Two Dimensional Power Spectral Estimation Using Constrained Iterative Spectral Deconvolution

Jerome Leslie Coggins

University of New Orleans, jlcoggin@uno.edu

Follow this and additional works at: https://scholarworks.uno.edu/td

Part of the Physics Commons

Recommended Citation

https://scholarworks.uno.edu/td/2821

This Thesis is protected by copyright and/or related rights. It has been brought to you by ScholarWorks@UNO with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in University of New Orleans Theses and Dissertations by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.
TWO DIMENSIONAL POWER SPECTRAL ESTIMATION USING
CONSTRAINED ITERATIVE SPECTRAL DECONVOLUTION

A Thesis
Presented to
the Faculty of the Graduate School
of the University of New Orleans
In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Applied Physics

in

The Department of Physics

by
Jerome Leslie Coggins
B.S., University of Illinois, 1980
May 1993
ACKNOWLEDGMENTS

I would like to thank Dr. George Loup for his guidance and insight. I would also like to thank Mr. Murali Ramaswamy and Dr. Paul Hatchell for their help.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Theoretical Development</td>
<td>2</td>
</tr>
<tr>
<td>Autocorrelation Estimation</td>
<td>2</td>
</tr>
<tr>
<td>Power Spectrum Estimation</td>
<td>4</td>
</tr>
<tr>
<td>Nonnegative Definite Property</td>
<td>5</td>
</tr>
<tr>
<td>Deconvolution</td>
<td>13</td>
</tr>
<tr>
<td>Van Cittert Deconvolution</td>
<td>15</td>
</tr>
<tr>
<td>Constrained Iterative Spectral Deconvolution</td>
<td>17</td>
</tr>
<tr>
<td>Comparison to Other Methods</td>
<td>24</td>
</tr>
<tr>
<td>Relaxation Parameter</td>
<td>32</td>
</tr>
<tr>
<td>Compensation for Missing Samples</td>
<td>38</td>
</tr>
<tr>
<td>Equivalent CISD Window</td>
<td>44</td>
</tr>
<tr>
<td>Two-Dimensional CISD</td>
<td>50</td>
</tr>
<tr>
<td>Application to Model Data</td>
<td>52</td>
</tr>
<tr>
<td>Application to Downhole Sonic Data</td>
<td>58</td>
</tr>
<tr>
<td>Conclusions</td>
<td>65</td>
</tr>
<tr>
<td>References</td>
<td>67</td>
</tr>
<tr>
<td>Appendices</td>
<td>69</td>
</tr>
<tr>
<td>Vita</td>
<td>83</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. Two Sine Model.
2. Biased Autocorrelation Estimate of Two Sine Model.
3. Unbiased Autocorrelation Estimate of Two Sine Model.
4. Twenty Point Bartlett Window.
5. Power Spectrum of the Two Sine Model (Biased Estimate).
8. CISD Autocorrelation Estimate of Two Sine Model.
9. CISD Power Spectrum of Two Sine Model.
10. Mean Square Error Vs. Iteration Number for Two Sine Model.
11. Comparison of CISD Spectrum to Spectrum of 64 Point Model.
13. True Power Spectrum of the Kay and Marple Model.
17. CISD Power Spectrum of the Kay and Marple Model.
18. Convergence Condition for CISD.
20. Iteration Vs. Relaxation Parameter for a fixed Mean Square Error.
22. Power Spectrum of the Two Sine Model with Missing Data Samples.
23. CISD Power Spectrum of the Two Sine Model with Missing Data Samples (Bartlett Window).
24. CISD Power Spectrum of the Two Sine Model with Missing Data Samples (Compensated Window).
25. MSE Vs. Iteration Number for the Two Sine Model Using a CISD Equivalent Window.
26. MSE Vs. Iteration Number for the Two Sine Model before Constraint Application on Iteration 100.
27. CISD Power Spectrum for the Two Sine Model before Constraint Application on Iteration 100.
28. Two-Dimensional Two Sine Model.
29. Biased Autocorrelation Estimate of the 2-D Two Sine Model.
30. Power Spectrum of the 2-D Two Sine Model.
31. CISD Autocorrelation Estimate of the 2-D Two Sine Model.
32. CISD Power Spectrum of the 2-D Two Sine Model.
33. Downhole Sonic Data.
34. 2-D Power Spectrum of the Downhole Data.
35. 2-D CISD Power Spectrum of the Downhole Data.
36. 2-D Power Spectrum of the Downhole Data (Traces 5,6 set to zero).
37. 2-D CISD Power Spectrum of the Downhole Data (Traces 5,6 zero) Using a Compensated Window.
38. 2-D CISD Power Spectrum of the Downhole Data (Traces 5,6 zero) Using a 2-D Bartlett Window.
ABSTRACT

Two-dimensional power spectral estimation is an important tool for seismic data analysis and other applications. Some datasets, however, have a limited number of points in one or both dimensions. In seismic applications, there are typically fewer points in the spatial domain as compared to the temporal domain. Conventional spectral estimation techniques suffer from poor resolution on short datasets due to inherent smoothing or bias.

Ramaswamy and loup developed a one-dimensional method for the estimation of power spectra for short datasets. Constrained Iterative Spectral Deconvolution (CISD) greatly improves the power spectral resolution using a straightforward algorithm. In a comparison to other techniques, CISD is shown to perform very well on a standard 1-D dataset.

Two modifications to the CISD method are introduced that enhance its performance. A simple modification to the algorithm, the inclusion of a relaxation parameter, speeds convergence by a factor of two. Another modification uses an equivalent window to calculate multiple iterations between constraint applications. This enhancement did not improve convergence.

A method was developed that compensates the CISD technique for missing samples in the dataset. This promises to be of great practical value to real datasets. This method is demonstrated on both model and real datasets.

Finally, the CISD method is extended to the two-dimensional case, incorporating both modifications. This algorithm performs very well on a synthetic dataset and on real data from a downhole sonic tool. FORTRAN subroutines are given that implement the modified Constrained Iterative Spectral Estimation technique in both one and two dimensions.
INTRODUCTION

The method of deconvolution by successive substitutions, often referred to as iterative deconvolution, was developed by van Cittert (1931). Several authors (Bracewell and Roberts, 1954; loup, 1968; Lacoste, 1982; loup and loup, 1983; Jansson, 1984) have discussed this method and related topics such as convergence behavior. Perhaps the most interesting studies have applied iterative deconvolution, together with nonlinear constraints, to spectral estimation. For example, Ramaswamy and loup (1989) applied a constrained iterative deconvolution technique to the problem of autocorrelation estimation for short datasets. Using a nonnegative definite constraint, they extended the autocorrelation well beyond the original lags. The corresponding increase in resolution of the power spectrum was dramatic.

In seismic data applications, short datasets more often occur in the distance or offset dimension rather than in time. Although wavenumber spectra are sometimes used, the two dimensional (F-K) power spectrum is a more common diagnostic tool. Constrained Iterative Spectral Deconvolution (Ramaswamy and loup, 1989) would seem well suited to the estimation of two-dimensional power spectra when one or both dimensions have short windows. In this thesis, I extend the CISD technique to two dimensions. In addition, I implement two useful modifications to the general CISD method and I develop a technique for missing data compensation.

