2012-07-20

Array-Based Characterization of Military Jet Aircraft Noise

David William Krueger
Brigham Young University - Provo

Array-Based Characterization of Military Jet Aircraft Noise

David W. Krueger

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kent L. Gee, Chair
Scott D. Sommerfeldt
Jonathan D. Blotter

Department of Physics and Astronomy

Brigham Young University

December 2012

ABSTRACT

Array-Based Characterization of Military Jet Noise

David W. Krueger
Department of Physics and Astronomy, BYU
Master of Science

Since the 1950s the jet aeroacoustics community has been involved in predicting and measuring the noise distribution in jets. In this work, cylindrical and planar Fourier near-field acoustical holography are used to investigate radiation from a full-scale, installed jet engine. Practical problems involving measurement aperture and the highly directional nature of the source are addressed. Insights from numerical simulations reveal usable reconstruction regions. A comparison of cylindrical and planar NAH for the respective measurement apertures shows cylindrical NAH outperforms planar NAH on reconstructions both towards and away from the source.

ACKNOWLEDGMENTS

Many people are responsible for my success as a graduate student at BYU. Inevitably, there will be many people who I forget to thank here. But I do want to point out a few choice friends that deserve recognition.

First, I want to thank Dr. Gee for his tireless and thorough zeal for quality research. His keen perspective on scope of a project and example of one-on-one instruction is something I hope to emulate. I want to thank Dr. Nielsen for her kindness and ability to dissect complicated theory and make it understandable. I also want to thank Dr. Sommerfeldt and Dr. Blotter for their insight and intuition. I thank Alan Wall, my lifeblood in this research and close colleague, for your energy and willingness to address an almost constant barrage of questions. I want to thank Jessica Morgan for happily tackling tricky problems. To my office mate Jarom Giraud, I give thanks for your example of dedication. To Matt Shaw, James Esplin, and Curtis Wiederhold, I thank you for your invaluable advice and friendship. To Dan Tengelsen, thank you for your experience and thought-provoking questions.

Most of all, I want to thank my sweet wife. Her many sacrifices have allowed me to spend long hours, weekends, and months of early mornings to earn this degree. Thank you, Karen. I couldn't have done it without you.

CONTENTS

List of Figures

1 Introduction

1.1 Background and motivation

Military aircraft maintainer personnel are exposed to high levels of noise produced in the jet exhaust region. Communities near air force bases are subject to loud noise during aircraft take-off and landing. Accurate characterization of the sound field in the exhaust can help reduce the sound levels and thus make the aircraft less of a hearing health riskand less of a community annoyance. The purpose of this work is to characterize the sound radiated from an F-22 Raptor.

Physical understanding and characterization of sound sources in turbulent jets is essential for construction of rigorous predictive models for jet noise emission. Beginning with theoretical work[1] in the 1950s, the jet aeroacoustics community has long been involved in predicting and measuring the noise source distribution in jets. Initial efforts utilized various acoustic far-field techniques such as the acoustic mirror,[2] acoustic telescope,[2-4] and polar correlation[2, 4, 5] methods. More recently, beamforming[6, 7] and other array-based methods[8] have been employed.

Experimental work to verify theoretical[1] and empirical models on a turbulent jet sound source has been performed in the acoustic far field of the source. These methods, such as beamforming,[6, 7] acoustic mirror,[2] acoustic telescope,[2-4] and polar correlation,[2, 4, 5] depend on an assumption that the source originates from uncorrelated, compact point sources. While beamforming and other methods can give source amplitudes and directivities, their accuracy is limited to a half-wavelength resolution. The most predominant sound energy of the turbulent

military jet, however, is produced at low frequencies around $100 - 200$ Hz , which have wavelengths of $3.3 - 1.6$ m $(10 - 5$ ft$)$.

Near-field acoustical holography (NAH) appeared in the 1980s,[9] and is typically used in the near-field of the source, on the order of centimeters, where evanescent wave information is high. Only recently has it been applied to aeroacoustic sources.[10-12] It potentially offers more accurate source information, especially at low frequencies because the spatial resolution is only limited by the microphone spacing – two microphones per wavelength. While NAH is typically used on correlated sound sources, it can also be used on partially-correlated sound sources,[13, 14] such as those found in the turbulent flow of jets. Also, it does not depend on the assumption of compact, uncorrelated sources, meaning the sources can have some width and spatial distribution. For these reasons, NAH was chosen as the processing technique in this work, even though the measurements, described in Chapter 2, were taken in the acoustic far field of the source.

Near-field acoustical holography and its various derivatives[15-17] are similar in concept: measure an acoustic field variable, typically pressure, over some surface and use analytic equations to propagate the measured acoustic field variable from one surface to another. The current work is to implement two methods, cylindrical NAH and planar NAH, which are both based on the Fourier transform. A discrete spatial Fourier transform is performed on the measurement plane to obtain the angular spectrum of the measured data in k-space, also called the wavenumber domain. The angular spectrum is then multiplied by a propagator to move from pressure at one location to pressure at another location. A multiplication in the wavenumber domain is equivalent to a convolution in the spatial domain. As multiplication tends to be simpler than convolution, Fourier NAH processing occurs in the wavenumber domain.

To assist in understanding the processes involved in implementing cylindrical and planar Fourier NAH, flowcharts are shown in Figures 1.1 and 1.2. Reference to these flowcharts will be made throughout this work to help the reader know what part of the NAH process is being discussed. There are a great many similarities between cylindrical and planar NAH. The underlying development and theory is analogous in each case. Indeed, the main difference is the change in coordinate system and, consequently, the propagators. The propagators and underlying theory for cylindrical and planar NAH are outlined simultaneously in Chapter 3. The differences between the two are highlighted as they arise in the discussion.
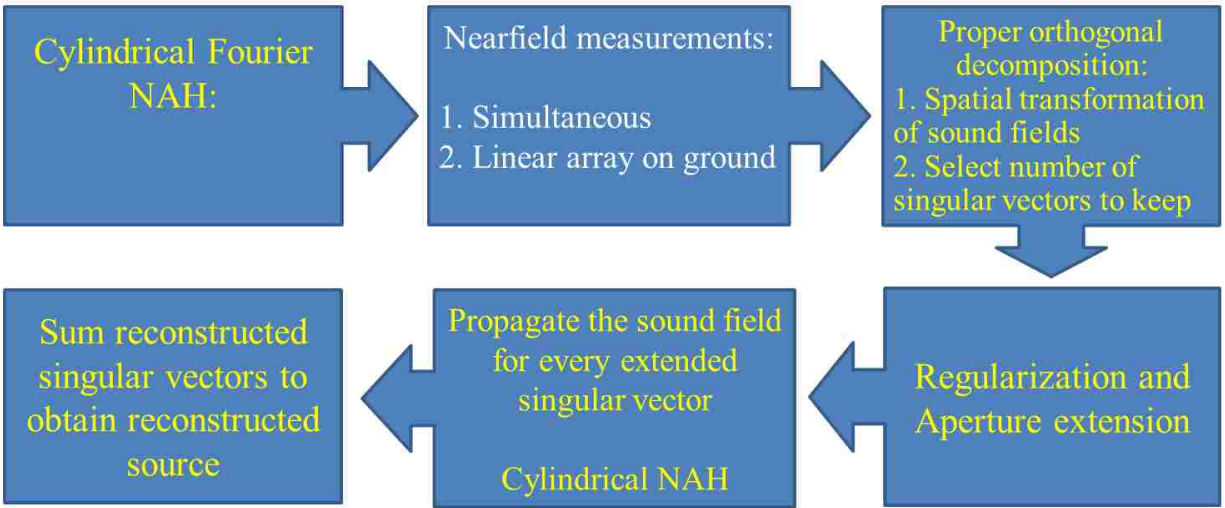
Figure 1.1 Flowchart of the process to implement cylindrical Fourier NAH. Yellow indicates work performed by the author. White indicates work performed by Wall.[1]
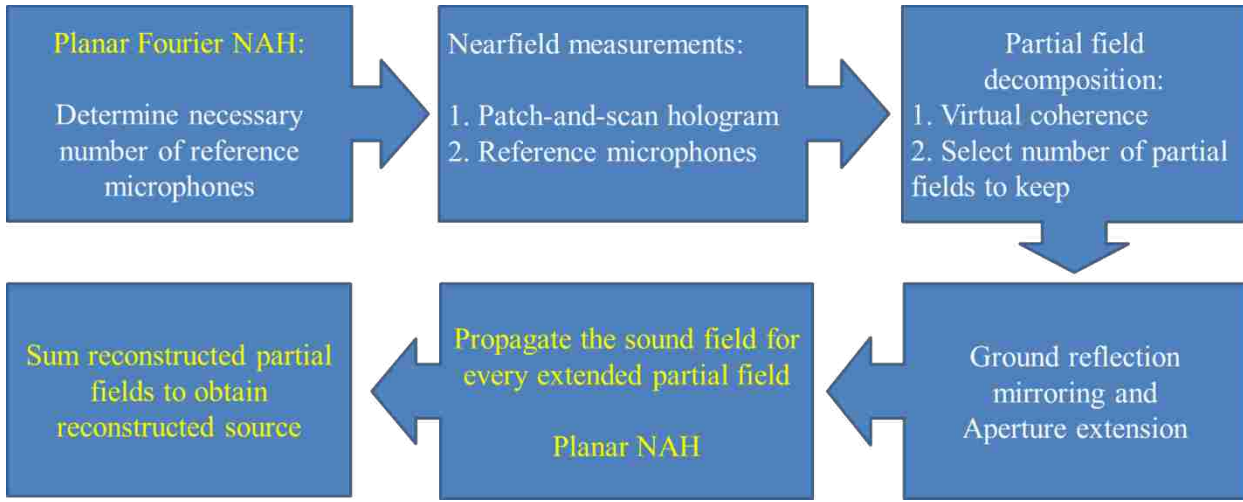
Figure 1.2 Flowchart of the process to implement planar Fourier NAH. Yellow indicates work performed by the author. White indicates work performed by Wall.

---

[1] Alan T. Wall is a colleague at Brigham Young University and is currently working on his doctoral degree. His work and the author's coincided substantially and collaboration was prolific. These flowcharts are intended to give both an overview of the NAH processes and offer distinction between who did what work.

To better understand the motivation of this work and to further establish the background, some of the challenges of Fourier NAH are now discussed. The first set of challenges deals with the measurement aperture, which is explained in Chapter 2. Fourier NAH reconstructs to the same spatial resolution with which it was measured. While this is not a direct disadvantage, it does pose feasibility limitations when considering the size of the jet noise source. Fourier NAH works best when the measurement aperture is much larger than the source. Here, "much larger" has reference to the aperture size such that acoustic levels along the edge of the aperture are small with respect to peak levels across the aperture. By having a large measurement aperture with respect to the source, when applying the discrete Fourier transform, wrap-around error will be reduced. In regards to the present work, the measurement aperture must be large with respect to the size of the jet noise source, which itself is on the order of tens of meters. This would require an enormous number of microphones to have both high spatial resolution and a large aperture with respect to the source.

In this work, ground reflections were addressed in two different ways, depending on the coordinate system. In cylindrical coordinates, the assumption of axisymmetry was made that the jet radiates sound equally in any direction from the jet centerline. This allowed the use of a linear array of microphones, as shown in Fig. 1.1.

Because of the size of the military jet and the limited number of microphones, a smaller portion of the sound field was measured. In cylindrical coordinates, a linear array of microphones was used, as indicated in Fig. 2.2, to simultaneously take measurements that spanned nearly 30 m. The linear array data were then smoothed and coherent pieces extracted using a process called spatial transformation of sound fields.[18, 19] In Cartesian coordinates (see Fig. 1.2), the aperture size was addressed using a scan-based NAH approach that allows

measuring the source in small sections, or patches, and stitching the patches back together using stationary reference microphones that have acquired data concurrently with every patch. This scan-based approach is called virtual coherence[20] and was used to obtain fully coherent measurement fields that are made up of patch-and-scan measurements.

Even using the long linear array of microphones and the patch-and-scan measurements, it was found in the course of this work that the measurement aperture was not large enough, especially when considering the directionality of the source. The overall sound pressure level has a maximum radiation of roughly 135° measured from the inlet.[21] That would extend the need for measurement scans even farther downstream. To such an end, each linear array measurement and stitched-together measurement plane was further extended using a process called analytic continuation[22] that uses the measured amplitude and phase to iteratively extrapolate the measured data while simultaneously reducing the amplitude at the extended edges.

The next set of challenges deals with the ill-posed nature of NAH. Near-field acoustical holography is an inverse problem that propagates an acoustical field variable in space towards the source. As an inverse problem, care must be taken to ensure the solution is unique and stable.[17] Equations that describe the evanescent wave components in the acoustic near-field of the source can dominate the field and become unstable very near the source. Consequently, regularization must occur when back propagating to ensure stability of the solution. Regularization is the process by which high wavenumbers associated with noise are filtered out and only wavenumbers pertaining to the signal remain. The regularization of the inverse problem is a multiplication in the wavenumber domain (instead of a convolution the spatial domain) and, thus, naturally occurs in k-space.[17, 23, 24]

Among the various regularization techniques, such as Tikhonov-based methods,[24, 25] Landweber iteration,[24] and the conjugate gradient approach,[24, 25] the Tikhonov-based methods have been most frequently adopted in recent work.[26] The regularization methods determine the shape of a k-space filter and their success depends on the cutoff frequency and the choice of the regularization parameter that governs the decay rate. Choice of the regularization parameter, in turn, depends on the variance of the measurement noise. If the variance of the noise measurement is known, then the Mozorov discrepancy principle (MDP)[24] is used. Alternatively, if the variance of the noise measurement is not known, then the generalized cross validation (GCV)[24, 27] or L-curve criterion[25] is used. In this work, a modified Tikhonov regularization method in conjunction with the Mozorov discrepancy principle serves as the basis of the k-space filtering during aperture extension, described in Section 3.3.

Fourier NAH has the ability to propagate both acoustic pressure and acoustic particle velocity. Both quantities are useful in characterizing the source because they are used in calculating acoustic intensity, which provides useful information on near-field energy flow and helps predict mid and far-field radiation. Consequently, part of this work will be to estimate the acoustic intensity through Fourier NAH. And the estimations will be compared to acoustic intensity measured by a tetrahedral intensity probe. Near-field intensity analysis of this sort has recently been performed[28] in a harsh environment similar to that of the military jet. A better understanding of the near-field source characteristics is essential to the eventual jet noise source reduction. Thus, understanding acoustic intensity as a function of frequency can provide insight into time-averaged acoustic energy flow.

1.2 Thesis outline

This thesis treats the use of Fourier NAH in cylindrical and Cartesian coordinates to characterize full-scale military jet noise sources. Acoustic intensity is also addressed by direct measurement using an intensity probe and by indirect estimation using NAH. The measurement setup is first outlined in Chapter 2 to provide a point of reference for and better understanding of some assumptions made in the NAH processing. Discussion of the processing theory included as Chapter 3 follows the same order as the flowcharts in Fig. 1.1 and Fig. 1.2. It discusses regularization and covers the method of analytically extending the measurement aperture. Because the theoretical background between cylindrical and planar NAH is so similar, Chapter 3 details the theory for both NAH methods simultaneously, highlighting their differences along the way. Chapter 4 discusses numerical simulations examining the NAH processing and gives understanding for regions of validity in the NAH reconstructions. Chapters 5 and 6 give pressure and intensity results, respectively, obtained through NAH on measured jet data. Comparative intensity probe results are also given in Chapter 6. A summary, conclusion and future work are treated in Chapter 7. An appendix is given at the end of the document, showing the NAH codes with dependent sub-functions generated in MATLAB[®]. Also included in another appendix is discussion on another patch NAH method, One-step patch NAH,[29] and why it was not used in this work.

2 Measurement Setup

A near-field, acoustical test was conducted from July 27 through July 30, 2009, on the F-22 ground run-up pad at Holloman Air Force Base in New Mexico. With over 6000 measurement positions, this data set constitutes the largest and most detailed ever recorded in the near field of a high-power military jet. Adding to its significance is the fine spatial resolution of 6-in along a measurement plane, the aperture extent, and the repetition of the measurements for four engine conditions ranging from idle to full afterburner. Figure 2.1 shows the 5×18 microphone array held on an adjustable rig in the geometric near field of the F-22A. The adjustable rig was set at three different heights giving a 1.83 m vertical span of the sound field. The rig was moved to 10 locations downstream along a guide rail during engine cool-down to give a downstream span of 22.9 m. Also shown in Fig. 2.1 is the tetrahedral intensity probe on top of the rig. Once an entire plane of data was taken, the guide rail was moved to one of three measurement planes – two parallel to the shear layer and one parallel to the centerline. No guide rail was used along the 23-m arc. The shear layer angle of 8.5° is an estimate based on engine exhaust hazard zone for military jet aircraft. Fig. 2.2 shows the measurement planes and the 23-m arc. The center of the microphone array is shown as triangles. Also shown as blue dots in Fig. 2.2 are 52 ground-based, fixed reference microphones spaced 0.61 m apart and spanning nearly 30 m.[30] The ground-based reference microphones will be used in the cylindrical NAH processing, and measurement plane 2 will be used in the planar NAH processing.

One important characteristic of the jet noise is its stationarity on subsequent run-ups. For scans within a measurement plane to be comparable to one another, the source must remain relatively constant both in frequency content and overall sound pressure level. Figure 2.3 shows

one-third octave band spectra[31] at a stationary microphone. Standard deviations averaged over many scans indicate the degree of stationarity.

Figure 2.4 shows the same information as Fig. 2.3 but for every reference microphone. The stationarity of the source on subsequent run-ups gives confidence in the ability to make detailed sound pressure level maps involving every measurement scan as in Fig. 2.5. Furthermore, spatial maps of the overall sound pressure levels at every measurement position can give a realistic idea of the overall sound propagation, as in Fig. 2.5 for afterburner engine condition. Figure 2.3 through Fig. 2.5 initially appeared in Ref.31.

