-1} \eta_k \boldsymbol{u}_k \right) \\
&= \rho_i^{-2} \beta_i^{-1} \boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0}.
\end{aligned}$$

The last equality holds, because the vectors $\boldsymbol{u}_i$ are mutually $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate (cf. Equation 57). First, set $i = 1$. It holds by using $\boldsymbol{u}_1 = -\rho_1 \boldsymbol{t}_1$ and Equation (52) that

$$\rho_1^{-2} \beta_1^{-1} \boldsymbol{u}_1^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \beta_1^{-1} = \eta_1.$$

Then, set $D \geq i \geq 2$ and substitute $\boldsymbol{u}_i$ by Equation (35)

$$\begin{aligned}\rho_i^{-2} \beta_i^{-1} \boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} &= \rho_i^{-1} \beta_i^{-1} r_{i-1,i} \boldsymbol{c}_{\mathrm{last}}^{(i-1),\mathrm{T}} \boldsymbol{T}^{(i-1),\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} - \rho_i^{-2} \beta_i^{-1} \boldsymbol{t}_i^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} \\ &= \left\| \boldsymbol{r}_{\boldsymbol{x}_0,d_0} \right\|_2 \rho_i^{-1} \beta_i^{-1} r_{i-1,i} c_{\mathrm{last},1}^{(i-1)} = \eta_i.\end{aligned}$$

For the second equality remember, that $\boldsymbol{t}_i^{\mathrm{H}} \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = 0$ for all $D \geq i \geq 2$. $\square$

Thus, we see that the remaining Equations (60) and (56) which compute the weight factors ψ_i and η_i are similar to Lines 4 and 7 of Algorithm 2. To put it in a nutshell, the Lanczos based MSNWF uses the same formulas as the CG algorithm if we make the following equivalences

$$\text{approximate solution:} \quad \boldsymbol{w}_0^{(i)} \leftrightarrow \boldsymbol{x}^{(i)}, \qquad \eta_i \leftrightarrow \gamma_i, \tag{62}$$

$$\text{search directions:} \quad \boldsymbol{u}_i \leftrightarrow \boldsymbol{p}_i, \qquad \psi_i \leftrightarrow \delta_i, \tag{63}$$

$$\text{residuals:} \quad \boldsymbol{g}_i \leftrightarrow \boldsymbol{r}_i. \tag{64}$$

5 A New Implementation of MSNWF

In Section 4 we have derived that the Lanczos based MSNWF can be expressed by the CG algorithm. Thus, Equations (60) and (56) can be replaced in the same manner as Lines 4 and 7 of Algorithm 2 by Equations (26) and (27), respectively. It follows

$$\eta_i = \frac{\boldsymbol{g}_i^{\mathrm{H}} \boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i}, \tag{65}$$

$$\psi_i = \frac{\boldsymbol{g}_{i+1}^{\mathrm{H}} \boldsymbol{g}_{i+1}}{\boldsymbol{g}_i^{\mathrm{H}} \boldsymbol{g}_i}. \tag{66}$$

This reduces computational complexity because the matrix vector multiplication $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_{i+1}$ in Equation (56) is avoided. Besides, the only product left, $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i$, which is needed in Equation (40) has already been computed in Equation (65). Therefore, the resulting computational complexity for a rank D MSNWF is $O\left(N^2 D\right)$, since a matrix vector multiplication with $O\left(N^2\right)$ has to be performed at each step.

Comparing to the Lanczos implementation of the MSNWF, the CG algorithm does not compute the mean square error $\mathrm{MSE}^{(i)}$ at each step. To get such a recursion formula for $\mathrm{MSE}^{(i)}$ in the CG implementation, consider the first elements in Line 12 of Algorithm 1. It holds for $D \geq i \geq 2$ that

$$c_{\mathrm{first},1}^{(i)} = c_{\mathrm{first},1}^{(i-1)} + \beta_i^{-1} r_{i-1,i}^2 c_{\mathrm{last},1}^{(i-1),2}. \tag{67}$$