After a theoretical review of the development of CISD, I first introduce the concept of a relaxation parameter (Jansson, 1984) to increase the convergence rate of the technique by a factor of two. I then implement another refinement that uses an equivalent CISD window to apply the equivalent of N iterations between each constraint application (Whitehorn, 1981). Then, I develop and implement a window function that enables the CISD method to compensate for missing samples in the dataset. Finally, I extend CISD to the two-dimensional case, while incorporating some of these modifications. I demonstrate all of these techniques on both model and real data.
THEORETICAL DEVELOPMENT

The purpose of this section is to lay the groundwork for the further development of the Constrained Iterative Spectral Deconvolution method. After reviewing autocorrelation estimation, power spectrum estimation, and van Cittert iterative deconvolution, the CISD method (Ramaswamy, 1985; Ramaswamy and Ioup, 1989) will be discussed.

AUTOCORRELATION ESTIMATION

Constrained iterative spectral deconvolution is basically an autocorrelation extension technique, so it is appropriate to review the subject of autocorrelation initially.

Given a finite, uniformly sampled time sequence \( x(t) \), the autocorrelation function is defined as (Cooley et al., 1970):

\[
b(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} x(t)x(t-\tau) .
\]  

(1)

There is often confusion in the use of the terms autocorrelation and autocovariance. This paper uses the above definition consistently. An autocorrelation normalized to have a maximum value of one will be called simply a normalized autocorrelation. The autocovariance is reserved for an autocorrelation with the mean removed (Marple, 1987).

This estimate of autocorrelation is a biased one. Since \( x(t) \) is of finite duration, only the zero lag value of \( b \) contains \( N \) points. The estimates for larger \( \tau \) values are deemphasized. This biasing can be desirable since these values at longer lags are less statistically reliable. For large datasets, only ten to twenty percent of the available lags are commonly used to estimate the autocorrelation (Blackman and Tukey, 1958), so the effect of this biasing is minimal. However, for short datasets, all of the lags may need to be used to calculate the autocorrelation. In this case, the effect of the bias is significant.
An unbiased estimate of the autocorrelation can be calculated such that all lags are emphasized equally (Cooley et al., 1970). This is accomplished by dividing the estimate by a Bartlett window

\[ w(\tau) = (N - |\tau|) / N \quad \tau = 0, 1, 2, \ldots N - 1. \]  

(2)

The unbiased estimate can then be described as

\[ u(\tau) = 1 / (N - |\tau|) \sum_{t=0}^{N-1} x(t)x(t - \tau). \]  

(3)

The biased autocorrelation can be expressed in terms of the unbiased estimate as

\[ b(\tau) = u(\tau)w(\tau). \]  

(4)
POWER SPECTRUM ESTIMATION

Although constrained iterative spectral deconvolution is a technique for autocorrelation estimation, the Fourier transform of the autocorrelation is often the ultimate goal of the method. The Fourier transform of the autocorrelation function is the power spectrum, as given by the Wiener-Khintchine Theorem (Kanasewich, 1981). The power spectrum can be expressed as

$$B(\omega) = \sum_{\tau=0}^{N-1} b(\tau) e^{-i\omega \tau}.$$  \hspace{1cm} (5)

This is commonly called the indirect method because the power spectrum is calculated via the autocorrelation. Likewise the Bartlett window and the unbiased autocorrelation estimate can be expressed in the frequency domain using their power spectra,

$$u(\tau) \leftrightarrow U(\omega)$$
$$w(\tau) \leftrightarrow W(\omega).$$  \hspace{1cm} (6)

Using these power spectra and the Convolution Theorem, equation 4 can be rewritten as a convolution (Bracewell, 1978),

$$B(\omega) = U(\omega) * W(\omega).$$  \hspace{1cm} (7)

The power spectrum $B(\omega)$ is simply the spectrum of the unbiased estimate convolved with the power spectrum of the window, $W(\omega)$. 
NONNEGATIVE DEFINITE PROPERTY

The autocorrelation estimate \( b(\tau) \) of a real function \( x(t) \) is an even function whose maximum value is at \( t=0 \). However, meeting these two criteria is not sufficient to describe an autocorrelation function. The autocorrelation must also be nonnegative definite. This condition is equivalent to the autocorrelation meeting the condition known as the Wiener-Khintchine Theorem (Robinson, 1980), which states

\[
b(\tau) = \int_{-0.5}^{0.5} e^{2\pi i \tau f} d\Lambda(f) . \tag{8}
\]

\( \Lambda(f) \) is defined as the spectral distribution function. It is a real monotonically non-decreasing function (Robinson, 1980). Since the power spectrum is defined as

\[
B(f) = \frac{d\Lambda(f)}{df} , \tag{9}
\]

it follows from the properties of \( \Lambda \) that \( B(f) \) must be nonnegative.

In summary, a function \( b(\tau) \) is an autocorrelation of a real function if:

i. \( b(\tau) \) is an even function (symmetric about \( \tau=0 \)).
ii. \( b(0) \) is greater than or equal to any \( b(t) \).
iii. \( B(f) \) is nonnegative.

The power of Constrained Iterative Spectral Deconvolution lies in the application of a nonnegative definite constraint by enforcing nonnegativity in the frequency domain. To see the necessity of such a constraint, it is important to return to the subject of the unbiased autocorrelation estimate.

A biased autocorrelation estimate by definition meets the above criteria, including the nonnegative definite property. In contrast, the unbiased estimate may not meet these criteria. Although removing the bias from the
autocorrelation is desirable, especially for short datasets, the resulting estimate may not be an autocorrelation. This effect is best illustrated by example.

Figure one is a noise-free synthetic example consisting of twenty samples of two sinusoids (Ramaswamy and loup, 1989).

\[ x(t) = \sin(0.7 + 7\pi t / 32) + \sin(0.9 + 9\pi t / 32) \] (10)

A digitization interval of one is used. Figures two and three are the biased and unbiased autocorrelation estimates. Figure four is a 20 point Bartlett window. Note that the unbiased estimate has a maximum at \( t=0 \). Since the Bartlett window is an even function, the unbiased estimate is always even. Figures five and six are the power spectra of the biased estimate and unbiased estimate. Here the problem becomes apparent. Although the power spectrum of the biased estimate is nonnegative, the spectrum of the unbiased estimate has negative values, a clear violation of the nonnegative definite property. Note, however, that the spectrum of the unbiased estimate has better resolved the two frequency components of the model.
FIGURE 1

TWO SINE MODEL

\[ x(t) = \sin(0.7 + 7\pi t/32) + \sin(0.9 + 9\pi t/32) \]
FIGURE 2

BIASED AUTOCORRELATION OF TWO SINE MODEL

AMPLITUDE

SAMPLE
FIGURE 3

UNBIASED AUTOCORRELATION
OF TWO SINE MODEL

AMPLITUDE

SAMPLE
FIGURE 4

BARTLETT WINDOW
(20 POINT)
FIGURE 5

POWER SPECTRUM

FOURIER TRANSFORM OF THE BIASED AUTOCORRELATION ESTIMATE FOR THE TWO SINE MODEL.
FIGURE 6

POWER SPECTRUM

FOURIER TRANSFORM OF THE UNBIASED AUTOCORRELATION ESTIMATE FOR THE TWO SINE MODEL.
DECONVOLUTION

The purpose of this section is not to cover the extensive topic of deconvolution in detail, but to provide some basic definitions in the time and frequency domain before the specific method of van Cittert iterative deconvolution is discussed.