The frequencies of interest in this study are limited to 105 Hz and 210 Hz. As shown in equation (3.11), the 2-ft spacing of the ground microphone array limits cylindrical NAH analysis to less than $f_{max} = c/2\Delta z = 343/2(2 \cdot .3048) = 281$ Hz. Also, from Fig. 2.3, the maximum sound pressure level occurs somewhere between 100 and 200 Hz. Consequently, we focus on 105 Hz and 210 Hz.

Figure 2.1 F-22A and 90-microphone array on adjustable-height rig. During engine cool-down, the rig was moved along the guide rail to 10 locations downstream.
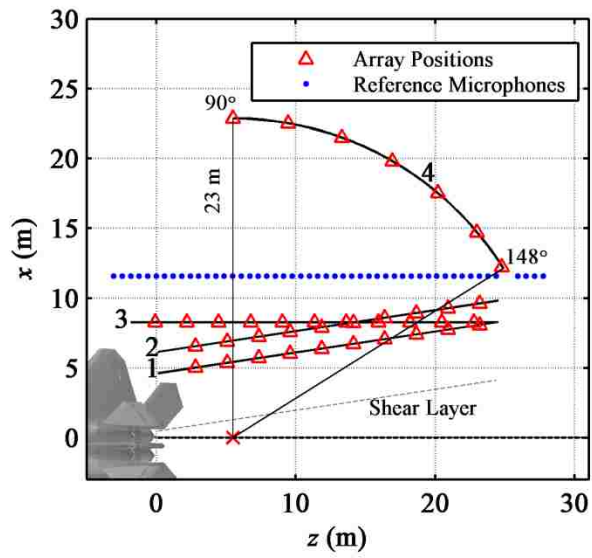
Figure 2.2 Measurement schematic showing the 23‐m arc and three measurement planes. The triangles indicate the rig center locations. The blue dots mark the locations of the ground reference array.
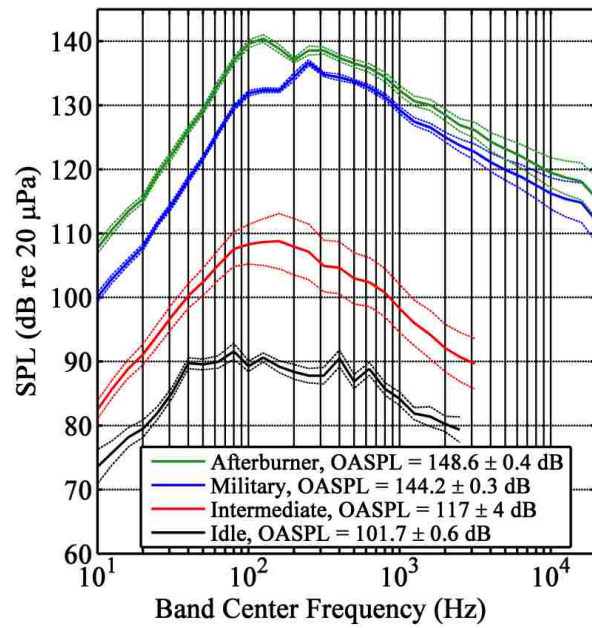
Figure 2.3 Overall sound pressure levels in one-third octave spectra a sationary reference microphone at various engine conditions. Dashed lines are the standard deviation averaged over many scans.
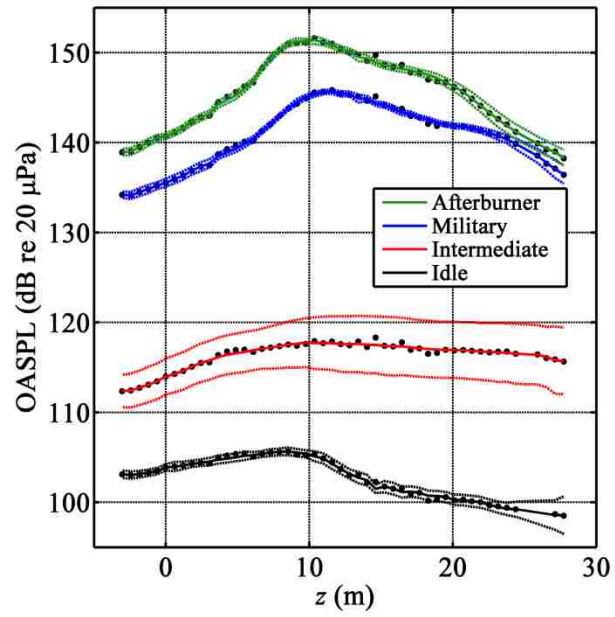
Figure 2.4 Overall sound pressure levels in one-thrid octave bands for all stationary reference microphones. Dashed lines are the standard deviation averaged over many scans.
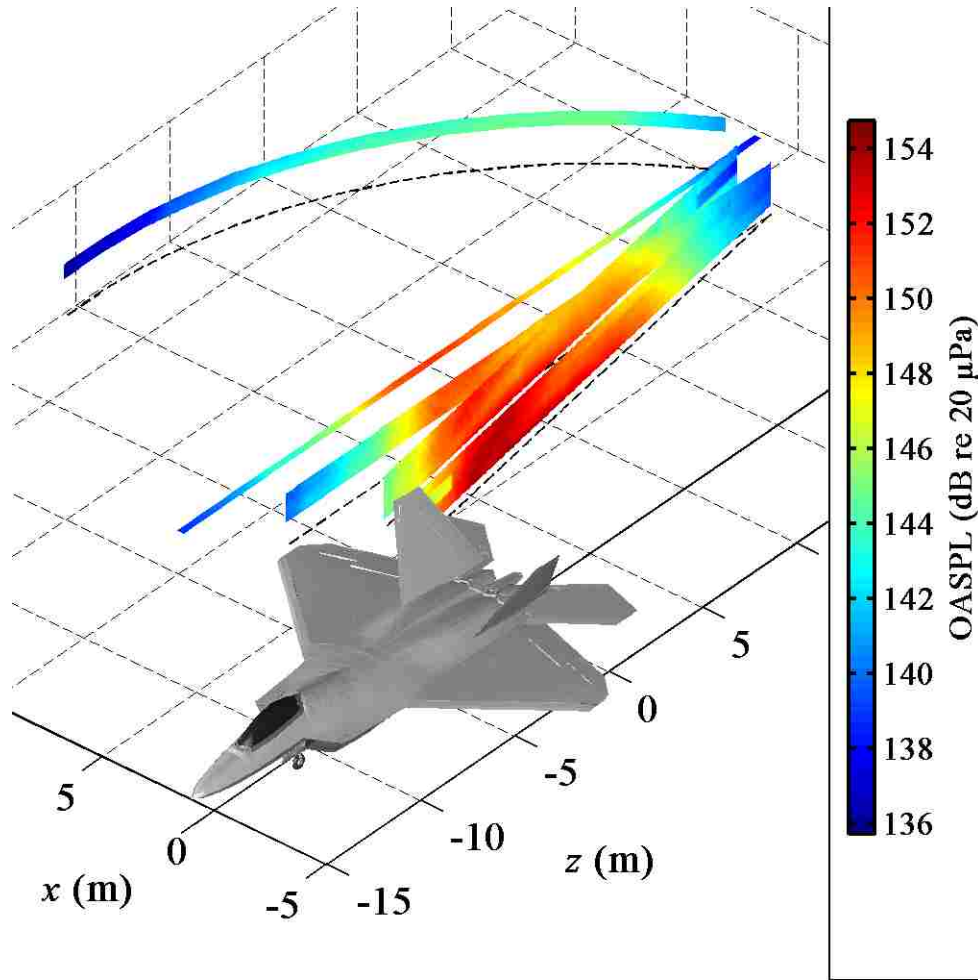
Figure 2.5 Overall sound pressure level recorded at afterburner at every measurement position.

3 Theory

3.1 Overview

Before delving into the theory of NAH processing, this section will give an overview. Conceptually, near-field acoustical holography can be loosely compared to an optical hologram: froma two-dimensional image, a three-dimensional projection can be created. An acoustical quantity such as pressure is measured over some conformal region in a separable geometry. A separable geometry is necessary for Fourier transform-based NAH because it offers a closed-form solution to the wave equation. Other boundary element methods[32, 33] do not require a separable geometry. Using the measured amplitude and phase information, the sound can be mathematically propagated to create a two- or three-dimensional sound field. When evanescent waves are present, an essential requirement is that the measurement of the sound occurs close enough to the source of interest, typically on the order of centimeters, to capture the evanescent material. This latter fact brings the term near-field to arrive at the name near-field acoustical holography.[9, 17]

An important distinction between traditional implementation of NAH and the implementation used in this work is the standoff distance. Traditionally, standoff distance from a source is on the order of centimeters. As described in Chapter 2, the standoff distance in this work was on the order of meters. Very little, if any, evanescent information was recorded in these measurements. Indeed it could be said this work is acoustical holography on far-field measurements. However, the prefixture near-field is maintained in this work because the

processing comes from near-field acoustical holography. Furthermore, NAH processing is still the method of choice for this work because, as discussed in Section 1.1, it is more accurate at low frequencies than other far-field techniques; it can be used on partially-correlated sound sources, which the jet is; and it does not depend on the assumption of compact, uncorrelated sources as do other far-field techniques, meaning the sources can have some width and spatial distribution.

As a discussion on NAH theory and its implementation, the requirements for NAH are discussed throughout this chapter in the following order:

- Minimum two microphones per wavelength.

- Coherent measurement.

- Appropriate regularization scheme for propagation towards the source.

- Measurement aperture large with respect to source size.

Near-field acoustical holography requires a coherent measurement over the entire measurement plane. The coherent measurement requirement for NAH necessitates a fixed-phase relationship between every point on the measurement hologram. There are two ways to achieve this: measure a small number of points and piece them together or measure a large number of points simultaneously.

In this project, for planar Fourier NAH in Cartesian coordinates, a patch-and-scan measurement has been employed which measures a small number of stationary points (a patch) at a time, moves to a new location and measures again, in this way "scanning" along the measurement aperture. The scans are then pieced together. In the scan-based approach, the discontinuities in phase information between scans may be accounted for with an array of fixed microphones that measure sound pressures simultaneously with each scan, which may then be

used to tie together the phases using partial field decomposition (PFD). Virtual coherence[20, 34, 35] is the PFD approach used in this work to synchronize the amplitudes and phases of each scan for planar Fourier NAH and is described further in Section 3.2.2.

Cylindrical Fourier NAH used in this work does not use a patch-and-scan technique. Instead, it is performed on measurements taken simultaneously in a linear array placed on the ground. Because the linear array measurements were taken simultaneously, the coherent measurement requirement is inherently met and virtual coherence is unnecessary. However, a preprocessing technique called spatial transformation of sound fields[11, 18] (STSF) was used in order utilize the entire measurement duration, reduce noise, and perform averaging over every time block. Spatial transformation of sound fields is a PFD-like technique that decomposes the information from the time blocks of the one-dimensional stationary reference array into an orthonormal basis set, a subset of which is used in cylindrical NAH processing. STSF is explained in greater detail in Section 3.2.1.

The two methods of NAH described in this work are based on the spatial discrete Fourier transform (DFT) in cylindrical and Cartesian coordinates. First, the pressure is measured over some constant dimension. For this work it was constant r and φ for cylindrical NAH and constant x for planar NAH. Once the pressure is measured and averaging through STSF has occurred for cylindrical NAH or the scans stitched together through virtual coherence for planar NAH, a one-dimensional spatial Fourier transform is computed to obtain the angular spectrum which resides in k-space. The foundational principle of Fourier NAH is that once the angular spectrum is known, the entire pressure or velocity field in a source-free region can be calculated without additional information. Hence, cylindrical and planar Fourier NAH allow us to calculate the pressure and velocity fields external to a source. Williams[17] offers a rigorous derivation of NAH

in these two coordinate systems, a brief summary of which will be offered in Section 3.5. Once the singular vectors from STSF (or partial fields from virtual coherence) have been propagated using cylindrical (or planar Fourier NAH), they are added together energetically, in the case of reconstructed pressure, or linearly in the case of reconstructed intensity vectors, to obtain the reconstructed source information.

3.2 Partial field decomposition

The first step in preprocessing for both cylindrical and planar NAH is to implement a partial field decomposition technique. A different decomposition was used in each NAH method. While the two decompositions are similar in that they depend on a singular value decomposition, their differences are enough to warrant separate explanation.

3.2.1 Spatial transformation of sound fields

The implementation of cylindrical NAH in this work utilizes the linear array of microphones located on the ground at a constant $r = 11.6$ m from the jet centerline instead of a two-dimensional array. This approach simplifies the propagation problem considerably, but requires an assumption of axisymmetry, which may not be appropriate at high frequencies or for a noncircular nozzle.

The procedure outlined here is called spatial transformation of sound fields (STSF),[11, 18] which is essentially a time block-wise partial field decomposition of the field. Even though the microphones in the linear array recorded simultaneously, STSF is used to utilize the full duration of the measurement, allowing for coherent fields with data from all time blocks. In essence,

information from every time block is decomposed into an orthogonal basis set with the same number of vectors as original time blocks. A subset of the basis set contains the most relevant measured information. The decomposition provides averaging across the time blocks, thus reducing noise. Also, because the subset is fewer in number than time blocks, fewer propagations are performed, and STSF saves on computation time, albeit very small. STSF relies on the singular value decomposition (SVD) to decompose the measurement time blocks into mutually orthogonal but individually coherent subspaces composed of singular vectors and scaling singular values. The individually coherent characteristic of each singular vector is important since NAH requires a complex (coherent) spatial pressure map.

The STSF process works as follows: First, a full cross spectral matrix (CSM) is computed as the outer product between all microphone pairs, as in Eq. (3.1),

$$K_{ij}(\omega) = \frac{1}{m} \sum P_i(\omega) P_j(\omega), \tag{3.1}$$

where m is the number of time blocks. Then the CSM is divided into singular vectors and singular values via a SVD. The singular vectors for 105 Hz at military power engine condition are shown as columns in Fig. 3.1. The number of singular vectors and singular values equals the number of reference microphones, and the first singular vectors will contain information relevant to the source, and the rest will contain lower amplitude noise. The SVD also orders the singular values from largest to smallest. The singular values provide an estimate to how much energy is included in each singular vector. Typically the singular values are monotonically decreasing, but after a certain point, including more singular values, each with its associated singular vector, does not change the final result because the contribution of comparatively small singular values is negligible. In the current work, only the singular vectors corresponding to singular values within 25 dB down from the largest singular value were included in the NAH processing.

Limiting the number of singular vectors that are used helps remove noise-related high

wavenumber components and thereby smoothes the data. Figure 3.2 shows the singular values

associated with the singular vectors of Fig. 3.1, the vertical line denoting the 25 dB down cutoff.
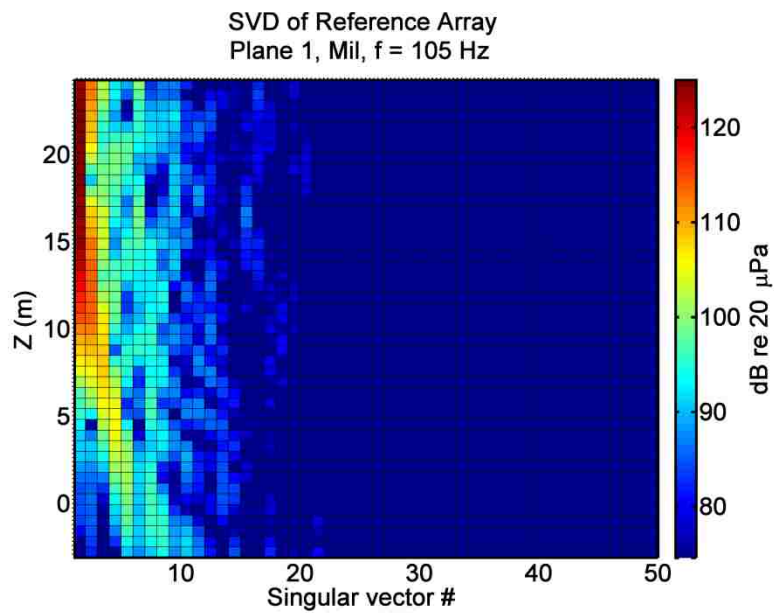
Figure 3.1 Singular value decomposition of cross-spectral matrix generated from ground reference microphones. The columns constitute the singular vectors. The SVD orders them left to right from greatest to least amplitude.
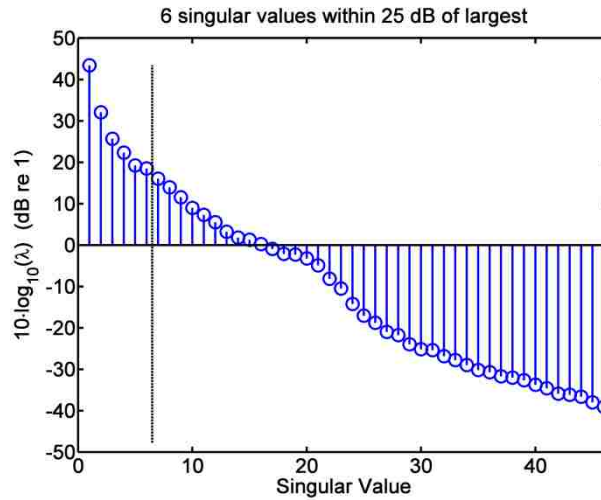
Figure 3.2 Singular values resulting from the singular value decomposition of the cross-spectral matrix generated from the ground reference microphones. Each singular value is associated to and scales a distinct singular vector. The vertical line shows the 25 dB down cutoff.