Use this equation to replace $c_{\mathrm{first},1}^{(i)}$ in Line 14 of Algorithm 1

$$\mathrm{MSE}^{(i)} = \mathrm{MSE}^{(i-1)} - \left\| \boldsymbol{r}_{\boldsymbol{x}_0,d_0} \right\|_2^2 \beta_i^{-1} r_{i-1,i}^2 c_{\mathrm{last},1}^{(i-1),2} = \mathrm{MSE}^{(i-1)} - \rho_i^2 \eta_i, \tag{68}$$

where $\mathrm{MSE}^{(1)}$ is defined by Line 4 of Algorithm 1. The last equality holds because of Equations (50) and (52). The fact that the factor ρ_i is the length of the residual $\boldsymbol{g}_i$ and that $\mathrm{MSE}^{(1)} = \sigma_{d_0}^2 - \rho_1^2 \eta_1$ yields for $D \geq i \geq 1$

$$\mathrm{MSE}^{(i)} = \mathrm{MSE}^{(i-1)} - \eta_i \boldsymbol{g}_i^{\mathrm{H}} \boldsymbol{g}_i, \quad \mathrm{MSE}^{(0)} = \sigma_{d_0}^2. \tag{69}$$

Finally, considering Equations (65), (33), (69), (40), (66), and (37) leads to a CG implementation of the MSNWF which is given by Algorithm 3. In the following section, the resulting CG/MSNWF hybrid algorithm is applied as a linear equalizer to an EDGE system.

Algorithm 3 CG MSNWF

$\boldsymbol{w}_0^{(0)} = \mathbf{0}$
$\boldsymbol{u}_1 = -\boldsymbol{g}_1 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0}$
$l_1 = \boldsymbol{g}_1^{\mathrm{H}} \boldsymbol{g}_1$
$\mathrm{MSE}^{(0)} = \sigma_{d_0}^2$
for $i = 1$ to D **do**
  $\boldsymbol{v} = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i$
  $\eta_i = l_i / \left(\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{v} \right)$
  $\boldsymbol{w}_0^{(i)} = \boldsymbol{w}_0^{(i-1)} + \eta_i \boldsymbol{u}_i$
  $\mathrm{MSE}^{(i)} = \mathrm{MSE}^{(i-1)} - \eta_i l_i$
  $\boldsymbol{g}_{i+1} = \boldsymbol{g}_i + \eta_i \boldsymbol{v}$
  $l_{i+1} = \boldsymbol{g}_{i+1}^{\mathrm{H}} \boldsymbol{g}_{i+1}$
  $\psi_i = l_{i+1} / l_i$
  $\boldsymbol{u}_{i+1} = -\boldsymbol{g}_{i+1} + \psi_i \boldsymbol{u}_i$
end for

6 Application to an EDGE System

In the following we consider an EDGE system with 8PSK modulation and *Laurent pulse shaping*. The Laurent impulse is a linearized GMSK impulse [8] which has a duration of five symbol times. Thus, we have severe intersymbol interference even without channel distortion. The symbol time $T = 3.69\,\mu$s and the two antennas of the mobile station (MS) receive the signal of a base station which propagates over Rayleigh multipath fading channels with a delay spread of $\tau_{\max} = 10\,\mu$s or three symbol times. We assume a constant channel during one burst with 148 symbols (excluding guard symbols).

The CG based implementation of the MSNWF is used as a linear equalizer filter for the received signal at the MS. We sample two times during one symbol duration and take 20 samples of each antenna to build the space-time observation vector $\boldsymbol{x}_0[n]$, thus, its dimension $N = 40$.