Given a function \( h(t) \) that represents the impulse response of any shift-invariant linear system, the output \( x(t) \) can be described as the convolution of \( h(t) \) and the input function \( f(t) \) (Kanasewich, 1981),

\[
x(t) = h(t) * f(t) = \sum h(t') f(t-t')
\]

Deconvolution is defined as the process of solving for \( f(t) \) in the above equation. This can be expressed as the equation,

\[
f(t) = h^{-1}(t) * x(t)
\]

Function \( h^{-1}(t) \) is defined as the inverse wavelet of \( h(t) \). Various techniques exist for arriving at this inverse, including polynomial division by Z-transform and least squares deconvolution (Kanasewich, 1981). Deconvolution can also be described in the frequency domain. Using \( X(\omega) \), \( H(\omega) \), and \( F(\omega) \) to denote the Fourier transforms of \( x(t) \), \( h(t) \), and \( f(t) \), respectively,

\[
F(\omega) = \frac{X(\omega)}{H(\omega)}
\]

This equation breaks down for \( H(\omega)=0 \). This case is discussed in some detail by loup and loup (1983).

Recall from equation 7 that the power spectrum of the biased autocorrelation estimate can be expressed as a convolution. Likewise, we can express the power spectrum of the unbiased estimate as a deconvolution,
\[ U(\omega) = B(\omega) * W^{-1}(\omega) \quad . \] 

(14)

In the time (lag) domain, we can express \( u(t) \) as a division of the biased estimate by the Bartlett window,

\[ u(\tau) = b(\tau) / w(\tau) \quad . \] 

(15)

Note that since the Bartlett window is by definition the same length as the biased autocorrelation estimate, there is no problem with a zero in the denominator.
VAN CITTERT DECONVOLUTION

Van Cittert (1931) developed the deconvolution method of successive substitutions, often referred to as iterative deconvolution. Several authors (Bracewell and Roberts, 1954; loup, 1968; Lacoste, 1982; loup and loup, 1983; Jansson, 1984) discuss this method and related concepts such as convergence criteria. The purpose here is to summarize van Cittert’s method and some key properties before discussing constrained iterative techniques.

In the previous section, we described the problem of deconvolution as solving for \( f(t) \) in the equation,

\[
x(t) = h(t) * f(t) \quad .
\]  

(16)

The van Cittert solution is given by loup and loup (1983) as

\[
\begin{align*}
f_0(t) &= x(t) \\
f_1(t) &= f_0(t) + [x(t) - f_0(t) * h(t)] \\
& \quad . \\
f_n(t) &= f_{n-1}(t) + [x(t) - f_{n-1}(t) * h(t)]
\end{align*}
\]  

(17)

where \( x(t) \) is the known filtered output, \( h(t) \) is the filter, and \( f_i(t) \) is the unfiltered (deconvolved) function for the \( i \)-th iteration.

These iterations can also be expressed as an equivalent multiplication in the frequency domain, where \( x, f, h \) are replaced by their Fourier transforms (not power spectra) \( X, F, H, \)

\[
F_n(\omega) = F_{n-1}(\omega) + [X(\omega) - F_{n-1}(\omega)H(\omega)].
\]  

(18)

This can also be written as (Bracewell and Roberts, 1954; Lacoste, 1982; loup and loup, 1983)

\[
F_n(\omega) = \{1 + [1 - H(\omega)] + [1 - H(\omega)]^2 + \cdots + [1 - H(\omega)]^n\}X(\omega) \quad .
\]  

(19)
loup and loup (1983) show that this sum can be rewritten as,

$$F_n(\omega) = \frac{(1 - [1 - H(\omega)]^{n+1})X(\omega)}{H(\omega)}.$$  \hspace{1cm} (20)

The relation in the braces is commonly referred to as the equivalent van Cittert window. For $|1-H(\omega)| < 1$, as $n$ goes to infinity, this function approaches the straightforward deconvolution result (equation 13).

In summary, the van Cittert approach to deconvolution is an alternate technique that in the limit yields identical results to a spectral division approach. Convergence can be demonstrated, given some limitations on the function $h(t)$ and its Fourier transform $H(\omega)$. An equivalent van Cittert window can be calculated for any number of iterations. As directly applied, it would seem easier to deconvolve using spectral deconvolution. However, the real power of this technique becomes apparent when constraints are applied between iterations, as in the Constrained Iterative Spectral Deconvolution method.
CONstrained iterative spectral deconvolution

Ramaswamy (1985) and Ramaswamy and loup (1989) describe the method of Constrained Iterative Spectral Deconvolution in some detail. The technique uses the van Cittert method to deconvolve the power spectrum, with a nonnegativity constraint applied in the frequency domain, to remove bias from an autocorrelation function. This section is a summary of their work.

Recall from equation 4 that we can represent the biased autocorrelation estimate in terms of the unbiased estimate by the relation

\[ b(\tau) = u(\tau)w(\tau) \]

(21)

where \( w(\tau) \) is the Bartlett window. Equation 7 shows the transform domain equivalent expression. This can easily be expressed as a van Cittert iterative deconvolution in either domain,

\[ u_n(\tau) = u_{n-1}(\tau) + b(\tau) - u_{n-1}(\tau)w(\tau) \]

\[ U_n(\omega) = U_{n-1}(\omega) + B(\omega) - U_{n-1}(\omega)*W(\omega) \]

(22, 23)

As we discussed in the previous section, as the number of iterations tends to infinity, the van Cittert solution approaches the result of applying the deconvolution of equation 15 directly. We demonstrated that this solution may not be desirable because it violates the nonnegative definite property of the autocorrelation function. Because the van Cittert method is an iterative one, a constraint can be applied ad hoc between iterations to force the autocorrelation solution to be nonnegative definite. This can be accomplished by setting negative values of \( U_n(\omega) \) (the power spectrum of the unbiased estimate) to zero at each iteration. The iterations can be accomplished in either domain, but it is a simple multiplication in the time (lag) domain (Amini, Chunduru, loup, and loup, 1992). Ramawamy and loup (1989) chose to do the van Cittert iterations in the frequency domain. The
iterations can be halted before too much bias is removed by monitoring the
the mean square error or some other error criteria (Chunduru, 1992).

It is quite interesting to consider the extension of the autocorrelation in
terms of the iterative equation versus the constraint application. Since both
the window function $w$ and biased estimate $b$ are zero for lags greater than $N$
points, the iterative equation in itself will not extend the autocorrelation.
Clearly, then, it is the constraint application that extends the unbiased
estimate $u$. Note that after $u$ is extended by the constraint application, the
iterative equation still does not modify these 'extended' values for $u$, nor do
they contribute to the error term. Recognizing this fact allows us to save
computer time by only calculating the iterations for the original $N$ points
instead of the longer extended dataset. In summary, the nonnegativity
constraint may modify the entire unbiased estimate, while the iterative
equation only modifies the original $N$ lags in the estimate. A flowchart of the
CISD method is shown in Figure 7.

A comparison of this method to a direct deconvolution illustrates the
power of applying the nonnegative definite constraint. The CISD method is
applied to the model in Figure 1. Figure 8 shows the extended
autocorrelation after 200 iterations. The corresponding power spectrum is
shown in figure 9. Note the vastly improved resolution of the two frequency
components over the power spectrum of the biased and unbiased estimates.
Figure 10 is an illustration of mean square error versus iteration number. The
CISD power spectrum is compared to the power spectrum of a 64 point
version of the two sine model in Figure 11.