3.2.2 Virtual coherence

Now we discuss the partial field decomposition technique used in the planar NAH processing. Because the source is large with respect to the measurement aperture, patch-and-scan measurements were used to create a dense measurement plane. However, sharp phase discontinuities exist between each scan because they were not acquired simultaneously. Virtual coherence is a partial field decomposition (PFD) method which ties together the various scans to obtain wholly coherent partial fields which individually have a fixed phase relationship.[20, 34, 35]

As does STSF, virtual coherence utilizes a singular value decomposition (SVD) on the signals measured by the reference microphones. But now the orthogonal basis set is used as a set of "virtual references" to tie together the magnitude and phase of each scan. Each virtual reference contains information from all the individual physical reference signals. The singular

values that describe the strength of each of these virtual references are sorted in descending order. The measurement hologram is then decomposed into partial fields, each of which is fully correlated with one virtual reference. Therefore, the partial fields are also sorted by strength. This is mathematically the "ideal" decomposition, since as much of the sound field as is possible is included in the first partial fields. Care is taken in this document to distinguish between the partial fields of virtual coherence and the singular vectors in STSF. While both are results of the SVD, partial fields will refer to the two-dimensional results during the virtual coherence method, and singular vectors will refer to that of the one-dimensional STSF.

Similar to STSF, the total number of partial fields that come out of the virtual coherence decomposition will equal the number of reference microphones. The first partial fields will contain information relevant to the source, and the rest will contain lower amplitude noise. Therefore, a sufficient number of partial fields must be selected to reconstruct the source, and the rest discarded. Limiting the number of partial fields that are used can help remove high-wavenumber components and thereby smooth the scan-based data.

If there were a single monopole source or a distribution of monopole sources, the entire source would be correlated. Only one partial field will contain relevant information. Consequently, only one reference microphone is needed to perform PFD. A sound field generated by N independent sources will require N reference microphones, and will be decomposed into N partial fields. More reference microphones may be used, producing more partial fields, but only the first N will contain useful information. In practice, using more reference microphones reduces noise and improves the PFD.[20] If the number of sources is unknown, the singular values of the SVD on the reference microphones may be observed. For N independent sources, there will be a sharp drop from the singular value N to the $N + 1$ singular

value. For a jet, the number of independent sources is unclear. The singular values tend to decrease somewhat steadily and monotonically. The number of partial fields and the minimum number of reference microphones required to fully measure the source must be determined. The virtual coherence method provides a way to determine this number.

For a chosen frequency three cross spectral matrices are calculated: one containing cross spectra between each virtual reference, another containing the cross spectra between each reference microphone, and a third containing cross spectra between each virtual reference signal and each measured hologram microphone signal. These are, respectively, C_{vv}, C_{pp}, and C_{vp}. Here, a subscript v denotes a virtual reference, and a subscript p denotes a hologram measurement position. The virtual coherence between the ith virtual reference and the jth measurement position in each scan is given by

$$\gamma_{j,i}^2 = \frac{\left|C_{v_i p_j}\right|^2}{C_{v_i v_i} C_{p_j p_j}}. \tag{3.2}$$

For perfect coherence between the same frequency for two signals, $\gamma_{j,i}^2 = 1$, and $\gamma_{j,i}^2 = 0$ would denote no relation. To select the number of partial fields used for NAH, this virtual coherence is summed over the first R elements of i, iteratively increasing R until the coherence criterion is met, namely

$$\sum_{i=1}^{R} \gamma_{j,i}^2 \geq \text{coherence criterion.} \tag{3.3}$$

Once the coherence criterion is reached for every measurement position j in a scan, the R value is the necessary number of partial fields for that scan. The median of these R values is selected as the number of partial fields that are processed using NAH. In practice, a coherence of unity often requires a large number of partial fields. Therefore, to reduce the number of required partial

fields while still maintaining good coherence, a coherence criterion of 0.9 is used in this work. This corresponds to a signal to noise ratio of approximately 10 dB by the relation

$$\text{SNR} = 10 \log_{10}\left(\frac{\gamma^2}{1-\gamma^2}\right), \qquad\qquad (3.4)$$

where the numerator in the log function represents the coherent power, and the denominator corresponds to noise or incoherent power.

Results for decomposing the measured field at 105 Hz, military power engine condition are shown in Fig. 3.3. Four partial fields are necessary to meet the coherence criterion. Also shown is the phase of the measurement scans and the first four partial fields.Notice in c) the phase across the aperture is initially disjointed then in d) becomes more smoothly varying after virtual coherence.
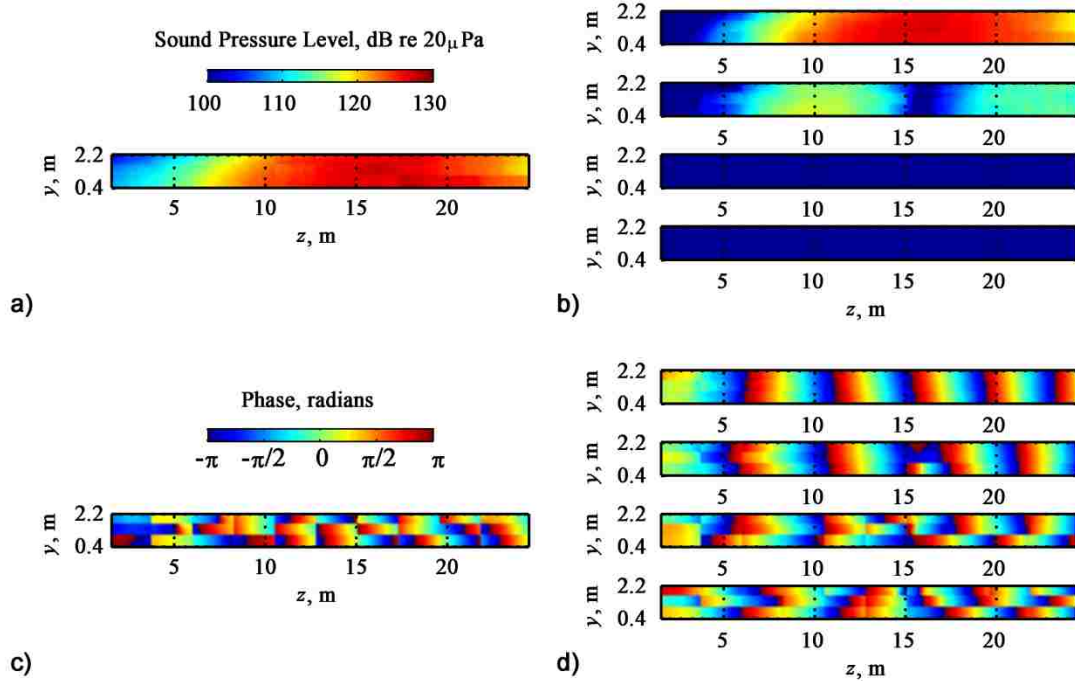
Figure 3.3 Measurement and PFD results for 105 Hz, military engine condition: a) SPL at measurement hologram; b) first 4 partial fields; c) phase of measurement scans, d) phase of first 4 partial fields .

3.3 Regularization using Modified Tikhonov with Morozov discrepancy principle

Because following steps in NAH processing depend on regularization, it will now be discussed. As mentioned previously, when back-propagating NAH becomes an inverse problem, and high wavenumbers can cause the reconstructions to blow up or become undefined. Some high wavenumbers are associated with noise. Other high wavenumbers are artifacts of the discrete Fourier transform used in Fourier NAH. Still other high wavenumbers may pertain to actual evanescent waves found in the near-field of the source. It becomes necessary, then, to

distinguish between noise and evanescent waves of the source, which is at the heart of the need for regularization.

Regularization is the process by which wavenumbers associated with noise are filtered out. It most logically occurs in k-space and is typically a multiplication of some type of filter centered about zero with a strategically chosen cutoff wavenumber. The difficulty then comes in choosing which type of filter and cutoff wavenumber to use. There is a vast knowledge base[23, 25] regarding filter design and filtering techniques. One filter used frequently in NAH,[17, 24] modified Tikhonov regularization, will be discussed in the following sub-section.

While there is, as Williams[24] mentions, no "holy grail" to regularization, Tikhonov regularization provides one possible estimate to the low-pass filter that should encode the physics of the radiation mechanism and decay slightly faster than the evanescent waves. There is a large body of literature dealing with Tikhonov regularization,[25] and a paper by Williams[24] treats its application to NAH. The details will not be repeated here, but the derived low-pass filter will be extracted [see Eq. (56) of Ref. 24]:

$$F^{\alpha} = diag\left(\cdots \frac{|\lambda_i|^2}{|\lambda_i|^2 + \alpha(\alpha/(\alpha+|\lambda_i|^2))^2} \cdots\right), 1 \le i \le M, \tag{3.5}$$

where α is the decay rate parameter, M the total number of singular values and λ_i are given by Eqs. (15) and (16) of Ref. 24 depending on the coordinate system as follows. For Cartesian coordinates,

$$\lambda_{ij} = \frac{\rho c k}{\sqrt{k^2 - k_{pij}^2}} e^{i\sqrt{k^2 - k_{pij}^2}d}, \tag{3.6}$$

and for cylindrical coordinates,

$$\lambda_{jn} = \frac{j\rho c k}{k_{rj}} \frac{H_n(k_{rj}r_h)}{H_n'(k_{rj}r_s)}.$$ (3.7)

To determine the decay parameter α we use the Morozov discrepancy principle (MDP), which works under the assumption that the filter is tapered so that it matches the decay with increasing wavenumbers of the evanescent waves and thus is successful in removing the random noise in the measurement. The Morozov discrepancy principle depends on a knowledge of the noise variance, σ. This is unknown in our problem. But we approximate the noise variance[22] by assuming the taper of the filter matches the evanescent wave decay with increasing wavenumber. The highest wavenumbers above the break point then correspond only to noise in the data. Thus, the expected value E of the last few basis vectors operating on the noise ϵ yields

$$E\left(\left|U_q^H p^\delta\right|\right) \approx E\left(\left|U_q^H \epsilon\right|\right) = \sigma, \text{for small } |\lambda_q|,$$ (3.8)

where E is the expected value, U_q^H is the qth row of the left singular matrix U^H from the SVD, p^δ is the filtered pressure, and ϵ is the noise. We average over the last few basis vectors,

$$\left\|U_q^H p^\delta\right\|/\sqrt{Q} \approx \sigma, M - Q + 1 \leq q \leq M,$$ (3.9)

for an approximate implementation of the expectation, assuming that the last evanescent waves associated with the last Q singular vectors/partial fields have all dropped below the noise. Once we have an estimate of the noise variance, we iteratively solve for α by minimizing

$$\left\|(UF^\alpha U^H - I)p^\delta\right\|/\sqrt{M} = \sigma.$$ (3.10)

For more details of this procedure and equations, see Williams[22, 24].

The filter obtained through modified Tikhonov regularization in conjunction with the Morozov discrepancy principle is the only filter needed for NAH because it tapers faster than the evanescent wave components.

3.4 Aperture extension

As discussed earlier, the measurement aperture does not fully encompass the sound source and therefore constitutes a "patch." Prior to applying the NAH processing, an extension of the measured patch is required. Otherwise, by use of the discrete Fourier transform, the discontinuity at the measurement edge will introduce wrap-around error, erroneous high-wave number content and consequently, spatial aliasing. There are multiple ways to deal with the edge effects: applying a spatial window,[17] zero-padding,[29] extrapolating, or applying analytic continuation.[22] Of these methods, the one that gave the best solution while introducing the least amount of error with the numerical test cases tried was analytic continuation.[22, 24, 36]

Analytic continuation is an attempt to approximate real data exterior to the measured region. The extent to which it is successful depends on the implementation and the available data. It is a process by which a zero-padded region outside the measurement edge, shown in Fig. 3.4 using numerically generated data, is iteratively filled with data that matches the phase and amplitude at the measurement edges and transitions smoothly outward to lower amplitude. Internal to the measured region, analytic continuation is also capable of smoothly interpolating the measured data. Continuation of an analytic field has a long history in electrostatics and was first applied to NAH by Williams[22, 24] in 2003.
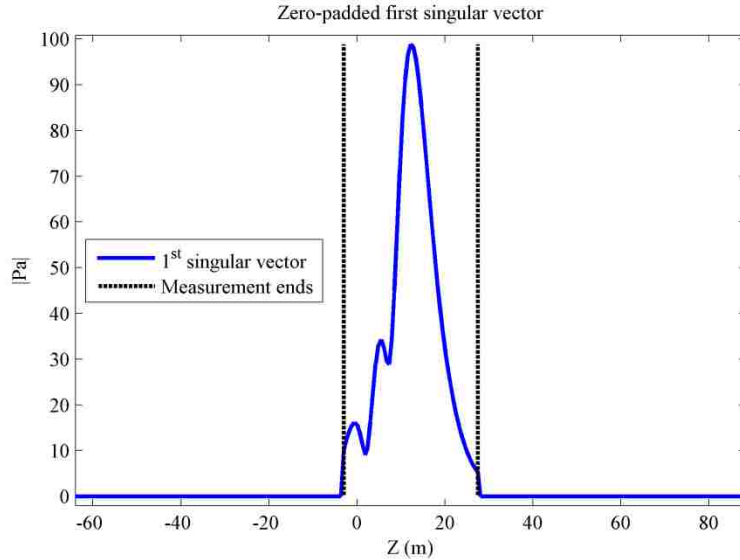
Figure 3.4 When beginning analytic continuation, each singular vector is initially zero-padded.

The process of analytic continuation used in this work for cylindrical and planar NAH is nearly the same. It differs only in what is being extended. For cylindrical NAH, the one-dimensional singular vectors, which originated from microphones located on the ground, are continued.

For planar NAH, the two-dimensional partial fields are continued after giving some more attention to the vertical dimension. It was found that there was not enough variation in the vertical direction to give physically meaningful continuations. Essentially, in the two-dimensional measurements, there is no tapering in amplitude, resulting in serious edge effects. In order to extend the vertical aperture while incorporating some knowledge of the sound field, the ground was assumed to be a perfect reflector, and each partial field was "mirrored" about the reflecting surface at $y = 0$ to give "image" data below the reflecting plane. The pressure values were interpolated between the measured and image regions. This gave maximum sound pressure levels at the ground, which is expected with a hard, reflecting surface. It was on these vertically mirrored partial fields that analytic continuation was performed for planar NAH.

Because analytic continuation theory is the same when applied to either one-dimensional data or two-dimensional data and because a one-dimensional continuation is easier to visualize, the representative examples will be for a numerically generated one-dimensional line of data. Following the representative examples, continuations from measured two-dimensional data will be shown.

As outlined by Williams,[22] the process of analytic continuation works as follows. First, a singular vector is zero-padded. Then its angular spectrum is calculated through a Fourier transform. Then the angular spectrum is initially zero-padded beyond a certain k-space cutoff, chosen during the modified Tikhonov regularization procedure described in Section 3.3. Figure 3.5 (top) shows the initial wave number spectrum, (middle) the k-space filter chosen during a Tikhonov regularization procedure, and (bottom) the product of the initial spectrum with the k-space filter. An inverse Fourier transform is performed. Then the original singular vector is replaced. The procedure is repeated with the exception of now using the smoothed data in the extended region instead of zero padding. The process continues until there is minimal change in the smoothed pressure, that is, until the L2 norm of the difference between iterations is less than the estimate of the noise variance times an ad hoc scaling factor[22] of 0.1. After the final iteration, the original singular vector is not replaced.

Figure 3.6 (top) demonstrates replacing the measured area with the original singular vector and (bottom) what the continued singular vector looks like after 10 iterations. The original data are not replaced on the last iteration. They are shown in Figure 3.6 (bottom) for reference only. In summary, the process of analytic continuation is achieved in the following manner:

1. Zero-pad the region outside the measurement edge.
2. Take a spatial DFT of the zero-padded singular vector.

3. Obtain a *k*-space filter using the modified Tikhonov regularization procedure of Section 3.3.

4. Multiply the spectrum and the *k*-space filter.

5. Inverse Fourier transform the filtered spectrum.

6. Check for convergence; when the L2 norm of the difference between iterations is less than the estimate of the noise variance times an ad hoc scaling factor.

7. Replace the measurement region with the original singular vector.

8. Repeat steps 2-7 until convergence has been met. Do not replace the measured data once convergence has been met.

For a given measurement to be extended, Williams recommends performing analytic continuation for every reconstruction distance. That is because one of the inputs to the modified Tikhonov filter is λ_i which depends on the reconstruction distance [see Eqs. (3.5) – (3.7)]. However, in the course of this work, the author found it unnecessary to perform analitytic continuation for every propagation distance. Rather, for each partial field or singular vector to be continued, analytic continuation need only be performed once, regardless of propagation distance. That is because for every propagation distance for a particular partial filed/singular vector, the filter shape is always the same because the α chosen by minimizing Eq. (3.10) is always much much greater than $|\lambda_i^2|$, corresponding to strong regularization. In other words, the evanescent wave information has decayed into the noise floor by the time the sound propagates to the microphone locations.