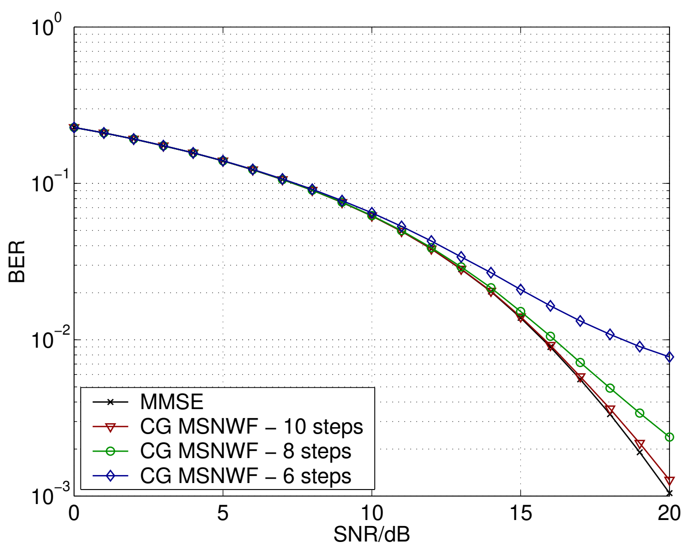

Figure 2: BER for known channel using CG MSNWF equalizer

Figure 2 shows the BER using the CG based MSNWF for $D \in \{6, 8, 10\}$ steps compared to the MMSE equalizer or WF which corresponds to the MSNWF with $D = 40$ steps. We use the ideal form of the covariance matrix and the crosscorrelation vector. We observe that the MSNWF with $D = 10$ steps is very close to the MMSE equalizer even for high SNR values.

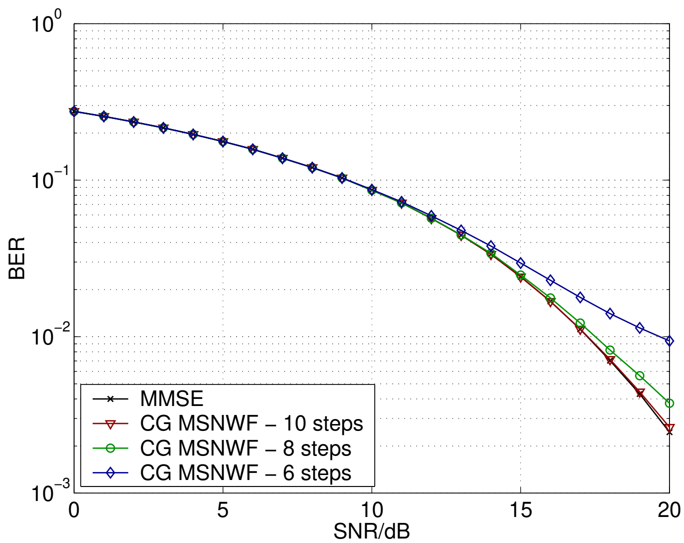

Figure 3: BER for estimated channel using CG MSNWF equalizer

Using the 26 training symbols of a burst to estimate the channel and therefore the crosscorrelation vector and the covariance matrix, leads to the simulation results shown in Figure 3. Again, the MSNWF with $D = 10$ steps is a good approximation of the MMSE equalizer although it has much less computational complexity.

Figure 2 and 3 plot uncoded BER. Due to error control coding, an uncoded BER of 10^{-1} results in acceptable speech transmission.

7 Conclusion

In this paper we derived the relationship between the Lanczos based implementation of the MSNWF and the CG method. A new implementation of the MSNWF is obtained by transforming its formulas into those of the CG algorithm. Simulation results of an application to an EDGE system showed that despite the reduced computational complexity, the CG based MSNWF yields almost the same results as the MMSE equalizer.

Acknowledgement

This research was supported by the Communications, Signal Processing, and Surveillance Program of the Air Force Office of Scientific Research, under Grant No. F49620-00-1-0127.

References

[1] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, vol. 24, no. 6/7, pp. 417–441, 498–520, September/October 1933.

[2] J. S. Goldstein and I. S. Reed, "Subspace Selection for Partially Adaptive Sensor Array Processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 539–543, April 1997.

[3] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A Multistage Representation of the Wiener Filter Based on Orthogonal Projections," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2943–2959, November 1998.

[4] M. L. Honig and W. Xiao, "Performance of Reduced-Rank Linear Interference Suppression for DS-CDMA," submitted to *IEEE Transactions on Information Theory*, December 1999.

[5] M. Joham and M. D. Zoltowski, "Interpretation of the Multi-Stage Nested Wiener Filter in the Krylov Subspace Framework," Tech. Rep. TUM-LNS-TR-00-6, Munich University of Technology, November 2000, Also: Technical Report TR-ECE-00-51, Purdue University.

[6] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, December 1952.

[7] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[8] European Telecommunications Standards Institute, "Digital cellular telecommunications system (Phase 2+); Modulation," EN 300 959 V7.1.1, ETSI, http://www.etsi.org, June 2000.