Before discussing some refinements of this technique and its extension
to two dimensions, the CISD method will be compared to some other spectral
estimation approaches.
FIGURE 7
CISD FLOWCHART

BIASED AUTOCORRELATION ESTIMATE
\[ b(n) \]

AUTOCORRELATION WINDOW
\[ u(n) \]

CISD ITERATIVE EQUATION
\[
\begin{align*}
\hat{u}(n) &= b(n) \\
u(n) &= \hat{u}(n) + R[b(n) - u(n)u(n)]
\end{align*}
\]

FFT
\[ U(\omega) \]

SET NEGATIVE VALUES OF \[ U(\omega) \] TO ZERO
\[ U'(\omega) \]

FFT\(^{-1}\)
\[ u(n) \]

MEAN SQUARE ERROR
\[
MSE = \sum_{f} \left| b(n) - u(n)u(n) \right|
\]

ERROR CONVERGING

\[ N \]
STOP
FIGURE 8

CISD AUTOCORRELATION

CISD AUTOCORRELATION OF THE TWO SINE MODEL
FIGURE 9
CISD POWER SPECTRUM

FOURIER TRANSFORM OF THE CISD AUTOCORRELATION ESTIMATE FOR THE TWO SINE MODEL
FIGURE 10

MEAN SQUARE ERROR

MEAN SQUARE ERROR VS. ITERATION NUMBER FOR THE TWO SINE MODEL
FIGURE 11

POWER SPECTRA COMPARISON

POWER SPECTRUM OF 64 POINT MODEL

CISD POWER SPECTRUM OF TWO SINE MODEL
COMPARISON TO OTHER METHODS

Constrained Iterative Spectral Deconvolution is an important tool for the spectral analysis of short datasets. To illustrate its value in light of the many other techniques available, the CISD method can be applied to a 'standard' model.

Kay and Marple (1981) applied eleven spectral estimation techniques to a common dataset. The dataset itself is shown in figure 12, and its true power spectrum is shown in Figure 13. The model consists of three sinusiods with signal-to-noise ratios of 10, 30 and 30 DB, with frequencies of 0.1, 0.2, and 0.21 Hz respectively. In addition, band-limited noise was added to the model. The passband is centered on 0.35 Hz. Figures 14, 15, and 16 are power spectral estimates using the Blackman-Tukey, Burg Maximum Entropy, and the Prony Spectral Line Estimation methods. (Figures are adapted from Kay and Marple, 1981) The Blackman-Tukey method is simply the indirect method described in an earlier section. The Burg Maximum Entropy technique (Burg, 1975) is based on an autoregressive data model. It is a very common technique for spectral estimates of short datasets. Prony's method models the data as a linear combination of exponentials. These and several other methods are described in detail by Kay and Marple (1981). It should be noted here that Kay and Marple's stated intention is to illustrate the properties of each technique, rather than to compare their relative performance.

Figure 17 is a CISD power spectrum of the dataset. Note the resolution of the two closely spaced frequency components, and the estimation of the component at 0.1 Hz. The band-limited noise is also depicted quite accurately. The Blackman-Tukey method (Figure 14) does not resolve the two closely spaced frequency components. The Burg Maximum Entropy estimate (Figure 15) resolves the three peaks, but this method does not give their true amplitude. Also note that the Burg spectrum given by Kay and Marple (1981) does not depict the peak at 0.1 Hz. This is later corrected in Marple's book (1987). The spectrum in Figure 15 is the correct one. The Prony Spectral Line Estimation method (Figure 16) provides an impressive
approximation to the true spectrum (Figure 13); Kay and Marple describe it as the most accurate estimate of the true model. In both the Burg and Prony methods, a model order must be determined. In contrast, the CISD method is model-independent.

The Constrained Iterative Spectral Deconvolution method performs very well compared to other methods on this standard dataset. In fact, CISD was the only model-independent method that accurately depicted the true model. These results illustrate that the CISD technique is an important new method for the estimation of power spectra for short datasets.
FIGURE 12

KAY AND MARPLE DATASET

64 POINT REAL DATASET
(KAY AND MARPLE, 1981)
FIGURE 13

TRUE POWER SPECTRUM

Kay and Marple Model

AFTER KAY & MARPLE, 1981
FIGURE 14

POWER SPECTRUM

BLACKMAN/TUKEY POWER SPECTRUM
OF KAY AND MARPLE MODEL
16 AUTOCORRELATION LAGS
FIGURE 15

POWER SPECTRUM

BURG MAXIMUM ENTROPY POWER SPECTRUM OF KAY & MARPLE MODEL
FIGURE 16

PRONY SPECTRAL LINE POWER SPECTRUM

Kay and Marple Model
8 Coefficients

AFTER KAY & MARPLE, 1981
FIGURE 17

CISD POWER SPECTRUM

CISD POWER SPECTRUM OF
THE KAY & MARPLE MODEL
263 ITERATIONS
RELAXATION PARAMETER

A relaxation parameter can be introduced to the correction term in the van Cittert deconvolution to speed convergence (Jansson, 1984). The general idea is to increase the size of the correction term to allow more rapid convergence. The CISD equation is modified as follows,

\[ u_n(\tau) = u_{n-1}(\tau) + R[b(\tau) - u_{n-1}(\tau)w(\tau)] \]  \hspace{1cm} (24)

where \( R \) is the relaxation parameter. In this application, \( R \) is a constant. In general, \( R \) may be a function (Jansson, 1984).

To understand the relaxation parameter, convergence conditions of the van Cittert method must be discussed. It was shown by Bracewell and Roberts (1954) that the van Cittert equation converges for

\[ |1 - H(\omega)| < 1. \]  \hspace{1cm} (25)

Ramaswamy and Ioup (1989) have shown that the CISD method converges under the corresponding condition

\[ |1 - w(\tau)| < 1. \]  \hspace{1cm} (26)

Note that for a real, positive \( w(\tau) \), this condition becomes

\[ w(\tau) < 2.0 \]  \hspace{1cm} (27)

In the CISD method, \( w(\tau) \) is a Bartlett window normalized so the maximum value is unity. Because this window is by definition positive and real, the CISD equation itself converges; however, since non-linear ad-hoc nonegative constraints are applied between iterations, convergence has not been proven in general. The equivalent van Cittert window incorporating the relaxation parameter is described by Kawata and Ichiooka (1980).
The equivalent CISD window easily follows from this. It is derived in Appendix A. The equivalent CISD window for N iterations is

\[ u_n(\tau) = (1 - [1 - w(\tau)][1 - Rw(\tau)]^n)b(\tau) / w(\tau) \] \hspace{2cm} (28)

By inspection, the convergence condition is clearly

\[ |1 - Rw(\tau)| < 1. \] \hspace{2cm} (29)

For the Bartlett window as defined previously, the condition becomes

\[ 0. < R < 2. \] \hspace{2cm} (30)

Figure 18 illustrates this condition in the complex plane.