Figure 3.7 a) shows the aperture extension for the first three partial fields and b) their sum for the 105-Hz military power engine condition case. The black rectangle in b) represents

the location of the measured data. The aperture extension and their sum shown in Fig. 3.7 can be directly compared to the virtual coherence decomposition shown in Fig. 3.3. Notice how the filtering that takes place in the analytic continuation process helps smooth out the discontinuities between scans that are not removed in the virtual coherence process.
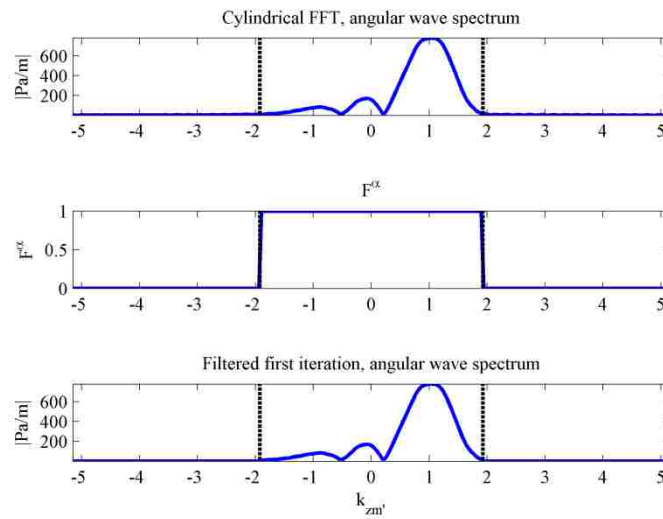
Figure 3.5 (Top) Initial angular spectrum of first, zero-padded singular vector. (Middle) k-space filter F_α. (Bottom) Filtered angular spectrum of first, zero-padded singular vector. Vertical bars denote $\pm k$.
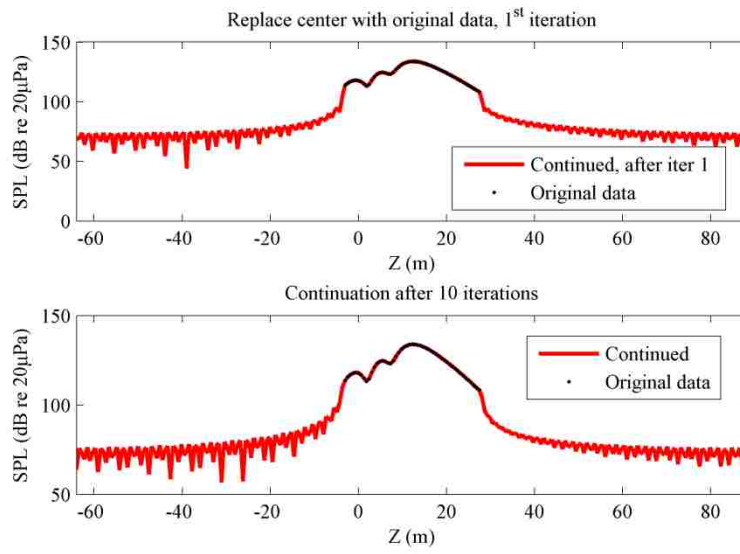
Figure 3.6 (Top) Replace the measured data into the measured region. (Bottom) Continued singular vector once convergence has been met.
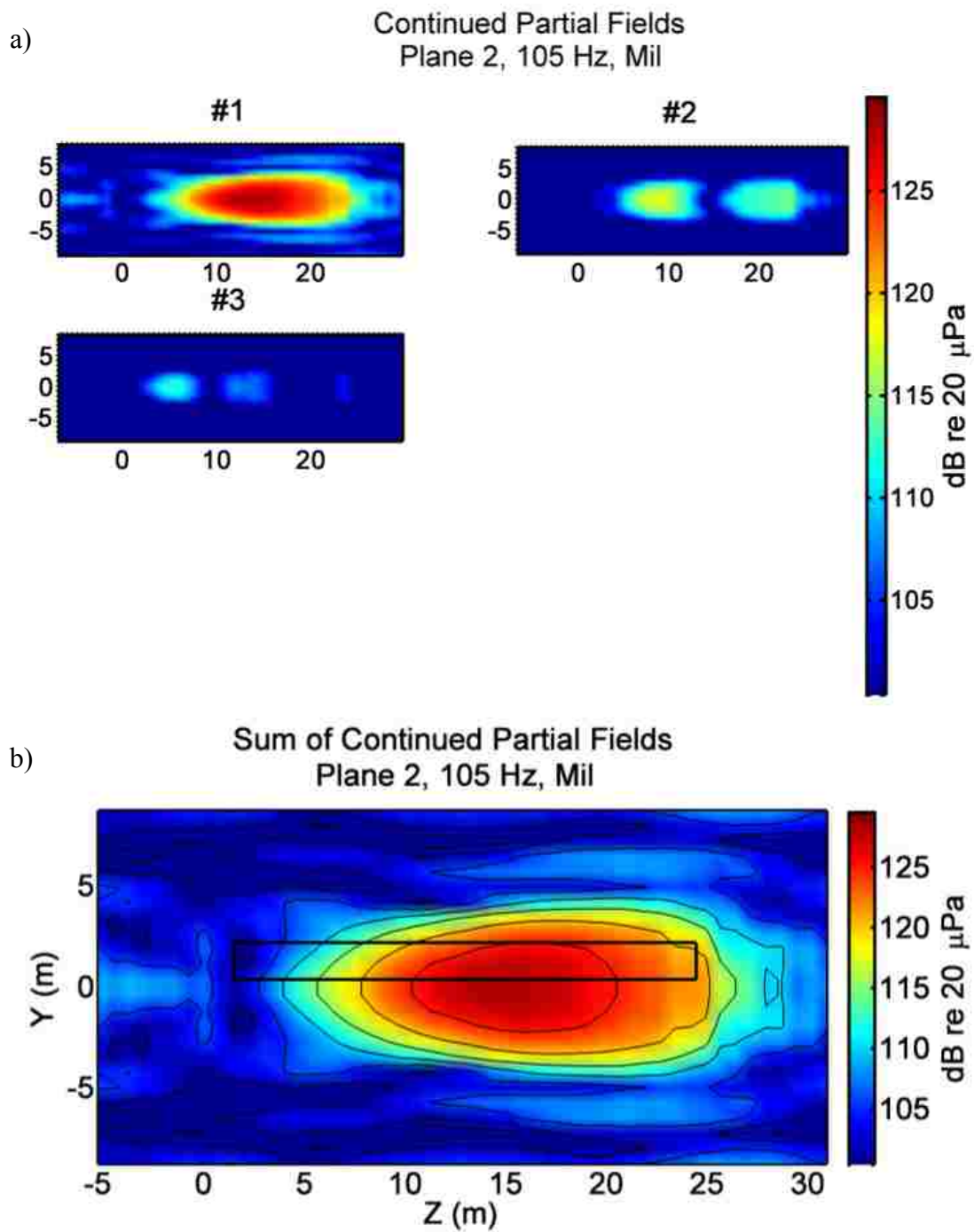
Figure 3.7 Analytic continuation results for 105 Hz, military engine condition: a) extended partial fields; b) sum of extended partial fields. The black box in b) represents the location of the measured data.

Fourier transforms are computationally efficient and well established, making their implementation relatively simple. Certain considerations must be addressed, however, especially when addressing a directional source. Use of a spatial DFT assumes the measurement is spatially periodic modulo N number of microphones. When a source is directional, the propagator will continue the main lobe along its original direction. When the main lobe reaches the edge of the continued domain, it will wrap around and enter through the other side of the continued domain. The effect of this wrap-around error can be minimized if the measurement is analytically continued out far enough. Numerical experiments discussed in Section 3.6.1.1 are used to decide how many points to continue are necessary.

3.5 Fourier NAH – cylindrical and planar

This section describes the implementation of Fourier NAH in cylindrical and Cartesian coordinates. Because of the number of similarities between cylindrical and planar NAH, their underlying theory will described simultaneously and their differences will be highlighted. The discussion will follow a pattern of first describing a piece of cylindrical NAH theory followed by the analogous piece of planar NAH theory. In general, a subscript h refers to the measurement or "hologram"; and a subscript s refers to the reconstruction or "source," be it the actual source location or not. For both cylindrical and planar NAH, we constrain ourselves to the source-free space external to any sources.

For both cylindrical and planar NAH, spatial resolution for a spatial DFT is analogous to frequency resolution for a temporal DFT. If a finer frequency resolution is desired for a temporal DFT, a smaller time increment Δt must be used. Similarly for a spatial DFT, if a finer spatial

resolution is desired, a smaller Δz must be used. When using the spatial DFT, the microphone

spacing Δz sets the upper frequency resolution bound. The DFT requires two microphones per

wavelength. With a given Δz, the upper frequency limit is

$$f_{max} = \frac{c}{2\Delta z},$$
(3.11)

where c is the speed of sound in the fluid. With our measurement setup shown in Fig. 2.2, the

two-foot spacing of the reference arrays gives an upper frequency limit of 281 Hz for cylindrical

NAH, and the six-inch spacing of the rig gives an upper frequency limit of 1125 Hz for planar

NAH.

In cylindrical NAH, the general expression to extrapolate the angular spectrum from the

cylindrical shell $r = r_h$ to the shell $r = r_s$ is

$$p(r_s, \phi, z) = \sum_{n=-\infty}^{\infty} e^{jn\phi} \int_{-\infty}^{\infty} P_n(r_h, k_z) e^{jk_z z} \frac{H_n^{(1)}(k_r r_s)}{H_n^{(1)}(k_r r_h)} dk_z,$$
(3.12)

where $P_n(r_h, k_z)$ is the two-dimensional Fourier transform in ϕ and z of the measured pressure at

$r = r_h$, and $H_n^{(1)}(k_r r)$ is a Hankel function of the first kind defined by

$$H_n^{(1)}(k_r r) = J_n(k_r r) + j Y_n(k_r r).$$
(3.13)

The subscript n comes from the solution to the wave equation in cylindrical coordinates and is

the separation of variables constant related to the $\Phi(\phi)$ dependence. Evanescent waves are

included in Eq. (3.12) since there are no restrictions on the values of k_r, allowing them to be real

or imaginary. If k_z is imaginary, the subsonic waves that would follow an exponential decay

actually follow a power law decay,[17] making k_r the only potentially evanescent term.

Since the two-dimensional Fourier transform of the left-hand side of Eq. (3.12) is

$P_n(r_s, k_z)$ then

$$P_n(r_s, k_z) = \frac{H_n^{(1)}(k_r r_s)}{H_n^{(1)}(k_r r_h)} P_n(r_h, k_z). \tag{3.14}$$

In planar NAH, the general expression to extrapolate the angular spectrum in the plane $x = x_h$ to the plane $x = x_s$ is

$$P(k_z, k_y, x_s) = P(k_z, k_y, x_h) e^{-jk_x(x_s - x_h)}, \tag{3.15}$$

where $P(k_z, k_y, x)$ is the two-dimensional Fourier transform of $p(z, y, x = constant)$. Evanescent waves are included in Eq. (3.15) because there are no restrictions on the values of k_x, allowing them to be real or imaginary.

Equations (3.12) and (3.15) provide the relationship between the wave spectra at different cylindrical and planar surfaces. We call

$$G_p(r_s, r_h, k_r) = \frac{H_n^{(1)}(k_r r_s)}{H_n^{(1)}(k_r r_h)} \tag{3.16}$$

and

$$G(x_s, x_h, k_x) = e^{-jk_x(x_s - x_h)} \tag{3.17}$$

the pressure propagators and use them to propagate the pressure from one shell, in the case of Eq. (3.16), or plane, in the case of Eq. (3.17), to another. The only restriction on r_s and x_s is that they be more positive than or equal to the radius or x-location of the source, respectively.

The radial and wavenumber k_r and is calculated by

$$k_r = \sqrt{k^2 - k_z^2}, \tag{3.18}$$

and the wavenumber in the x-direction is given by

$$k_x = \sqrt{k^2 - k_z^2 - k_y^2}, \tag{3.19}$$

where k is the acoustic wavenumber $k = \omega/c = 2\pi f/c$, and c the fluid sound speed.

Fourier NAH is capable of propagating not only pressure, but also acoustic particle velocity through use of Euler's equation in the frequency domain

$$j\omega\rho_0\vec{u} = \vec{\nabla}p. \tag{3.20}$$

A two-dimensional Fourier transform of Eq. (3.20) in either cylindrical or Cartesian coordinates leads to the velocity propagators that relate angular spectrum of the acoustic pressure at one surface to acoustic particle velocity at another.

In cylindrical coordinates we use the fact that $F_z\left(\frac{\partial}{\partial z}\right) = jk_z$ and $F_\phi\left(\frac{\partial}{\partial \phi}\right) = jn$ to get

$$\boldsymbol{U}_n(r_s, k_z) = \frac{1}{\rho_0 c}\left(-\frac{j}{k}\frac{\partial}{\partial r}\hat{e}_r + \frac{n}{kr}\hat{e}_\phi + \frac{k_z}{k}\hat{e}_z\right)P_n(r_s, k_z). \tag{3.21}$$

Thus, we can relate $\boldsymbol{U}_n$ to a pressure on a concentric surface at $r = r_s$. If we insert Eq. (3.14) into Eq. (3.21) we can obtain the radial $\boldsymbol{U}_{n,r}$ and axial $\boldsymbol{U}_{n,z}$ components of velocity:

$$\boldsymbol{U}_{n,r}(r_s, k_z) = \frac{-jk_r}{\rho_0 ck}\frac{H_n'(k_r r_s)}{H_n(k_r r_h)}P_n(r_h, k_z), \tag{3.22}$$

where $H_n'(k_r r)$ is the derivative with respect to the argument of the Hankel function

$$H_n'(k_r r_s) = \frac{\partial}{\partial k_r r_s}H_n^{(1)}(k_r r_s) = -H_{n+1}^{(1)}(k_r r_s) + \frac{nH_n^{(1)}(k_r r_s)}{k_r r_s}. \tag{3.23}$$

The axial component of velocity is

$$\boldsymbol{U}_{n,z}(r_s, k_z) = \frac{k_z}{\rho_0 ck}\frac{H_n^{(1)}(k_r r_s)}{H_n^{(1)}(k_r r_h)}P_n(r_h, k_z). \tag{3.24}$$

We can see from Eqs. (3.22) and (3.24) the radial and axial propagators, G_{ur} and G_{uz}, that move from pressure on a shell at $r = r_h$ to velocity on another shell at $r = r_s$ are:

$$G_{ur}(r_s, r_h, k_r) = \frac{-jk_r}{\rho_0 ck} \frac{H_n'(k_r r_s)}{H_n^{(1)}(k_r r_h)} \qquad (3.25)$$

and

$$G_{uz}(r_s, r_h, k_r) = \frac{k_z}{\rho_0 ck} \frac{H_n^{(1)}(k_r r_s)}{H_n^{(1)}(k_r r_h)}, \qquad (3.26)$$

where $H_n'(k_r r)$ is defined in Eq. (3.23).

In Cartesian coordinates, a two-dimensional Fourier transform of Euler's equation yields the velocity propagators that relate angular spectrum of the acoustic pressure at $x = x_h$ to the angular spectrum of the acoustic particle velocity at $x = x_s$. Namely, we have the velocity propagators

$$G_{ui}(x_s, x_h, k_i, k_x) = \frac{k_i}{\rho_0 ck} e^{-jk_x(x_s - x_h)}, \qquad (3.27)$$

where the subscript i represents the Cartesian coordinate axis z, y, or x. Similar to Eq. (3.15) for pressure propagation, the velocity propagators move from pressure at $x = x_h$ to velocity at $x = x_s$ through

$$U_i(x_s, x_h, k_i, k_x) = G_{ui}(x_s, x_h, k_i, k_x) P(k_z, k_y, x_h). \qquad (3.28)$$

In Eqs (3.12)-(3.19) and (3.22)-(3.28), the values of r_s and r_h and x_s and x_h play critical roles. When $r_s \geq r_h$ or $x_s \geq x_h$, the solution is a forward problem, as provided by Rayleigh's integral,[17, 37], and no regularization is necessary. When $r_s < r_h$ or $x_s < x_h$, the solution is an inverse problem, and regularization may be necessary.

For cylindrical NAH, if the wavelength in the axial direction is smaller than the acoustic wavelength, $\lambda_z < \lambda = 2\pi/k$, then $k_z > k$, and k_r as given by Eq. (3.18) is pure imaginary, and the associated waves are evanescent. The arguments of H_n in Eq. (3.16) then become pure

imaginary and become modified Bessel functions of the second kind $K_n(k_r r)$ which blow up at the origin. Because we have chosen $H_n^{(1)}(k_r r)$ for diverging waves, if k_r is imaginary, we choose the positive square root for evanescent wavenumbers.

Similarly for planar NAH, if the sum of squared wavelengths in the z and y directions is less than the acoustic wavelength squared, that is if $\lambda_z^2 + \lambda_y^2 < \lambda^2$ or if $k_z^2 + k_y^2 > k^2$, then k_x as defined by Eq. (3.19) is pure imaginary, and the associated waves are evanescent. Because we have defined our time waveform as having $e^{+j\omega t}$ time dependence, we choose the negative square root in Eq. (3.19) for subsonic wavenumbers, or wavenumbers $|k_x| > k$. Evanescent waves corresponding to imaginary k_x in Eq. (3.17) will blow up at the origin, making regularization necessary for back propagation.

3.6 Axisymmetric cylindrical NAH

There is an assumption in this work of axisymmetry for cylindrical NAH that merits careful attention and will be the focus of this section. Assuming axial symmetry about the jet center line, each analytically continued singular vector can be propagated using spatial DFT-based cylindrical NAH.[17] Cylindrical coordinates were chosen since they most closely resemble the jet geometry. An assumption of axial symmetry is not entirely accurate given the noncircular jet nozzle, however a low-frequency argument is necessary for axisymmetry, regarless of jet axis. Reflections from a source above a reflecting plane will certainly create destructive interference nulls above the surface, which nulls will not be accounted for in the axisymmetric assumption. However, since the reference microphones are on the ground, they can then

approximate an acoustical free field because there are no destructive interference nulls in the data measured on the ground.

The axisymmetric assumption also has an impact on the propagators found in Eqs. (3.16), (3.25), and (3.26). The subscript n comes from the solution to the wave equation in cylindrical coordinates and is the separation of variables constant related to the $\Phi(\phi)$ dependence. If there is no Φ dependence, then $n = 0$ and the solution has axial symmetry. The Hankel functions in Eqs (3.16) and (3.26) simply become zeroeth order. From Eq. (3.23) and $n = 0$, Eq. (3.25) reduces to

$$G_{ur}(r_s, r_h, k_r) = \frac{jk_r}{\rho_0 ck} \frac{H_1^{(1)}(k_r r_s)}{H_0^{(1)}(k_r r_h)} \tag{3.29}$$

In essence, using the microphones along the ground for cylindrical NAH allows for reconstructions along the plane consisting of the jet centerline and reference microphone array. Processing along this plane can occur independent of an axisymmetric assumption, although $n = 0$ implies axisymmetry. Where an axisymmetric assumption is needed is when cylindrical NAH reconstructions are compared to benchmark measurements at other azimuthal angles, such as along the shear layer or the 23-m arc.

3.6.1 Considerations on the axisymmetric assumption

3.6.1.1 Numerical experiments on the axisymmetric assumption

In order to examine the axisymmetric assumption, numerical simulations were performed that were intended to mimic the directionality of the source and the geometry of the measurement setup.