Figure 19 shows the mean square error versus iteration number for the Kay-Marple model for several values of R. Immediately obvious is the oscillation in the error curve for R=2.5, as expected by our convergence criteria. Note that the curve for R=2.0 is also beginning to oscillate. After 80 iterations, these two curves merge, indicating that although R values above 2.0 may converge, there is no benefit to the final solution. With the exception of some erratic behaviour in early iterations, larger R values converge more quickly, for R less than 2.0. Figure 20 is a plot of the number of iterations to reach a specific mean square error value (0.73) versus R value. Note that an R value of 2.0 reaches the target error in 50 iterations as compared to 100 iterations for R=1.0. The corresponding power spectra for these two solutions are indistinguishable. The relationship between R value and convergence can be expressed as:

\[ \frac{R_1}{R_2} \equiv \frac{N_2}{N_1} \] \hspace{2cm} (32)

where R is the relaxation parameter and N is the number of iterations to reach a given mean square error value. Since the method is non-linear, it may be beneficial to some datasets to slow convergence by using a small (less than one) R value. This paper will not address that particular issue.
The addition of a relaxation parameter to the CISD method speeds convergence by a factor of two while yielding results equivalent to the unmodified method for the datasets examined in this paper. In a completely hands-off approach, the relaxation parameter should be set to slightly less than two to maximize the convergence rate and avoid oscillation. The subroutine for 1-D Constrained Iterative Spectral Deconvolution, IDECON, contains this modification. It is included in Appendix B.
FIGURE 18

CISD CONVERGENCE CONDITION

$|1 - R_w(\tau)| < 1$

after Hill and Ioup, 1976
FIGURE 19
MEAN SQUARE ERROR

MEAN SQUARE ERROR VS. ITERATION FOR 5 VALUES OF THE RELAXATION PARAMETER R. (KAY & MARPLE MODEL)
FIGURE 20

ITERATIONS VS. R

NUMBER OF ITERATIONS TO REACH A MEAN SQUARE ERROR OF 0.73 FOR THE KAY AND MARPLE MODEL.
COMPENSATION FOR MISSING SAMPLES

In many real world applications, a sampled data sequence may have some missing samples. For example, seismic data has missing samples due to missing stations in the spatial domain. In the time domain, missing samples are caused by noise edits and initial blanking application. Although most spectral applications cannot compensate for missing samples, the CISD method can be modified easily to accomplish this.

Recall that the standard window function for the CISD technique is the Bartlett window

\[ w(\tau) = \frac{(N - |\tau|)}{N} \quad \tau = 0, 1, 2, \ldots, N - 1 \quad (32) \]

This window represents the bias inherent in the autocorrelation estimate for a N point time series. In fact, the Bartlett window is the autocorrelation of a N point time series consisting of unit spikes. This observation suggests an approach for datasets with missing samples. Instead of using the standard Bartlett window, a modified window should be used. This window is the autocorrelation of an N point time series consisting of unit spikes only where there are legitimate sample values in the data. (There can certainly be legitimate zero values in a dataset that do not represent missing samples.) This window correctly represents the bias in the autocorrelation estimate, and is the Bartlett window if all N samples are present. Figure 21 illustrates the calculation of this modified window.

The necessity of this modification is obvious for an extreme case where the biased autocorrelation \( b(\tau) \) has null samples due to missing values in \( x(t) \). If a Bartlett window is used in the CISD method, these samples will remain zero values in the unbiased estimate. Since \( w(\tau) \) is non-zero for the Bartlett window, \( u(\tau) \) must be zero to yield a zero value of \( b(\tau) \). This is clearly not a desirable result. If we use the modified window function, with a zero value corresponding to a zero bias value in \( b \), the unbiased estimate \( u \) is not constrained to be zero. Although the iterative equation itself will not alter this
particular u value, it may be restored by application of the constraint. Since the corresponding window value w is zero, this modification of u will not contribute to the error term. In a more practical case where the window is reduced due to missing samples, the constraints have a larger effect (Yoerger, 1978; Yoerger and Ioup, 1983).

A variation on the two-frequency model used previously demonstrates the value of this modification to the CISD method. Points 11-15 of this 20 point model were set to zero. Figure 22 shows the conventional power spectrum of this time series. Figure 23 shows the CISD power spectrum using a Bartlett window function. Note the remnant sidelobes in this estimate. Finally, figure 24 shows the CISD power spectrum using the modified window approach described above. Both figures 23 and 24 used 200 iterations.

A straightforward modification to the CISD method can be used to compensate for null values in the input dataset. The benefit is seen very clearly on a model dataset. This is an important result, because many spectral estimates in real world applications are plagued by sidelobes resulting from null values in the dataset.
This window is more desirable than the Bartlett window in that it represents the true relative bias for each point in the associated biased autocorrelation. Note that $W$ must be normalized so that the maximum value is unity in order to preserve area. If $x(t)$ has no missing samples then $h(t)$ is unity for all $N$ samples. The window function $w(\tau)$ then becomes the Bartlett window.

$$w(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} h(t)h(t-\tau)$$

$$w(\tau) = \frac{(N-\tau)}{N} \quad \tau=0,1,2,\ldots,N-1$$
FIGURE 22

POWER SPECTRUM

POWER SPECTRUM OF THE TWO SINE MODEL WITH MISSING SAMPLES
FIGURE 23

CISD POWER SPECTRUM

CISD POWER SPECTRUM OF TWO SINE MODEL WITH MISSING SAMPLES.
(BARTLETT WINDOW FUNCTION)
FIGURE 24

CISD POWER SPECTRUM

CISD POWER SPECTRUM OF TWO SINE MODEL WITH MISSING SAMPLES.
(COMPENSATED WINDOW)
EQUIVALENT CISD WINDOW METHOD

The standard CISD method involves the application of constraints after each iteration. Some authors (Whitehorn, 1981; Whitehorn and Ioup, 1981; Ioup and Ioup, 1983) have suggested doing multiple iterations between the constraint application. This can be accomplished by using an equivalent CISD window (Equation 28).

Appendix A derives the equivalent CISD window for a starting $u_0$ that could be any intermediate unbiased estimate. This relation is given below

$$u_\tau = \frac{b(\tau)}{w(\tau)}[1 - (1 - Rw(\tau))^n] + u_\tau [1 - Rw(\tau)]$$

(34)

Note that for a given $n$ and $\tau$, this expression can be simplified to

$$u = G + u_0 H$$

(35)

where $G$ and $H$ need be calculated only once.

This modification to the CISD method was implemented and tested. The subroutine IDECONN in Appendix B includes this modification. Initially, the implementation consisted of applying $n$ iterations between each constraint application, starting with the first iteration. However, this was not beneficial to convergence. The algorithm was then modified to allow a variable number of equivalent iterations according to a user input schedule of absolute iteration number versus number of equivalent iterations. The reasoning was that the constraint was needed more often on the earlier iterations where the estimate was changing significantly. However, this method was not beneficial to convergence either.

Figure 25 illustrates the effect of this modification on the mean square error for the two sine model. The first 100 iterations were done using a single 'length' window between constraint applications. The next 100 iterations used a window equivalent to two iterations between each constraint application.
Note how the error increases for a few iterations before it again begins to converge. For comparison, figure 10 shows the MSE curve for 200 iterations with a standard single 'length' window.

Recall that without the application of the nonnegative definite constraint, the iterative equation will calculate the unbiased estimate given by equation 3. This unbiased estimate is undesirable because it violates the autocorrelation nonnegative definite property; there are negative values in the power spectrum. Since the iterative equation does not affect the 'extended' values of the autocorrelation, the constraint application can be thought of as coupling the extended part of the estimate with the unextended part. Doing multiple iterations without constraints allows these 'parts' to become uncoupled. Apparently, without constraints, the solution converges very rapidly toward the undesirable unbiased estimate. After each single application of the iterative equation, the error decreases substantially. Then, after application of the constraint, the error increases as negative power is eliminated. Overall, the energy after each constraint application is decreasing. When multiple iterations, or their equivalent, are done between constraint applications, more negative power is introduced and must be removed via the constraint. The mean square error diverges for a few iterations until a balance is again achieved between the iterative equation error reduction and the 'rebound' in error introduced by the constraint. Figure 26 shows the mean square error curve for 99 single iterations with constraints, followed by a N=2 equivalent window application (two iterations) without the followup constraint application. Note the dramatic decrease in error, as compared to figure 10. Figure 27 is the CISD Power Spectrum associated with figure 26. As expected, a significant amount of 'negative power' has been introduced.

In summary, an approach was implemented that applied multiple iterations using an equivalent window between each application of constraints. This modification to the CISD method was harmful rather than beneficial to convergence. Decoupling of the 'extended' part of the unbiased estimate is likely the undesirable effect of this approach. Further study with other datasets and parameter choices (i.e., relaxation parameter) may yield
more desirable results. An 'optimization by simulation' study (Chunduru, 1992) may shed light on this interesting convergence behavior.
MEAN SQUARE ERROR VS. ITERATION NUMBER FOR THE TWO SINE MODEL. N=1 FOR FIRST 100 ITERATIONS. N=2 FOR ITERATIONS 101-200.
FIGURE 26

MSE VS. ITERATION

MEAN SQUARE ERROR VS. ITERATION NUMBER FOR THE TWO SINE MODEL.
N=1 FOR FIRST 99 ITERATIONS.
N=2 FOR ITERATION 100.
NO CONSTRAINTS APPLIED FOR ITERATION 100.
FIGURE 27

CISD POWER SPECTRUM

CISD POWER SPECTRUM CORRESPONDING TO FIGURE 26. NO CONSTRAINTS HAVE BEEN APPLIED FOR THE LAST ITERATION.
TWO-DIMENSIONAL CISD

The Constrained Iterative Spectral Deconvolution method has been demonstrated to be very effective for spectral estimation of very short datasets in the one dimensional case. The extension of CISD to two dimensions is straightforward. This section will first develop the CISD method in two dimensions. Then, the method will be demonstrated on a synthetic and a real dataset.

The two-dimensional biased autocorrelation function of a two dimensional sequence \( y(x,t) \) is defined as (Brigham, 1988)

\[
b(\xi, \tau) = \frac{1}{MN} \sum_{t=0}^{N-1} \sum_{x=0}^{M-1} y(x,t)y(x-\xi, t-\tau)
\]

(36)

Likewise, the two-dimensional equivalent of the Bartlett window can be described as

\[
w(\xi, \tau) = \frac{(N-|\tau|)(M-|\xi|)}{(MN)}
\]

(37)

The biased autocorrelation estimate in 2-D can be described in terms of the unbiased estimate \( u \) and window \( w \) as

\[
b(\xi, \tau) = u(\xi, \tau)w(\xi, \tau)
\]

(38)

The relationship of the power spectrum and autocorrelation can also be expressed easily in 2-D (Brigham, 1988)

\[
B(\kappa, \omega) = \sum_{t=0}^{N-1} \sum_{\xi=0}^{M-1} b(\xi, \tau)e^{-i\kappa\xi}e^{-i\omega\tau}
\]

(39)

Having established the two-dimensional autocorrelation and power spectrum, the two-dimensional CISD iterative equation easily follows

\[
u_n(\xi, \tau) = u_{n-1}(\xi, \tau) + R[b(\xi, \tau) - u_{n-1}(\xi, \tau)w(\xi, \tau)]
\]

(40)
The relaxation parameter $R$ is included here. It is straight-forward to alter the one-dimensional algorithm to the two-dimensional case. The FORTRAN program IDC2D in Appendix B implements this iterative equation, along with the nonnegativity constraint on the 2-D power spectrum.
APPLICATION TO MODEL DATA

A two-dimensional equivalent to the two sine model was built, using two sine functions in time, with slightly different velocities. The program used to generate this model is included in Appendix B.

Figure 28 is a T-X plot of the model. It consists of twenty points in time and 8 traces in offset. Figure 29 shows the autocorrelation of the model, and figure 30 shows the power spectrum. Note that the two distinct velocities are not resolved. After two-dimensional Constrained Iterative Spectral Deconvolution, the unbiased autocorrelation and CISD power spectrum are shown in figures 31 and 32, respectively. Note the resolution of the two velocity components.
FIGURE 28

TWO-DIMENSIONAL MODEL

OFFSET

SAMPLE (TIME)

TWO-DIMENSIONAL TWO SINE MODEL
FIGURE 29

TWO-DIMENSIONAL AUTOCORRELATION

2-DIMENSIONAL AUTOCORRELATION OF 2-D TWO SINE MODEL
FIGURE 30

TWO-DIMENSIONAL POWER SPECTRUM
MODEL DATA
FIGURE 31

2-D CISD AUTOCORRELATION

TWO-DIMENSIONAL CISD AUTOCORRELATION
OF THE 2-D TWO SINE MODEL
FIGURE 32

TWO-DIMENSIONAL CISD POWER SPECTRUM

MODEL DATA

FREQUENCY

WAVENUMBER

POWER
APPLICATION TO DOWNHOLE SONIC DATA

The 2-D CISD method was next applied to a real dataset from a downhole sonic tool. The dataset consists of eight traces, each with 1024 time samples. The digitization interval is 16 microseconds, and the trace spacing is 0.5 feet. As is typical with many seismic applications, this dataset is exceptionally narrow in the X domain. Figure 33 is a plot of the dataset. Figure 34 is the two-dimensional power spectrum of the data. After 2-D CISD, the resolution of the spectrum is greatly enhanced. Figure 35 shows the CISD power spectrum.

As a test of the missing sample compensation described earlier, traces five and six were set to zero, and these missing samples were reflected in the window function. Figures 36 shows the 2-D power spectrum of this modified dataset. Figure 37 shows the 2-D CISD power spectrum using a compensated window. This compares very favorably to the CISD spectrum derived from the entire dataset. Figure 38 shows the 2-D CISD power spectrum for the modified dataset using a standard 2-D Bartlett window. Note the undesirable sidelobe energy in this estimate.

The Two-Dimensional Constrained Iterative Spectral Deconvolution method has been demonstrated to greatly improve the resolution of both a synthetic and a real data example. The implementation is a straight-forward extension of the 1-D CISD method. The relaxation parameter and compensation for missing data samples were implemented and were shown to be effective.
FIGURE 33

DOWNHOLE SONIC DATA

TRACE SPACING: 0.5 FEET
DIGITIZATION INTERVAL: 16 MICROSEC.
FIGURE 34

TWO-DIMENSIONAL POWER SPECTRUM
DOWNHOLE SONIC DATA

FREQUENCY (HZ)

DB SCALE

WAVENUMBER (FT-1)
FIGURE 35

TWO-DIMENSIONAL CISD POWER SPECTRUM
DOWNHOLE SONIC DATA

FREQUENCY (HZ)

DB SCALE

WAVENUMBER (FT-1)
FIGURE 35

TWO-DIMENSIONAL CISD POWER SPECTRUM
DOWNHOLE SONIC DATA

WAVENUMBER (FT-1)

FREQUENCY (HZ)

DB SCALE

0.0
-7.1
-14.2
-21.4
-28.5
-35.6
-42.8
-50.0
FIGURE 36
TWO-DIMENSIONAL POWER SPECTRUM
DOWNHOLE DATA

Traces 5 and 6 are set to zero on the data.
FIGURE 37

TWO-DIMENSIONAL CISD POWER SPECTRUM

DOWNHOLE DATA

Traces 5 and 6 of the input data are set to zero. The window has been compensated for the missing points.

200 Iterations

R=1.0
FIGURE 38

TWO-DIMENSIONAL POWER SPECTRUM
DOWNHOLE SONIC DATA

Traces 5,6 are set to zero. The window function is a two-dimensional Bartlett window.
CONCLUSIONS

The Constrained Iterative Spectral Deconvolution technique is a powerful tool for power spectral estimation (autocorrelation extension) of short datasets. This study has extended the technique to two dimensions and has investigated two enhancements to the technique. A method was also developed to compensate for missing samples in the input dataset.