The source of interest is the jet flow coming from the aircraft nozzle. As such, a distributed line array of monopole sources was placed along the jet centerline. This may not be the source distribution for the actual jet, but it simplifies the numerical simulations. Laufer, Schlinker, and Kaplan[38] discuss how the turbulent jet, viewed as a source of noise, is the summation of a number of independent acoustic sources distributed along the jet axis. In array applications, it is possible to steer sound in a preferred direction, or a known steer angle, ϕ_0, without actually rotating the array physically.[39, 40] This is accomplished by increasingly introducing phase, or time delay, along the line array. This time delay, τ_0, is given by

$$\tau_0 = \frac{d\cos(\phi_0)}{c},$$

(3.30)

where d is the spacing between each consecutive monopole source and c is the fluid sound speed. The steer angle ϕ_0 is measured from the upstream, end-on direction.

The pressure at a field point r from a monopole source is given by

$$p(r,t) = \frac{j\rho_0 ckQ}{4\pi} \frac{e^{-j(kr-\omega t)}}{r},$$

(3.31)

with ρ_0 the fluid density, c the fluid sound speed, $k = \omega/c = 2\pi f/c$ the acoustic wavenumber, and Q the source strength. By applying the phase shading to each monopole and adding up their contributions, and assuming each monopole has the same source strength, the total pressure from each monopole at a field point r can be found via

$$p(r,\phi_0,t) = \frac{j\rho_0 ckQ}{4\pi} \sum_{n=1}^{N} \frac{e^{-j(kr_n - \omega(t + n\tau_0))}}{r_n},$$

(3.32)

where r_n is the distance from the nth monopole to the field point r. The time delay is independent of drive frequency, thus the steer angle does not change as frequency is changed.

Noise of a desired signal to noise ratio, SNR, can be added to the simulation by use of

$$\text{noise} = \sqrt{\frac{|p|}{\left(\frac{\text{SNR}}{\sqrt{N_r}}\right)}} \sqrt{\frac{1}{2}} (\text{randn} + j\text{randn}),$$ (3.33)

where

$$\text{SNR} = 10^{\left(\frac{\text{SNRdB}}{20}\right)}$$ (3.34)

and SNRdB is the desired signal to noise ratio in decibels, randn is an array of normally distributed pseudorandom numbers, and N_r is the number of simulated pressures at a given x.

Now a brief word about the addition of two sound sources. If two identical sources the same distance away from a field point add coherently, there is a sound pressure level increase of 6 dB. If two identical sources the same distance from a field point add incoherently, there is a sound pressure level increase of 3 dB. This range stems from the difference between coherent and incoherent addition of two sound sources. Coherence refers to the degree to which there is a constant phase difference between signals. Coherent signals have a constant phase difference. Incoherent signals have no phase relation. The sum of two incoherent sound sources, L_A and L_B,

$$L_A + L_B = 10 \log_{10} \left(10^{\frac{L_A}{10}} + 10^{\frac{L_B}{10}} \right).$$ (3.35)

can have a maximum value of 3 dB. The sum of two coherent sound sources, L_C and L_D,

$$L_C + L_D = 20 \log_{10} \left(10^{\frac{L_C}{20}} + 10^{\frac{L_D}{20}} \right).$$ (3.36)

can have a maximum value of 6 dB

The coherence between direct sound and that from a reflection such as the ground depends on the reflecting surface. If the reflecting surface is infinitely rigid and perfectly smooth, then the reflected wave is fully coherent with the incident wave. If it is rough and porous, then

the reflection is more likely incoherent. In the problem at hand, the reflection occurs on hard, flat asphalt which acts as a near perfect reflector. We can assume the reflected sound is more coherent than incoherent with the direct sound, but still we have a pressure magnitude range of 3 to 6 dB.

Another issue for coherent sources is destructive interference. Should the path length differ by half a wavelength, destructive interference will occur. Destructive interference of this sort can be seen in the actual measured data in the results section of this work in Fig. 5.16. The interference nulls, however, are not infinitely deep, so the measured data from the jet and its image are best termed partially coherent.

Now back to the numerical simulation. Using Eqs. (3.30) and (3.32) – (3.34) we are in a position of generating simulated pressures from a directional line array. We will set up the line array off the ground and an image array below the ground simulate a coherent ground reflection.

The line array is at a height of $y = 1.905$ m (75 in) and consists arbitrarily of $N = 11$ monopoles in a line starting at $z = 2$ m and with $d = 0.75$-m spacing. The chosen steer angle is $\phi_0 = 120°$. To approximate the ground reflecting plane, an image line array of data is placed at $y = -1.905$ m. Noise with a SNR of 60 dB is added to both line arrays. Numerical benchmark pressures at 105 Hz are shown in Fig. 3.8. Also shown as black dots are the locations of the numerical reference microphones.
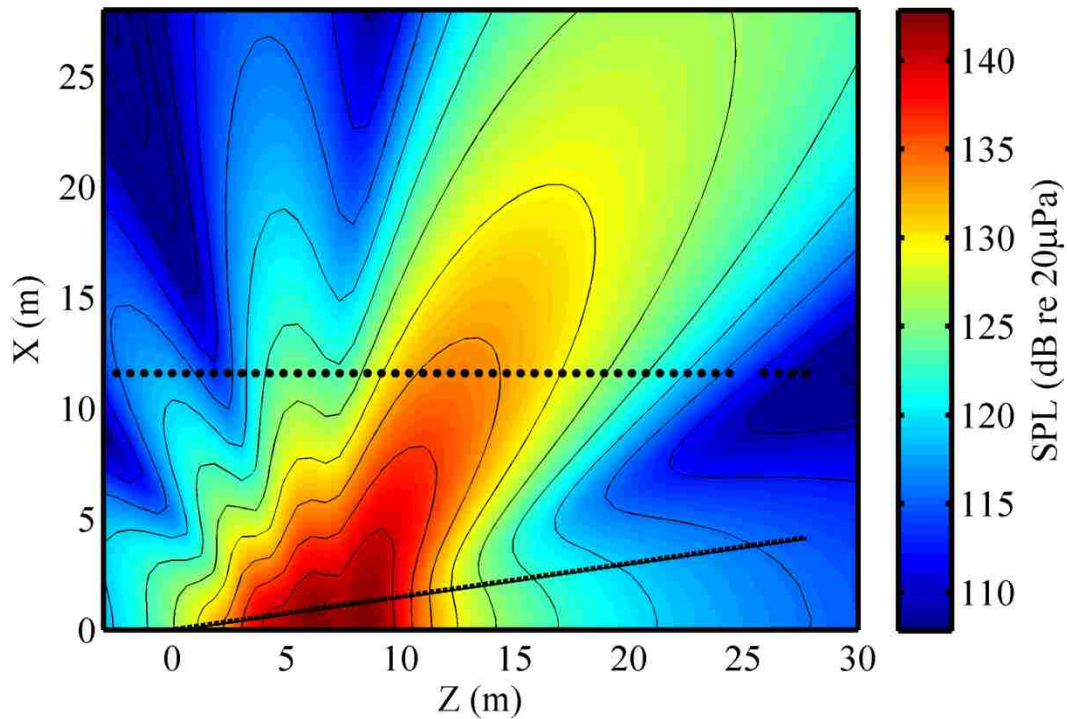
Figure 3.8 Numerical simulation benchmark pressures at 105 Hz. Numerically measured pressures at each location come from both a line array above the ground and an image line array below the ground. The dots show locations of the numerical reference microphones and the angled line the estimated shear layer.

By keeping the values at the numerical reference microphones, which are located on the ground ($y = 0$ m), we are able to decide how many points to analytically continue, as discussed in Section 3.4, as well as test the functionality of the cylindrical processing described in Section 3.5. Also we can test the difference between including the image line array or not, as well as regularization schemes and windowing. In essence, we have a simulated source that can produce measurement and benchmark reconstruction data and allows us to test the functionality and validity of NAH processing.

First, we consider how many points to analytically continue. The spatial DFT assumes the measurement is spatially periodic modulo N number of microphones. If the source is

directional, then wrap-around error yields erroneous results. Figure 3.9 shows the wrap around

error from continuing only 5 points while attempting to propagate to 100 m from the jet.
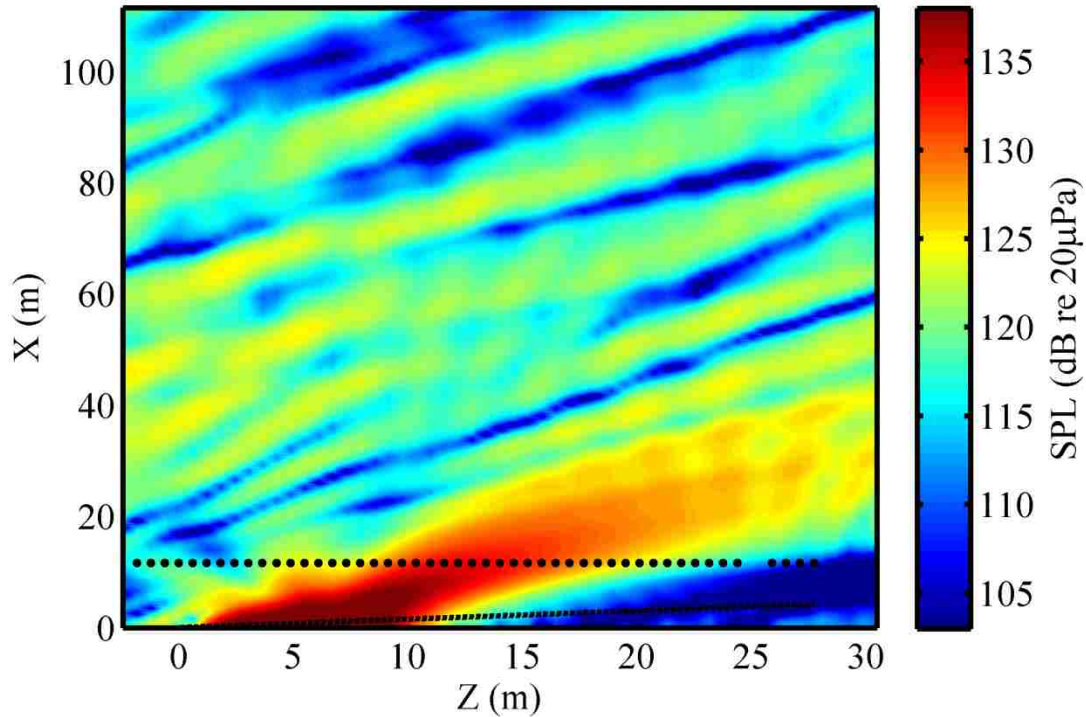
Figure 3.9 The detrimental effects of wrap-around error, having only applied analytic continuation out 5 points on each side. This pressure reconstruction shows wrap-around error as a high-amplitude band leaves one side and enters from the other. This is caused by the periodic assumption of the Fourier transform.

If analytic continuation is used to extend the measurement more points, then the periodic

replicas caused by the DFT are spatially farther away from the reconstruction region of interest

and the directional lobe has less of an impact. Figure 3.10 shows the reduced impact of wrap-

around error when continuing 100 points on both sides.
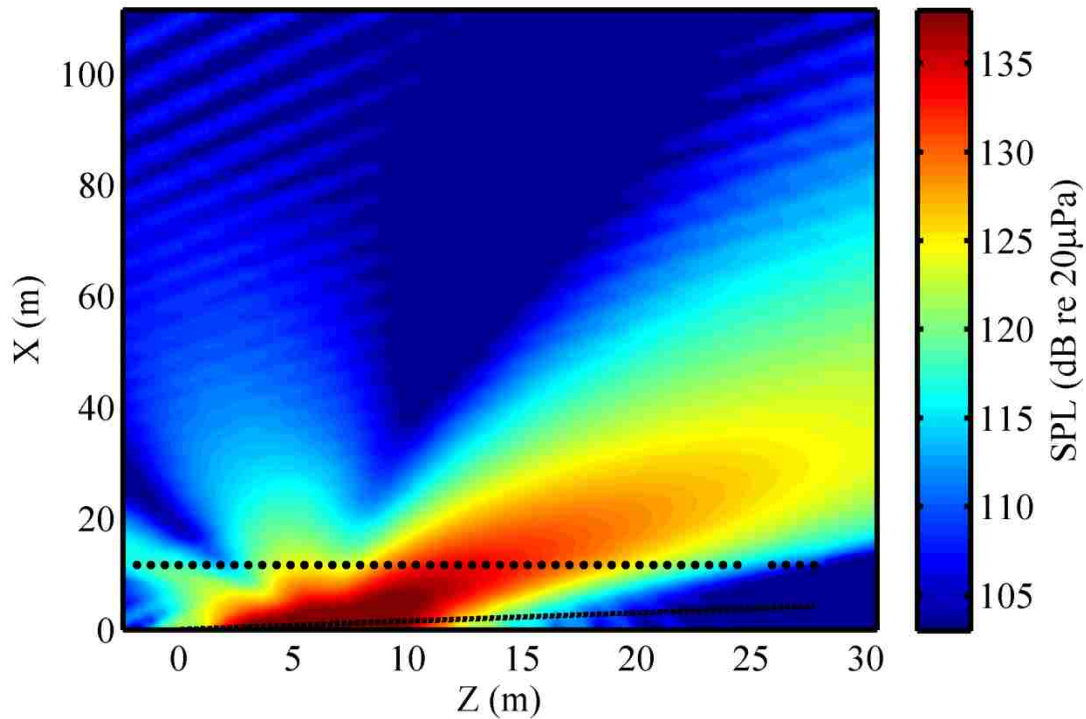
Figure 3.10 Pressure reconstruction using numerical data that was analytically continued 100 points on each side. The effect of wrap-around error is greatly reduced.

Now to test the cylindrical NAH processing: reconstructed pressure results for cylindrical NAH on the above numerical simulation are shown in Fig. 3.11. The results were generated by keeping the values at the numerical microphone locations, performing STSF, analytic continuation, and recombining the singular vectors on an intensity basis. Reconstruction agrees well with Fig. 3.8 for the lobes which passed through the numerical microphones, shown in the figure as black dots. That other lobes are not reconstructed well suggests not enough of their information was recorded at the numerical microphones. For data whose information is recorded in the microphones, however, cylindrical NAH processing performs well.
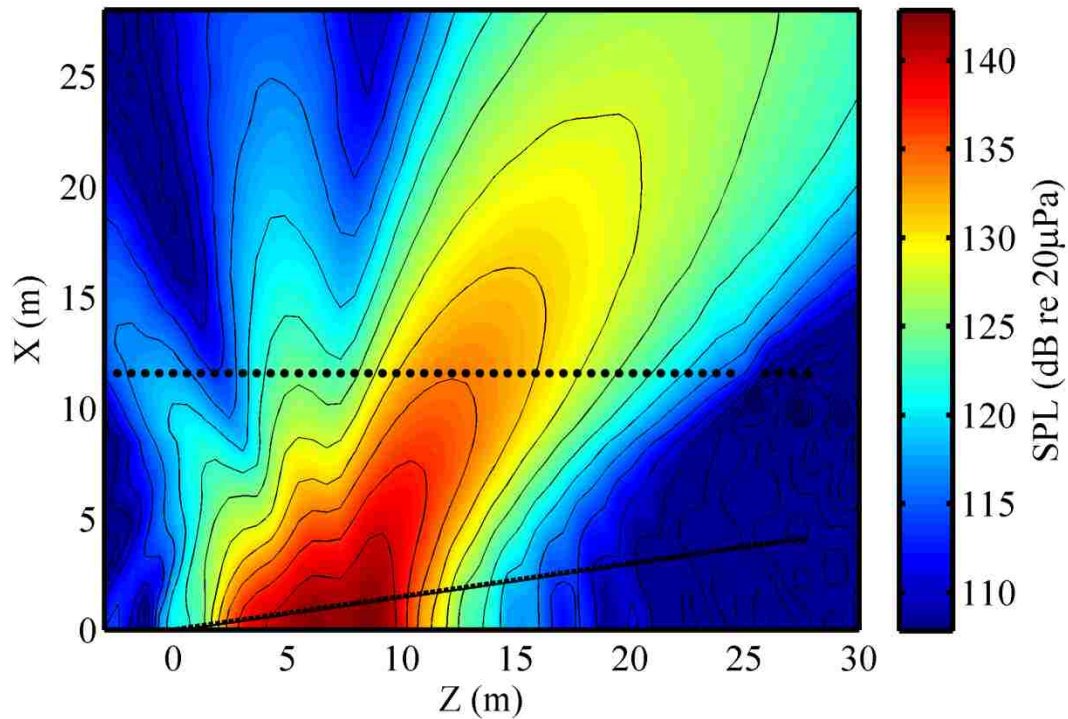
Figure 3.11 Reconstructed pressures at 105 Hz, obtained from a numerically simulated line array and image line array recorded at the black dots. The angled line is the estimated shear layer location. Lobes which pass through the numerical microphones are reconstructed well.

We now remove the image line array. Figure 3.12 shows the reconstructions along the shear layer from data generated from a line array both with and without an image source. As expected, the reconstructed pressures are 6 dB lower because the two sources no longer add coherently at the numerical microphones.
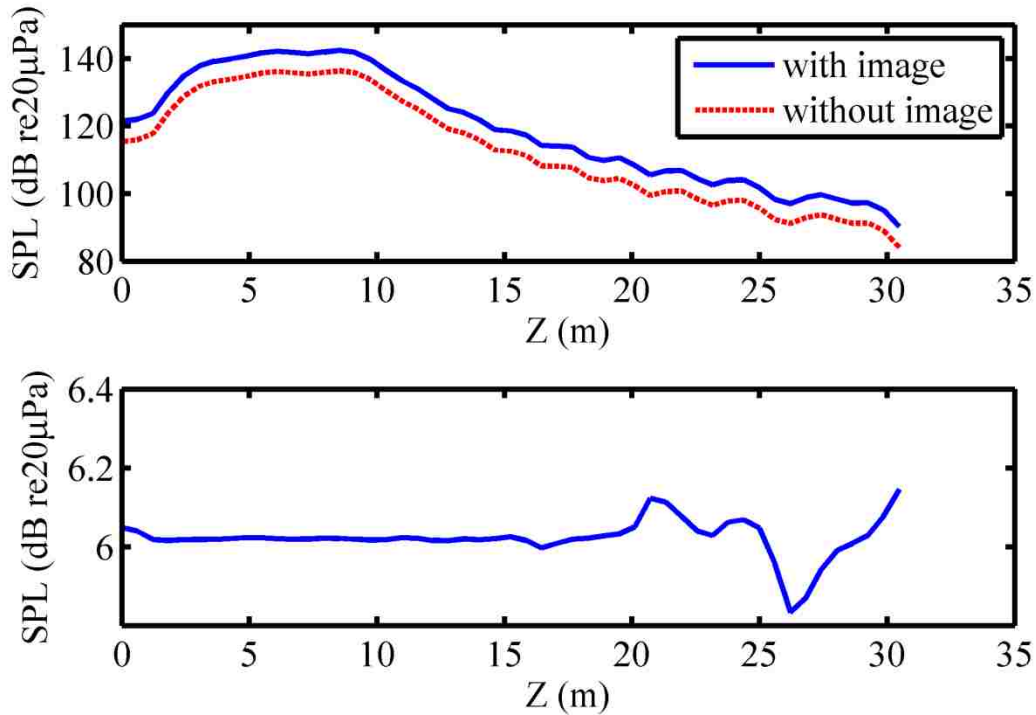
Figure 3.12 (Top) Reconstructions along the shear layer from numerically generated data with and without the image line array. (Bottom) The difference between the two curves is 6 dB.

3.6.1.2 Method of images

The method of images is one which if the measurements are taken at points along the bisecting plane—the plane for which every point on the plane is the same distance away from both the original and image source—6 dB can be subtracted from the measured data and it is as if the measurements were taken in free space, devoid of the reflecting plane and image source. Furthermore, if the sound radiation is axisymmetric, reconstructing at a certain radius and azimuthal angle is equivalent to reconstructing at the same radius and a different azimuthal angle. Alternatively, propagation could occur along the ground and the minus 6 dB correction would not be used. Figure 3.13 shows two locations, the black circles, which under the

axisymmetric assumption would yield the same result. The white circle would yield the same result boosted by 6 dB. As discussed in the introduction to Section 3.6, an axisymmetric assumption is not necessary for cylindrical NAH processing, but it is if comparing to benchmark measurements not along the azimuthal angle of the reference microphones. In this work, for cylindrical coordinates, the 6 dB correction was not used and propagation occurred along the ground by means of assuming axisymmetry and reconstructing to radii along the original azimuthal angle that correspond to distances of benchmark measurements, such as along the arc or for numerical simulations along the estimated shear layer.
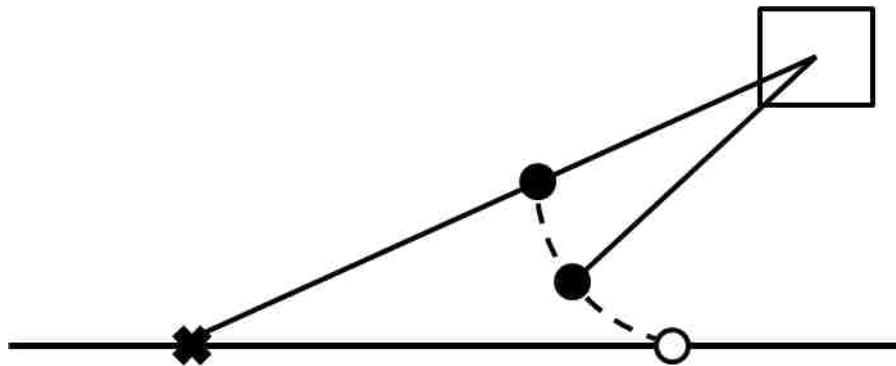
Figure 3.13 Diagram showing two locations, the black circles, that under the axisymmetric assumption will yield the same result. The white circle represents a point along the bisecting plane that gives a result 6 dB higher. The black x represents the microphone location. The center of the square is the jet centerline and the axis of the cylindrical coordinate system. The black horizontal line is the bisecting plane below which would be an image source.

3.6.1.3 Mutual radiation impedance

Another consideration for assuming axisymmetry is the issue of mutual radiation impedance between both the source above the reflecting plane and the source below the reflecting plane, or equivalently, between the source and its own reflection. Mutual radiation impedance deals with the effect that two sources – or any number of sources – in close proximity

have on each other. The acoustic radiation of one source will impact the radiation of the other source and vice versa. This section discusses how the source and its image are not coupled, assuming a linear-array source distribution.

Now in the problem at hand, the sound generated by the turbulent flow of a high-speed jet, the exact source mechanism is not well understood. Thus calculating the mutual radiation impedance between the jet and its reflection becomes very difficult. If, however, we approximate the turbulent source as a simple directional line array of monopole sources as in Section 3.6.1.1, we can use well-established theory of mutual radiation impedance between monopole sources[41, 42] to understand approximately how the jet and its reflection affect each other. If we find that the mutual impedance is weak, then addressing the image source in the method of images by either subtracting 6 dB or keeping the 6 dB and propagating along the ground has merit because the source and its image are not coupled.

Mutual impedance is defined as the ratio of the sound pressure at source 1 due to source 2 and volume velocity of source 2. From Eq. (3.31), the complex pressure amplitude at source 1 due to source 2, p_{12}, is

$$p_{12}(d_{12}) = j \frac{k^2 \rho_0 c}{4\pi} Q_2 \frac{e^{-jkd_{12}}}{kd_{12}}, \tag{3.37}$$

where

$$d_{12} = |\overrightarrow{r_2} - \overrightarrow{r_1}|. \tag{3.38}$$

The mutual impedance is then

$$Z_{12} = \frac{p_{12}}{Q_2} = \frac{k^2 \rho_0 c}{4\pi} \left[\frac{\sin(kd_{12})}{kd_{12}} + j \frac{\cos(kd_{12})}{kd_{12}} \right] \tag{3.39}$$

At source 1, $p_{12} = Q_2 Z_{12}$, and at source 2, $p_{21} = Q_1 Z_{21}$. Because $d_{12} = d_{21}$,

$$Z_{12} = Z_{21},$$ (3.40)

which represents reciprocity.

Now we can consider the effect of multiple sources on the pressure at each source and represent it in matrix form

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1N} \\ Z_{21} & Z_{22} & & Z_{2N} \\ & \vdots & \ddots & \vdots \\ Z_{N1} & Z_{N2} & \cdots & Z_{NN} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_N \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{N} Z_{1n}Q_n \\ \vdots \\ \sum_{n=1}^{N} Z_{Nn}Q_n \end{bmatrix},$$ (3.41)

where N is the total number of monopoles in both line arrays. The impedance matrix $\bar{Z}$ in Eq. (3.41) is an $N \times N$ square symmetrical matrix, and represents the mutual impedance between each source and itself and every other source. The diagonal terms of $\bar{Z}$ represent the self-impedances, whereas the off-diagonal terms are the mutual impedances.

Each element in $\bar{Z}$ is complex-valued and given by Eq. (3.39) but with indexing as in Eq. (3.41). The real part of each element pertains to the radiated field. The imaginary part of each element pertains to the evanescent field. We are going to examine the mutual radiation impedance by looking at the radiated sound power, Π, given by[42]

$$\Pi = \frac{1}{2}\bar{Q}^H Re[\bar{Z}]\bar{Q}.$$ (3.42)

By setting the source strength of every source to one, we can consider just the real part of the impedance matrix. The impedance matrix then becomes

$$Re[\bar{Z}] = \frac{k^2 \rho_0 c}{4\pi} \begin{bmatrix} \frac{\sin(kd_{11})}{kd_{11}} & \frac{\sin(kd_{12})}{kd_{12}} & \cdots & \frac{\sin(kd_{1N})}{kd_{1N}} \\ \frac{\sin(kd_{21})}{kd_{21}} & \frac{\sin(kd_{22})}{kd_{22}} & & \frac{\sin(kd_{2N})}{kd_{2N}} \\ & \vdots & \ddots & \vdots \\ \frac{\sin(kd_{N1})}{kd_{N1}} & \frac{\sin(kd_{N2})}{kd_{N2}} & \cdots & \frac{\sin(kd_{NN})}{kd_{NN}} \end{bmatrix}.$$ (3.43)

The diagonal elements in the matrix of Eq. (3.43) are all 1 by definition of the sinc function. Estimating the effect of the mutual radiation impedance then becomes a matter of comparing the off-diagonal elements of $Re[\bar{Z}]$ to the diagonal elements; that is, how does the mutual impedance compare to the self-impedance?

By calculating $Re[\bar{Z}]$ for the numerical simulation outlined in Section 3.6.1.1, we learn some important information in regards to our geometry. Figure 3.14 graphically shows $Re[\bar{Z}]$, neglecting the scaling coefficient. The top left and bottom right quadrant are the self and mutual impedances of the source array on itself and image array on itself, respectively. The top right quadrant shows the mutual impedance between the source array and the image array. Because $Re[\bar{Z}]$ is a symmetric matrix, the bottom left quadrant shows transposed information of the top right quadrant. None of the mutual impedance values from one line array to the next (the top right quadrant of Fig. 3.14) are greater than 0.12.
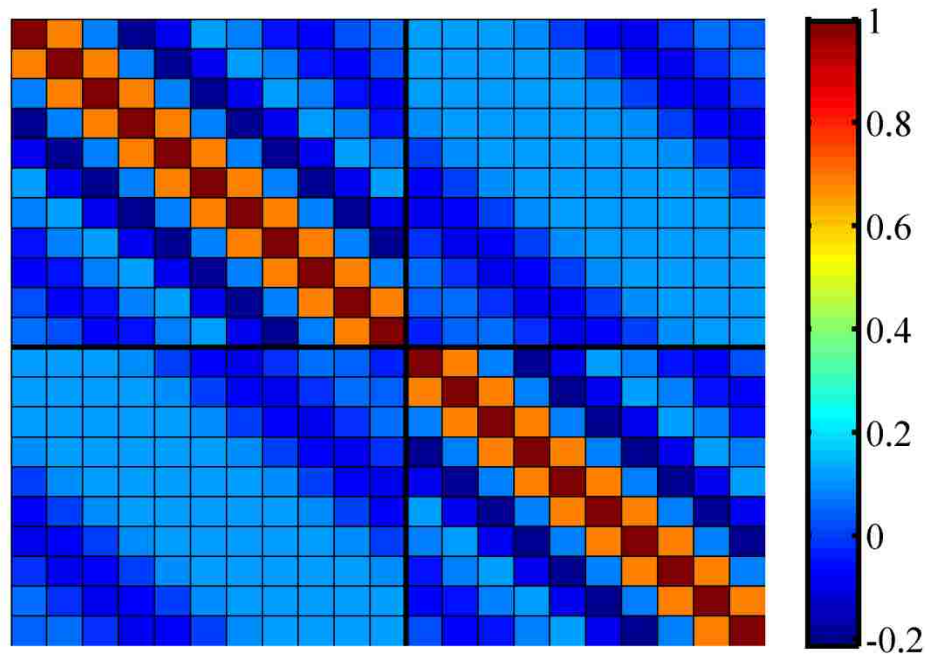
Figure 3.14 The real part of the impedance matrix given by Eq. (3.43).

Now we examine some of the mutual impedance values more explicitly. The smallest distance between the line array and its image is $d = 3.81$ m. Our frequencies of interest, $f = 105$ and 210 Hz, then yield kd values of $kd = 7.33$ and 14.66, which when input into $Re[\bar{Z}]$ yield off-diagonal elements of $\sin(7.33)/7.33 = 0.1180$ and $\sin(14.66)/14.66 = 0.0592$, both of which are considerably less than the diagonal elements of 1. Furthermore, for our given distance, as we increase in frequency, kd will continue to increase and the sinc function will yield smaller off-diagonal elements. This means the mutual radiation impedance of the image line array does not have a significant impact on the source line array for all frequencies of interest. The two line arrays can be considered as two separate and uncoupled sources, further strengthening the use of the method of images to assume axisymmetry about the jet.

3.7 Acoustic intensity

An additional array-based measurement performed in this work is that of vector acoustic intensity using a tetrahedral intensity probe. The data are analyzed briefly in this work mostly to see if it is viable for further analysis. However, qualitatively comparisons to results provided by cylindrical NAH are also made. This section and its subsections outline theory behind the tetrahedral intensity probe.

Acoustic pressure and acoustic particle velocity are useful quantities in the characterization of a sound source because not only can we know the pressure magnitude but they help calculate the time averaged acoustic intensity as defined by:

$$\langle \boldsymbol{I} \rangle_t = \frac{1}{2} \mathrm{Re}\{p\boldsymbol{u}^*\}. \tag{3.44}$$

Determination of sound intensity flow at a point in a plane is useful in determining a sound propagation direction. If a vector field is obtained over multiple locations, a sound source location or propagation direction can be inferred through ray tracing. While phase and amplitude of a source are not obtained in this method, it is useful in determining source location, frequency-dependent propagation direction, and general source magnitude trends. Measuring intensity of a jet sound source has not seen broad application and only recently has been applied to near-field source characterization analysis.[28, 43, 44]

As discussed in Section 3.5, acoustic pressure and particle velocity are related by Euler's equation

$$j\rho_0 ck\boldsymbol{u} = -\boldsymbol{\nabla}p \tag{3.45}$$

in the frequency domain. A knowledge of the pressure gradient is required. The standard method for approximating the pressure gradient is known as the finite-difference method and will be discussed in the following subsection.

3.7.1 Finite-difference method

The finite-difference method is a standard formulation for calculating acoustic particle velocity.[45] It offers significant advantages over directly measuring the particle velocity. Direct measurement of particle velocity is difficult, expensive, and in many cases the measurement devices are fragile and unsuited for the harsh environment as is near the military jet. On the other hand, the finite-difference method is relatively simple, inexpensive, and the measurement device is durable and thus well suited for the near-flow environment. The finite-difference method uses two pressure transducers – typically nearly identical, phase-matched condenser microphones – that are spaced a known, small distance from each other.

In one dimension, the gradient of the pressure may be easily calculated by taking the difference between two closely spaced microphones divided by the separation distance Δz to obtain

$$\frac{\partial p}{\partial z} \approx \frac{p_1 - p_2}{\Delta z}.$$

(3.46)

This gives the pressure gradient parallel to a line connecting the centers of the two microphones. That the microphones be phase-matched is crucial if the separation distance is small. The effect of a slight phase mismatch is reduced as the separation distance increases. The maximum separation distance is limited by the finite-difference approximation of the spatial derivative which requires the separation distance be much less than a wavelength, λ. That is, $\Delta z \ll \lambda$.

Fahy[45] suggests the spacing should be less than 0.13λ to avoid errors greater than 5%. These requirements limit the usable frequency bandwidth for the finite-difference method.

3.7.1.1 Tetrahedral probe

The finite-difference method may be extended to three dimensions by use of a pair of microphones along each cardinal axis using a total of six sensors. Fewer microphones are possible and various configurations can be used.[43, 46] As few as four microphones may be used if they are placed at the four corners of a tetrahedron. Since three points define a plane, adding a fourth point, as in a tetrahedral arrangement, ensures some component out of plane. Figure 3.15 shows a diagram of the tetrahedral intensity probe. During the actual measurements, the tetrahedral probe, seen in Fig. 2.1, was placed in the same orientation as Fig. 3.15 but with sensors 2 and 4 parallel to the guide rail.
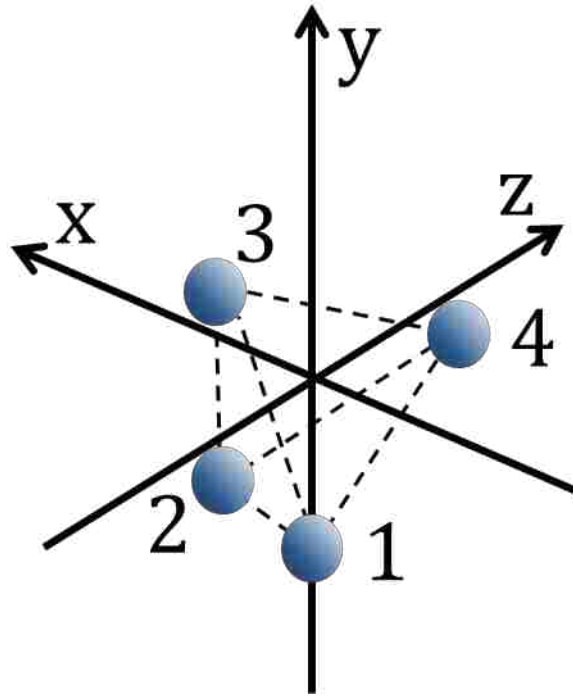
Figure 3.15 A tetrahedral intensity probe with microphones placed at the vertices of a regular tetrahedron with a circumsphere of radius a. Sensor 1 lies on the y-axis. Sensors 2 and 4 are parallel to the z-axis.

Two different and complete derivations with equivalent results for how to use the tetrahedral probe and the finite-difference method to estimate the pressure gradient are given by Thomas[47] and Pascal and Li.[46]

To summarize Thomas: a finite-difference, minimum least-squares estimate of the pressure gradient is obtained by computing a left pseudo-inverse[48] of a matrix representing the pressure difference between any sensor pair. This pressure gradient estimate may be used with Euler's equation, Eq. (3.45), to obtain the frequency-domain particle velocity

$$\boldsymbol{u} = \frac{j}{\rho_0 \omega} \boldsymbol{\nabla} p. \tag{3.47}$$

The frequency-domain pressure is the Fourier transform of the pressure at the acoustical center of the probe, which is approximated as the average of the pressure measured at the four microphone locations

$$p \approx \frac{p_1 + p_2 + p_3 + p_4}{4}.$$ (3.48)

Finally, defining a single-sided cross-spectrum as

$$G_{ij} = 2\left[p_i^* p_j\right]_{\omega=0\ldots\infty},$$ (3.49)

and substituting the estimated values for $\boldsymbol{u}$ and p, and multiplying by 2 to account for a single sided definition of intensity,, the time averaged intensity is

$$\langle \boldsymbol{I} \rangle_t = \frac{2}{32 \rho_0 \omega a} \begin{bmatrix} \sqrt{2} \, \mathrm{Im}\{-G_{12} + 2G_{13} - G_{14} + 3G_{23} - 3G_{34}\} \\ \sqrt{6} \, \mathrm{Im}\{G_{12} - G_{14} - G_{23} - 2G_{24} - G_{34}\} \\ 4 \, \mathrm{Im}\{G_{12} + G_{13} + G_{14}\} \end{bmatrix}.$$ (3.50)

The expressions given in Eqs. (3.48)-(3.50) allow us to compute benchmark acoustic intensity estimates using data collected by the tetrahedral intensity probe.