The extension of CISD to the two-dimensional case is important in that conventional two-dimensional spectra often suffer from short windows in the x (offset) domain. Application of this method to a real data case with only eight samples in the x domain showed a dramatic improvement in resolution. Since the algorithm is very straight-forward, any software utilizing a 2-D FFT can be easily modified to implement 2-D CISD. In contrast to other methods such as the Burg technique, CISD in not model-based..

Compensation for missing data samples is an important development in that many real datasets suffer from missing samples. Conventional spectral techniques often contain sidelobes generated by gaps in the data. The ability of CISD to compensate by using a specialized window function may be quite useful in many applications. This study demonstrated the power of this extension of CISD on both 1-D and 2-D datasets.

A relaxation parameter can be incorporated into the CISD method to improve the rate of convergence by a factor of two. A study of the convergence behavior of the modified CISD iterative equation yields a range of valid values for this parameter. Application of the modified technique to both 1-D and 2-D data yielded more rapid convergence with results indistinguishable from the unmodified result.

A CISD window equivalent to N iterations can be applied between each constraint application. Although this enhancement was implemented, there was no benefit in terms of convergence.
Possible future work could include an optimization study on datasets with noise. Also of interest is the effectiveness of the CISD method on more broad-band signals.
REFERENCES

von Cittert, P. H., 1931, Zum Einfluss der Spaltbreite auf die Intensitätsverteilung in Spektrallinien: Z. Physik., V. 69, 298-308.
Perspective: Proceedings of the IEEE, Vol. 69, No. 11, 1380-1419.
APPENDIX A

Derivation of the Equivalent CISD Window

A single window equivalent to the application of N iterations of the CISD technique can be derived (Ramaswamy and Ioup, 1988). This appendix derives the relation both with and without the relaxation parameter.

Recall the CISD iterative equation is given by

\[ u_n = u_{n-1} + b - u_{n-1}w = b + u_{n-1}(1 - w) \quad , \quad (1) \]

where \( b, u, \) and \( w \) are the biased autocorrelation, the unbiased autocorrelation, and the window function (Bartlett Window), respectively. Although not explicitly shown, these are all functions of \( \tau \). If we write a few iterations, the form of an equivalent equation is apparent

\[ u_1 = b + u_0(1 - w) \]
\[ u_2 = b + u_1(1 - w) = b + b(1 - w) + u_0(1 - w)^2 \]
\[ u_3 = b + u_2(1 - w) = b + b(1 - w) + b(1 - w)^2 + u_0(1 - w)^3 \]
\[ u_n = b \sum_{i=0}^{n-1} (1 - w)^i + u_0(1 - w)^n \]

The summation can be replaced by an expression given by

\[ [1 - (1 - w)^{n+1}] / w = \sum_{i=0}^{n} (1 - w)^i \quad , \]

yielding the resulting equivalent CISD window

\[ u_n = b[1 - (1 - w)^n] / w + u_0(1 - w)^n \quad . \quad (2) \]
Using the biased autocorrelation $b$ as an initial estimate for $u$, this relation becomes that given by Ramaswamy and Ioup (1989)

$$u_n = [1 - (1 - w)^{n+1}]b / w \quad . \quad (3)$$

Equation (1) can be written to include the relaxation parameter

$$u_n = u_{n-1} + R(b - u_{n-1}w) \quad . \quad (4)$$

Equation (2) then becomes

$$u_n = b[1 - (1 - Rw)^*]/w + u_0(1 - Rw)^* \quad . \quad (5)$$

If we again use $b$ as an initial estimate for $u$, equation (3) is modified to be

$$u_n = [1 - (1 - Rw)^{n+1}]b / w \quad . \quad (6)$$

This is equivalent to the result given by Kawata and Ichioka (1980).
APPENDIX B

FORTRAN PROGRAMS
subroutine ideconr(x,h,f,n,imin,imax,aerr,itnum,rval)

IDECON does iterative deconvolution in the time domain, with constraints applied in the frequency domain.

by Jerome L. Coggins, from an algorithm described by Ramaswamy and loup, 1989.

OTHER SUBROUTINE CALLS: FFT

Arrays x,h,f are stored as follows:
zero lag at point 1
positive lags at points 2 through N/2-1
negative lags at points N/2+1 through N
point N/2 is ambiguous between pos. and neg. maximum lag

x(n) is the autocorrelation of the input dataset
h(n) is the autocorrelation of the window function
f(n) is the deconvolved (output) autocorrelation
n is the number of points in x,h, and f.
imin is the minimum number of iterations
imin iterations will be done regardless of the mean square error
imax is the maximum number of iterations
iteration will continue to imax iterations, or until mean square error stops decreasing
aerr(imax) returns the MSE error curve
itnum is the number of iterations completed
rval is the relaxation parameter

OTHER SUBROUTINE CALLS: FFT

parameter (isize=1024)

real x(n),h(n),f(n),aerr(imax)
complex c(isize)

do 5 i=1,imax
   aerr(i)=0
5 continue

c initialize the deconvolved autocorrelation

   do 10 i=1,n
      f(i)=x(i)
10 continue

   iflag=0
c Begin Iterations
    do 20 i=1,imax
       write(6,'Iteration Number ',i
       if((iflag.eq.0).or.(i.le.imin)) then

       c CISD Iterative Equation
           do 25 j=1,n f(j)=f(j)+rval*(x(j)-f(j)*h(j))
           continue
           25

       c Move Current Estimate into complex array
           do 30 j=1,n
c(j)=cmplx(f(j),0.)
           continue
           30

       c Calculate Power Spectrum
           call fft(c,n,1)

       c Apply Constraint by zeroing negative values
           do 40 j=1,n
               if(real(c(j)).lt.0.) c(j)=cmplx(0.,0.)
           continue
           40

       c Inverse Fourier Transform
           call fft(c,n,-1)
           do 45 j=1,n
c(j)=c(j)/n
           continue
           45

       c Calculate the error
           aerr(i)=0.
           do 50 j=1,n aerr(i)=aerr(i)+*(x(j)-h(j)*real(c(j)))**2
           continue
           aerr(i)=aerr(i)/n
           write(6,'*Error= ',aerr(i)

       c Check for Convergence
           if(i.gt.imin) then if(aerr(i).gt.aerr(i-1)) iflag=1
           endif
           do 60 j=1,n
               f(j)=real(c(j))
           continue
           itnum=i
           endif

20 continue
    return
end
subroutine ideconN(x,h,f,n,imin,imax,aerr,itnum,numcrd,niter,nwin
+
, rval)

IDECON does iterative deconvolution in the time domain, with
constraints applied in the frequency domain.

**** this version allows the user to do Nnum 'iterations'
for each application of constraints

by Jerome L. Coggins, from an algorithm described by
Ramaswamy and loup, 1989.

OTHER SUBROUTINE CALLS: FFT

Arrays x,h,f are stored as follows:
- zero lag at point 1
- positive lags at points 2 through N/2-1
- negative lags at points N/2+1 through N
point N/2 is ambiguous between pos. and neg. maximum

x(n) is the autocorrelation of the input dataset
h(n) is the autocorrelation of the window function
f(n) is the deconvolved (output) autocorrelation
n is the number of points in x,h, and f.
imin the the minimum number of iterations
- imin iterations will be done regardless of the mean
- square error
imax is the maximum number of iterations
- iteration will continue to imax iterations, or until
- mean square error stops decreasing
aerr(imax) returns the MSE error curve
itnum is the number of iterations completed
numcrd is the number of 'cards' in the iteration schedule
niter(numcrd) is the maximum iteration associated with a
given 'anum' value
Anum(numcrd) is the number of iterations done between constraint
applications. (itnum refers to full iterations w/constraints)
rval is the relaxation parameter

OTHER SUBROUTINE CALLS: FFT

parameter (isize=1024)

real x(n),h(n),f(n),aerr(imax)
complex c(isize)
integer niter(numcrd),nwin(numcrd)
do 5 i=1,imax
   aerr(i)=0
5 continue

if (niter(numcrd).lt.imax)
niter(numcrd)=imax

! initialize the deconvolved autocorrelation

do 10 i=1,n
   f(i)=x(i)
10 continue

iflag=0
icrd=1
Anum=nwin(1)

! Begin Iterations
! do 20 i=1,imax
!   if (i.gt.niter(icrd)) then
!      icrd=icrd+1
!      Anum=nwin(icrd)
!   endif
!   write(6,*)'Iteration Number ',i
!   if((iflag.eq.0).or.(i.le.imin)) then

! Equivalent CISD Window Application

   do 25 j=1,n
      if(abs(h(j)).gt.1.e-20) then
         if((rval*h(j)).ne.1.) then
            f(j)=x(j)*(1-(1-rval*h(j))**Anum)/h(j)+f(j)*(1-rval*
            h(j))**Anum
         else
            f(j)=x(j)/h(j)
         endif
      else
         f(j)=0.
      endif
25 continue

! Move Current Estimate into complex array

   do 30 j=1,n
      c(j)=cmplx(f(j),0.)
30 continue
c Calculate Power Spectrum

    call fft(c,n,1)

c Apply Constraint by zeroing negative values

do 40 j=1,n
    if(real(c(j)).lt.0.) c(j)=cmplx(0.,0.)
40 continue

c Inverse Fourier Transform

    call fft(c,n,-1)
    do 45 j=1,n
        c(j)=c(j)/n
    45 continue

c calculate the error

    aerr(i)=0.
    do 50 j=1,n aerr(i)=aerr(i)+(x(j)-
        h(j)*real(c(j)))**2
    50 continue
    aerr(i)=aerr(i)/n
    write(6,*) 'Error= ',aerr(i)

c Check for Convergence

    if(i.gt.imin) then if(aerr(i).gt.aerr(i-1))
        iflag=1
    endif
    do 60 j=1,n
        f(j)=real(c(j))
    60 continue
    itnum=i
    endif
20 continue
return
end
subroutine idec2dr(x,h,f,m,n,mm,nn,imin,imax,aerr,itnum,rval)
   c
   IDEC2D does iterative deconvolution in the time-offset domain, with
   c constraints applied in the frequency-wavenumber domain.
   c
   x(m,n) is the autocorrelation of the input dataset
   c h(m,n) is the autocorrelation of the window function
   c f(m,n) is the deconvolved (output) autocorrelation
   c m is the number of traces (offsets)
   c n is the number of time samples
   c mm,nn are the actual dimensions of the 2-D arrays
   c imin the the minimum number of iterations
   c imax is the maximum number of iterations
   c aerr(imax) returns the MSE error curve
   c itnum is the number of iterations completed
   c rval is the relaxation parameter
   c
   OTHER SUBROUTINE CALLS: FFT2D
   c
   parameter (isize=256)
   real
   x(mm,nn),h(mm,nn),f(mm,nn),aerr(imax
   ) complex c(isize,isize)
   do 5 i=1 ,imax
      aerr(i)=0
   5 continue
   c initialize the deconvolved
   autocorrelation
   do 10 i=1 ,m
      do 11 j=1 ,m
         f(i,j)=x(i,j)
      11 continue
   10 continue
   iflag=0
   c Main loop of imax iterations
   do 20 i=1 ,imax
      write(6,*) 'Begin Iteration Number ',i if((iflag.eq .0).or.(i.le.imin)) then
         do 30 j=1 ,m
            do 31 k=1 ,n f(j,k)=f(j,k)+rval*(x(j,k)-
               f(j,k)*h(j,k)) c(j,k)=cmplx(f(j,k),0.)
         31 continue
      30 continue
c Do a 2-D FFT

call fft2d(c,m,n, isize, isize, 1, 1)

c Zero Negative Values

do 40 j=1,m
   do 41 k=1,n
      if(real(c(j,k)).lt.0.) c(j,k)=cmplx(0.,0.)
   continue
40 continue

c Inverse 2-D FFT and normalize

call fft2d(c,m,n, isize, isize, -1, -1)
a=m*n
do 50 j=1,m
   do 51 k=1,n
      c(j,k)=c(j,k)/a
   continue
50 continue

c calculate the error

aerr(i)=0.
do 60 j=1,m
   do 61 k=1,n
      aerr(i)=aerr(i)+(x(j,k)-
h(j,k)*real(c(j,k)))**2
   continue
60 continue

aerr(i)=aerr(i)/a
write(6,*) 'Error= ',aerr(i)

c Check for convergence

if(i.gt.imin) then
   if(aerr(i).gt.aerr(i-1)) iflag=1
endif
if(iflag.eq.0) then
   do 70 j=1,m
      do 71 k=1,n
         f(j,k)=real(c(j,k))
      continue
70 continue
itnum=i
endif
endif
20 continue
return
program model2sin
  c build a 20 point two-freq. model and its window
  character*30 datfile,winfile
  integer ioerr
  real dat(64),win(64),pi
  datfile='dh0:thesis/models/mod2sin.dat'
  winfile='dh0:thesis/models/mod2sin.win'
  pi=3.14159
  Amax=0.
  do 9 i=1,64
      dat(i)=0.
      win(i)=0.
  9 continue
  do 10 i=1,20
      dat(i)=sin(.7+7.*pi*(i-1)/32.)+sin(.9+9.*pi*(i-1)/32.) if (dat(i).gt.Amax)
      Amax=dat(i) win(i)=1.
  10 continue
  n=20
  call IOarray(dat,n,Amax,datfile,'write')
  Amax=1.
  call IOarray(win,n,Amax,winfile,'write')
  stop
end
program model2d4
  c build a model for use with decon2d program
  c uses two sinusoids with different velocities
  character*30 ofile
  integer ioerr
  real dat(32,32),pi,amax
  pi=3.1415926
  ofile='dh0:thesis/models/mod2d4.dat'
  amax=0.
  do 10 i=1,8
    do 20 j=1,32
      dat(i,j)=sin(.7+7*pi*(j-1)/32.)+sin(.8+8*pi*(j-1)/32.+i*pi/16.)
      if (abs(dat(i,j)).gt.amax) amax=abs(dat(i,j))
    20 continue
  10 continue
  caiii02D(dat,8,20,32,32,amax,ofile,'write')
  do 30 i=1,8
    do 40 j=1,32
      dat(i,j)=1.
    40 continue
  30 continue
  ofile='dh0:thesis/models/mod2d4.win'
  call IO2D(dat,8,20,32,32,1.,ofile,'write')
  stop
end
VITA

Jerome Leslie Coggins was born in Hopedale, Illinois, on June 6, 1958. He graduated with a Bachelor of Science in Engineering Physics from the University of Illinois in 1980. That same year, he joined Shell Oil Company in New Orleans as a geophysicist. In 1982, he began graduate studies at the University of New Orleans. After a transfer to Houston, Texas, in 1986, he completed his course work at the University of Houston. He is currently employed by Shell Oil Company in Houston as a Senior Geophysicist.