4 Numerical simulations

With the measurement setup outlined in Chapter 2 and the theory behind the NAH process in both cylindrical and Cartesian coordinates outlined in Chapter 3, the success of the process using the numerical source described in Section 3.6.1.1 may be investigated. The advantage of the numerical simulation is the ability to generate benchmark reconstructions at any location. As discussed in Section 3.6.1.1, a directional line array of monopole sources was chosen for its simplicity, axisymmetry, definable steer angle, and because the turbulent jet, viewed as a source of noise, could be represented as a summation of a number of independent acoustic sources distributed along the jet axis.[38]

The frequencies of interest in this study will be limited to 105 Hz and 210 Hz. As shown in Eq. (3.11), the 2-ft spacing of the ground microphone array limits cylindrical NAH analysis to less than $f_{max} = c/2\Delta z = 343/2(2 \cdot .3048) = 281$ Hz. Also, from Figure 2.3, the maximum sound pressure level occurs somewhere between 100 and 200 Hz. Consequently, we focus on 105 Hz and 210 Hz.

The entire holography processing illustrated in the flowcharts of Fig. 1.1 and Fig. 1.2 is performed on numerical data. Comparisons are made between the two types of NAH in this work and numerically generated benchmark reconstructions. These numerical comparisons provide guidelines when using the process on actual measured data; guidelines that answer three key questions: how does source directionality affect the end results? Is the source location accurate? Is the source level reconstruction accurate? It is shown that if a directional lobe was not well recorded in the measurement, it is not accurately reconstructed. Also shown is that cylindrical NAH accurately reconstructs the maximum source location and the source level amplitude.

Planar NAH grossly understimates source level amplitude and only gives an aperture wihin which the maximum location resides..

4.1 Numerical results for cylindrical NAH

We begin by establishing numerical benchmark measurements according to the numerical source outlined in Section 3.6.1.1. Figure 4.1 shows an aerial view of numerical benchmarks at 105 Hz and 210 Hz. Clearly seen is the directionality of the source. Also readily shown is the lobing caused by steering the line array.
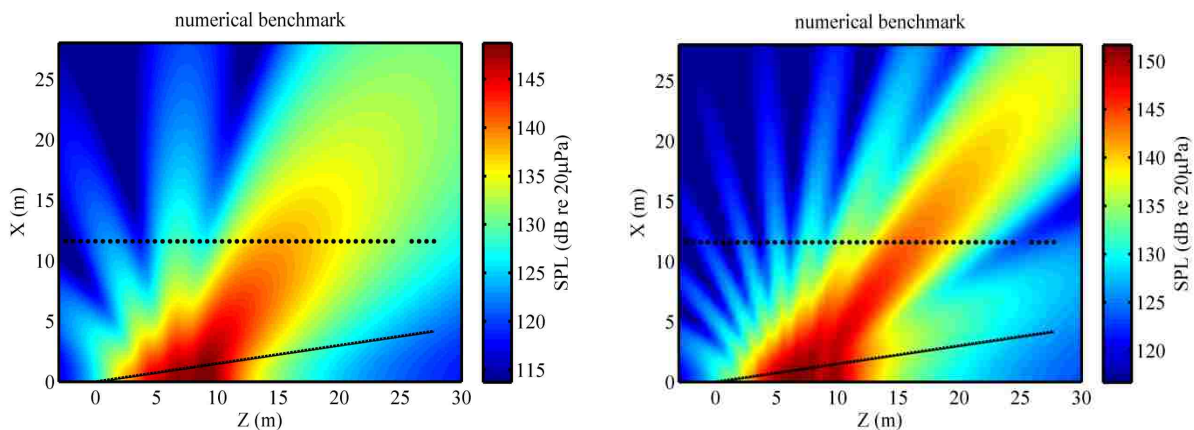
Figure 4.1 Numerical benchmark measurement at (Left) 105 Hz and (Right) 210 Hz. Dots represent locations of numerical microphones. Slanted line is the estimated shear layer.

These lobes provide important insight into the limitations of the reconstructions. Of particular note are the lobes that do not pass across the numerical microphone locations that are shown as black dots in Fig. 4.1. Reconstructions using cylindrical NAH on the numerical measurement, shown in Fig. 4.2, reveal the lobes not recorded in the measurement are not reconstructed. In particular, see the reconstruction at 210 Hz in the region X < 10 m and Z > 15 m. Similarly for 210 Hz, the upstream reconstruction from Z < 0 m and X < 5 m did not

reconstruct an upstream lobe because that lobe did not pass through the line of numerical microphones.
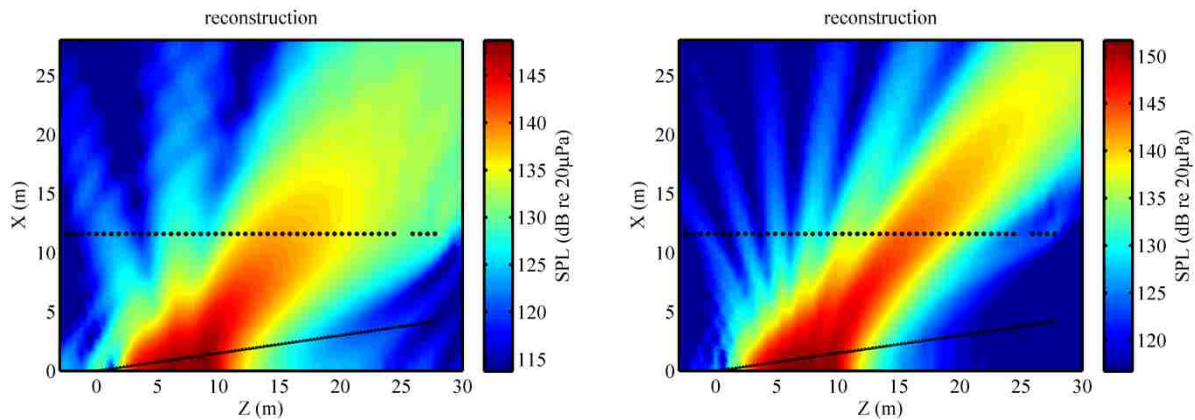
Figure 4.2 Reconstructions of numerical measurement using cylindrical NAH at (Left) 105 Hz and (Right) 210 Hz. Dots represent locations of numerical microphones. Slanted line is the estimated shear layer.

We now look along the shear layer as projected onto the Z axis at the reconstructed source locations and amplitude in Fig. 4.3 having used cylindrical NAH on the numerical measurement. Figure 4.4 shows the difference between the reconstruction and benchmark. The maximum source region occurs between Z=5 m and Z=10m over which the difference between reconstruction and benchmark is within 2 dB. Within the maximum source region, the actual maximum source location agrees within 1 dB.

This leads to the conclusion that even though the measurements occurred in the acoustic far field of the source, cylindrical NAH accurately reconstructs the source location and maximum amplitude propagating towards this numerical source. Consequently, when applying cylindrical NAH to actual measured data, confidence can be placed in the reconstructed source location and maximum amplitude.
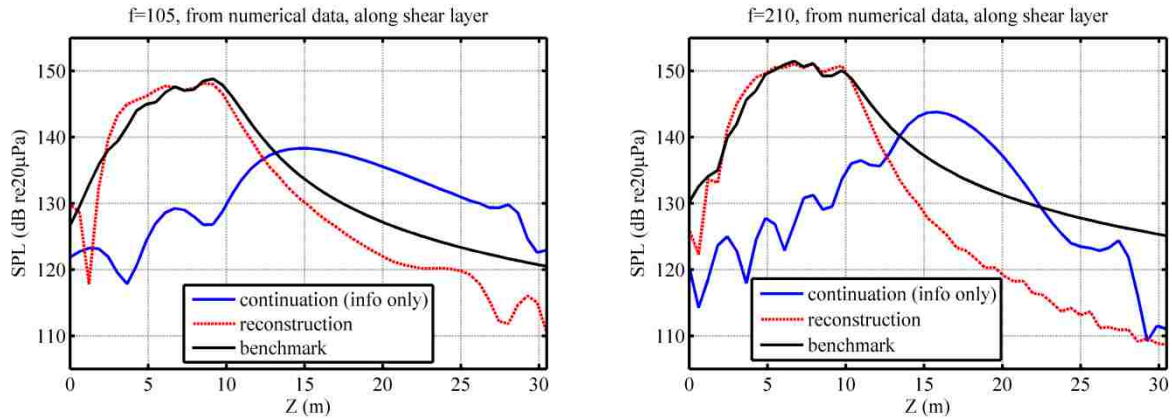
Figure 4.3 Reconstructions using cylindrical NAH along the shear layer of numerical measurement at (Left) 105 Hz and (Right) 210 Hz. Analytically continued measured data, labeled continuation (info only), are shown as a reference only.

Moving downstream along the shear layer, $Z > 12$ m in Fig. 4.3 and Fig. 4.4, there is a large discrepancy between benchmark and reconstruction not near the maximum source location. This is likely caused by having not measured the directional lobe(s) of which the downstream shear layer is a part. Also a factor could be the lack of evanescent wave information in the measurement which would further increase the reconstructed amplitude.

One important lesson from cylindrical NAH reconstructions along the shear layer is that once the pressure level falls more than 2 dB from the maximum, the discrepancy between reconstruction and benchmark increases quickly, shown in Fig. 4.3 and Fig. 4.4. Therefore, when applying cylindrical NAH to measured jet data, the location of maximum reconstructed source location along the shear layer will be the area encompassing a pressure 2 dB within the maximum reconstructed pressure
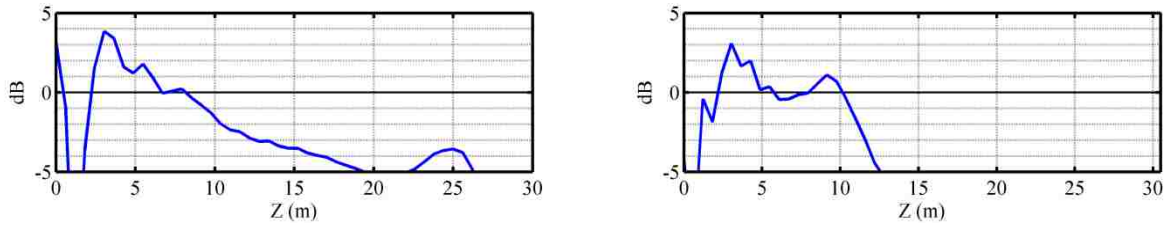
Figure 4.4 Difference between reconstruction and benchmark using cylindrical NAH along the shear layer of numerical measurement at (Left) 105 Hz and (Right) 210 Hz.

Outward propagation away from the source, from a theoretical standpoint, is a tremendously easier problem. The solution is a forward problem, as provided by Rayleigh's integral,[17, 37], and no regularization is necessary. The evanescent information that was decaying continues to do so. Propagating the sound simply involves a Green's function from one location to another, and no consideration of evanescent wave information need be made.

When propagating outwards, we can compare to measurements taken along the 23-m arc [see Fig. 2.2 for measurement schematic]. Reconstructions along the 23-m arc are shown in Fig. 4.5. Reconstructions match or slightly overpredict radiated sound pressures. Measurements along the arc encompass SPLs that are within ~15 dB of the maximum along the shear layer. Reconstructions along the arc agree to measurements within 2 dB. Therefore, for outward propagation, confidence is placed in cylindrical NAH reconstructions within 15 dB of the maximum SPL along the the shear layer.

Ideally, the entire outward reconstructed region would be trusted, but analytic continuations exterior to the measurement region purposefully taper towards zero, which may not be the actual case. Consequently, only the top 15 dB of outward cylindrical NAH reconstructs is valid.
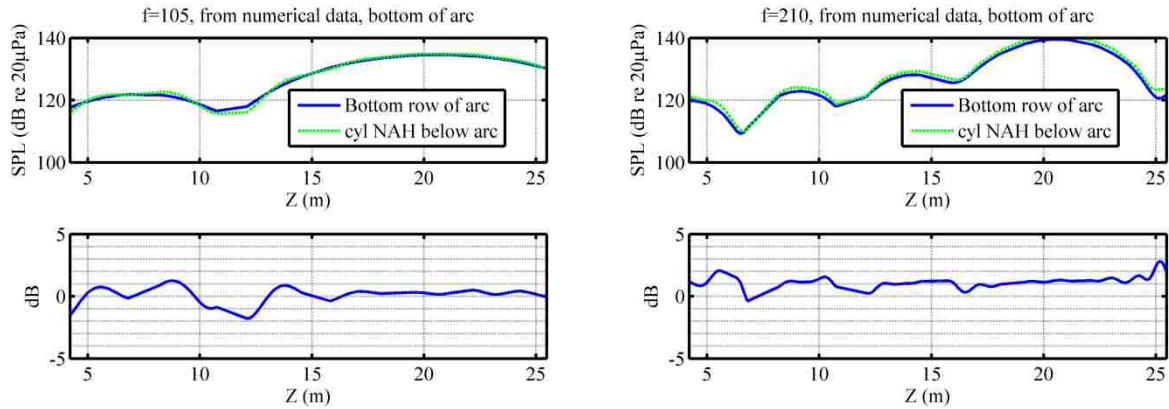
Figure 4.5 Reconstructions of numerical measurement along the bottom of 23-m arc using cylindrical NAH at (Left top) 105 Hz and (Right top) 210 Hz. (Left bottom) and (Right bottom) are the difference between reconstruction and numerical measurement along the bottom of the 23-m arc.

4.2 Numerical results for planar NAH

Now we move to numerical results using planar NAH. Again using the numerical source described in Section 3.6.1.1, numerical measurements were taken along plane 2 [see Fig. 2.2 for a measurement schematic] and propagated to both inwards to the shear layer and outwards to locations along the 23-m arc.

Significant differences are seen in maximum source amplitude along the shear layer as well as a large over-estimation of the sound pressure along the 23-m arc. The differences are caused by both the large measurement standoff distance and the planar Fourier NAH propagator, Eq. (3.17). Plane 2 has a standoff distance of roughly 6 m, over which distance the evanescent wave information decays below the noise, as discussed in Section 3.4. Because the evanescent wave information is in the noise floor, modified Tikhonov regularization in conjunction with the Morozov discrepancy principle (see Section 3.3), filtered virtually out all subsonic wave

information, meaning the remaining wavenumbers of significance are real. If the planar NAH pressure propagator, Eq. (3.17), only has real wave numbers, the radiated pressures are simply oscillating plane waves with no decrease in amplitude over any propagation distance. Their sum at a location farther away may yield decreased amplitude because of the plane waves' directions, but each individual plane wave with supersonic wave number does not decrease in amplitude. There comes a point radiating away from the source, at which the summation of plane waves begins to look locally planar and fewer plane waves are needed to represent geometrical speading. Conversely, moving towards the source, a greater number of plane waves is needed to represent geometrical spreading.

Looking at results along the shear layer, Fig. 4.6 shows both the two dimensional reconstruction along the estimated shear layer using planar NAH and the numerical benchmark at the same locations. Qualitatively, the reconstructions are underestimating the maximum source amplitude and give a general idea of where the maximum location is.
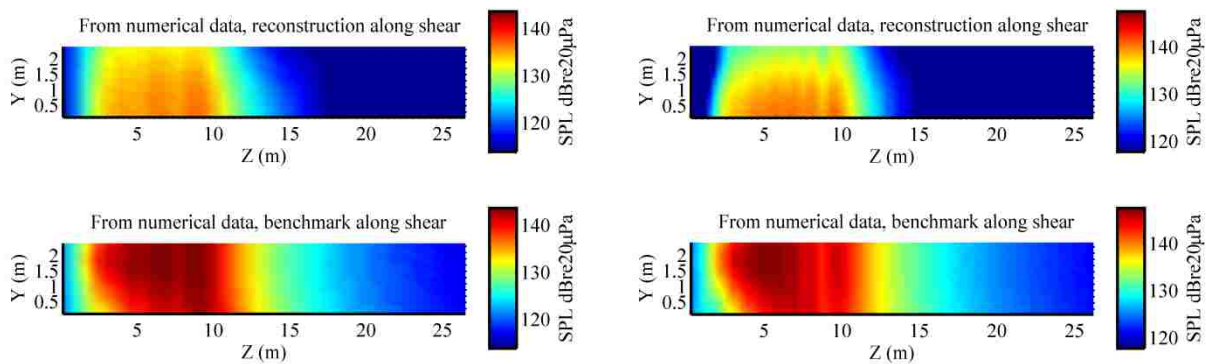
Figure 4.6 Benchmark and reconstructed data along the shear layer using planar NAH on numerical measurements at (Left) 105 Hz and (Right) 210 Hz. (Left and Right Top) Reconstruction. (Left and Right Bottom) Benchmark.

To quantitatively compare reconstructions and benchmark along the shear layer, horizontal cross-sections were taken three inches off the ground. Figure 4.7 shows the latter cross

sections at both 105 Hz and 210 Hz. The continued pressure field is included for reference. The reconstructed field does not increase in amplitude because the measurement standoff distance was too great to sufficiently record evanescent wave information above the noise floor. Consequently, no confidence can be placed in the maximum reconstructed source amplitude.

Determining the validity of the planar NAH reconstructed source location is similarly difficult because there is no way of telling where along the flat-top maximum source location the peak actually occurs. For example, in Fig. 4.7, at 105 Hz, the benchmark shows a maximum source location at $Z = 9$ m. The reconstruction, on the other had, is generally flat from $3<Z<9$. Perhaps a best guess would be the reconstruction at $Z = 9$ m, but such a guess is weak. More confidence can be placed in the cylindrical NAH reconstructions. At best, planar NAH gives a reconstruction aperture within which the maximum source location resides, but it does not specify where the maximum location is. The maximum source location, on the other hand, coincides with the benchmark.
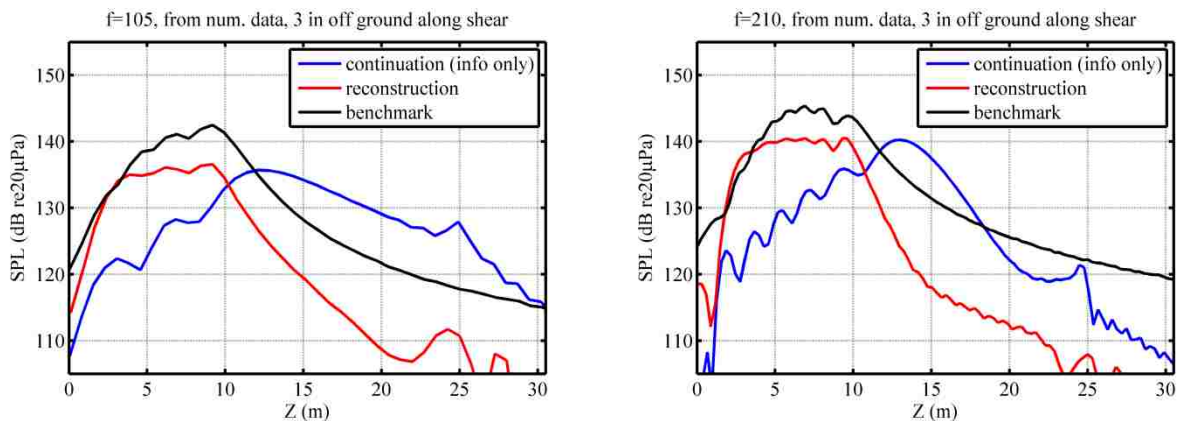
Figure 4.7 Benchmark and reconstructed data 3 inches off the ground along the shear layer using planar NAH on numerical measurements at (Left) 105 Hz and (Right) 210 Hz.

If we now look at outward propagation to locations along the bottom of the 23-m arc, Fig. 4.8 shows that planar NAH overestimates the pressure along the 23-m arc. In terms of

geometrical spreading, the outward radiated sound should decrease in amplitude. However, as discussed previously in this section, the propagated waves using planar NAH do not decrease in amplitude. By the time they reach the 23-m arc, their summation has not decreased sufficiently from the waves spreading out, thus overestimating the outward radiated sound pressures.

While planar NAH reconstructions and benchmark measurements in Fig. 4.8 along the arc agree within almost 5 dB, the agreement is not as good as cylindrical NAH reconstructions in Fig. 4.5.
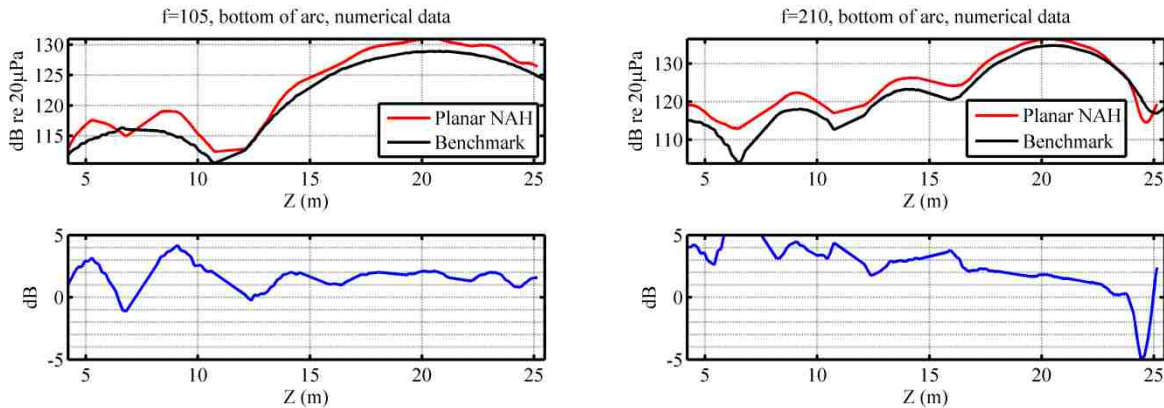
Figure 4.8 Reconstructions using cylindrical NAH on numerical measurments along the 23-m arc at (Left) 105 Hz and (Right) 210 Hz. (Left bottom) and (Right bottom) are the difference between reconstruction and numerical measurement along the bottom of the 23-m arc.

4.3 Discussion of numerical results

There are many things to learn from the numerical simulations. First we learn cylindrical NAH best reconstructs the maximum source location and amplitude along the shear layer. The peak itself is accurate within 2 dB. And the region over which the reconstruction is valid is that which encompasses pressures within 2 dB of the maximum pressure.

Significant differences are seen between the results of cylindrical NAH and planar NAH, particularly the pressure reconstructions along the shear layer, Fig. 4.3 and Fig. 4.7, and how well they match the benchmark. Directional information must be recorded in the measurement to be accurately reconstructed.

Differences between cylindrical and planar NAH pressure reconstructions can be attributed to the pressure propagators. The cylindrical NAH propagator Eq. (3.16) includes geometrical spreading for supersonic waves, those with wavenumber magnitude less than the acoustic wavenumber. Whereas the planar NAH pressure propagator Eq. (3.17) has no inherent geometrical spreading built in to the propagator. Both cylindrical and planar NAH yield near matching maximum source regions. But only cylindrical NAH accurately reconstructs the maximum source level location and amplitude. Consequently, in this work, the method of NAH that gives the most accurate reconstructed source amplitude and location is cylindrical NAH.

Something else we learn from the numerical simulations is that if a directional lobe is not well recorded in the measurement, the source from which it stems is not accurately reconstructed. For instance, the directional lobe downstream in Fig. 4.1 is not well reconstructed in Fig. 4.2.

From the numerical results we learn the reconstruction aperture that we can trust along the shear layer is the region using cylindrical NAH that encompasses 2 dB of the maximum SPL. For the numerical results that was 5<Z<10 m along the shear layer. Planar NAH does not give a specific maximum source location, but it does give a region within which the maximum level is located.

The following chapter presents results using cylindrical and planar NAH on actual measured data.

5 NAH pressure results from measured jet data

With an idea of the succeses and limitations of the NAH processing, we are now in a position of implementing the process on actual measured data. We find that results along the 23-m arc agree with benchmark measurements, as did the numerical simulations. In addition, comparing the shapes of the shear layer reconstructions offers some valuable insights. At 105 Hz, the maximum region of the shear layer reconstruction is fairly broad compared to the maximum region at 210 Hz. This indicates the source at the lower frequency is larger and more distributed than the source at the higher frequency.

With increasing frequency, the maximum source region along the shear layer moves upstream. Additionally, the reconstructed directionality is shown to move upstream with smaller polar angle as measured from the jet inlet. Shifting directionality of this sort towards smaller polar angle with increasing frequency qualitatively correlates to measurements of full-scale supersonic jets and model-scale jets.[49]

As results along the estimated shear layer are presented, key features to look for are narrowing of the reconstructed source region and an upstream movement of the maximum reconstructed source location.

Like the numerical study, the frequencies of interest in this study are limited to 105 Hz and 210 Hz. As shown in equation (3.11), the 2-ft spacing of the ground microphone array limits cylindrical NAH analysis to less than $f_{max} = c/2\,\Delta z = 343/2(2\cdot.3048)\ = 281$ Hz. Also, from Fig. 2.3, the maximum sound pressure level occurs somewhere between 100 and 200 Hz. Consequently, we focus on 105 Hz and 210 Hz.

5.1 Cylindrical NAH pressure results from measured jet data

Before calculating the pressure results from cylindrical NAH, the measured phase across

the ground reference microphone array needs to be addressed. The ground reference array

consists of 46 microphones with 0.6096-m (2-ft) spacing, then a 1.524-m (5-ft) gap, followed by

4 microphones with 0.6096-m (2-ft) spacing (see Fig. 2.2). To utilize the discrete spatial Fourier

transform, the microphones need to have equal spacing along the whole array. Accordingly, the

measured amplitudes and phases are interpolated using a spline interpolation. The interpolation

for four data blocks at 105 Hz is shown in Fig. 5.1. The interpolation at 210 Hz requires a bit

more finesse. As can be seen in Fig. 5.2, the unwrapped phase of the last 4 microphones does not

follow the same smoothly varying trend as the previous microphones. This is because the phase

goes through a phase change greater than π between microphone 46 and the one spaced 1.524 m

(5 ft) away. Consequently, when MATLAB$^{®}$ unwraps the phase, it gives a phase that is

incorrectly shifted by some multiple of π. To correct this, -2π is added to the unwrapped phase;

then the phase is interpolated using a spline interpolation, also shown in Fig. 5.2.
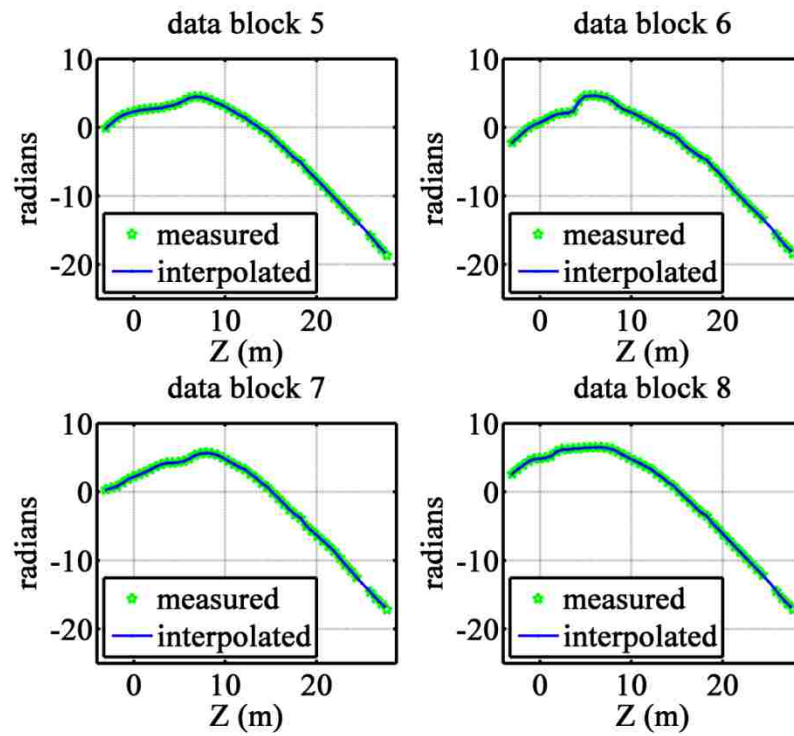
Figure 5.1 Unwrapped measured phase and interpolated phase of four data blocks at 105 Hz, military engine condition, from linear ground reference array. Spline interpolation is used to interpolate between first 46 microphones with 0.6096-m spacing, the 1.524-m gap, and the remaining 4 microphones with 0.6096-m spacing to achieve 0.6096-m spacing over 51 measurement points.
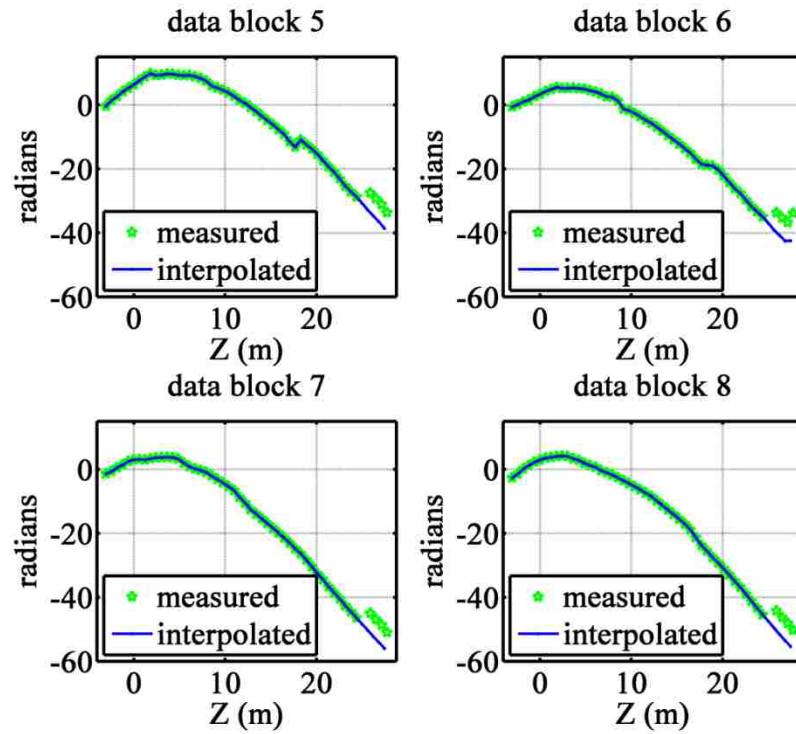
Figure 5.2 Unwrapped measured phase and interpolated phase of four data blocks at 210 Hz, military engine condition, from linear ground reference array. Because phase differences greater than π occur in the 1.524-m gap between consecutive microphones, -2π radians of phase is added to each data block of the last four microphones before interpolation.

Once the measured line array data are interpolated, the measurement points are then input into the STSF processing, the resulting singular vectors are analytically continued 100 points on both sides to avoid wrap-around error, and the pressures or particle velocities are propagated according to Eqs. (3.14) or (3.22) and (3.24).

The following two subsections show results for cylindrical NAH at 105 Hz and 210 Hz military power engine condition.

5.1.1 Cylindrical NAH results for 105 Hz, military power engine condition

Figure 5.3 shows reconstructed sound pressure levels using cylindrical NAH at 105 Hz, military power engine condition. The continued region upstream and most of the one downstream have been discarded. The dotted lines show the locations of the ground microphones. The dotted, angled line shows the estimated shear layer location, interior to which the results are erroneous due to the extreme mean air flow and temperature gradient. Both the mean air flow and temperature gradient would put the source location more upstream than predicted in these results. From the numerical experiment in Section 4.1, we learn that the reconstructions at location $(Z, X) = (Z>20, X<11)$ cannot be trusted because the measurement aperture was not sufficiently large to record data downstream of the main lobe because of the source directionality. Similarly for upstream of the main lobe $(Z<0, X<11)$. The usable bound along the shear layer is the region which encompasses the top 2 dB SPL, shown more explicitly in Fig. 5.6. Off of the shear layer, usable reconstructions are those which have a SPL within 15 dB of the maximum, which here are the reconstructions greater than 118 dB. Essentially, the usable reconstructions are those which follow the main lobe of the reconstruction.
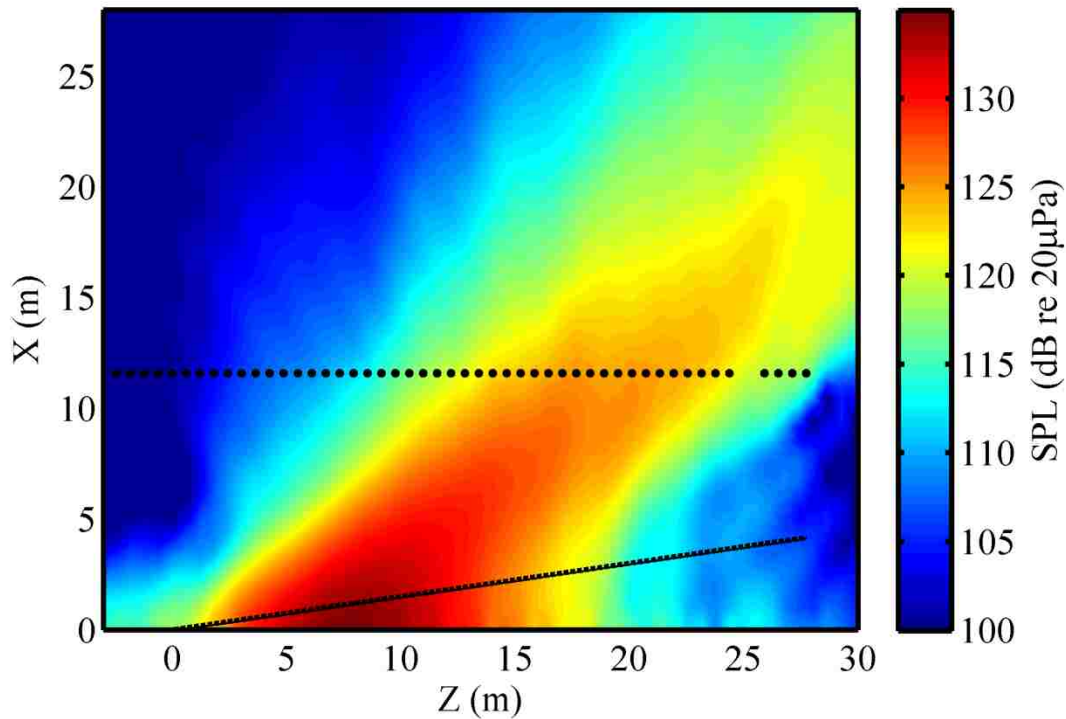
Figure 5.3 Pressure reconstructions using cylindrical NAH at 105 Hz, military engine condition.

Figure 5.4 shows the same reconstruction as Fig. 5.3 with the addition of measured data along the 23-m arc. The estimated values of cylindrical NAH are further compared to the measured data along the 23-m arc in Fig. 5.5, which plots the bottom row of measured data along the 23-m arc and cylindrical NAH estimates below the arc and 0.0762 m (3 in) off the ground. The horizontal axis of Fig. 5.5 is the curved surface of the arc projected onto the Z axis. Also shown in Fig. 5.5 is the difference between the cylindrical NAH reconstruction below the arc and the bottom row of data long the arc.

Similar to the numerical simulations in Fig. 4.5, reconstructions along the arc in Fig. 5.5 using cylindrical NAH agree with and overestimate measured data within 2 dB.
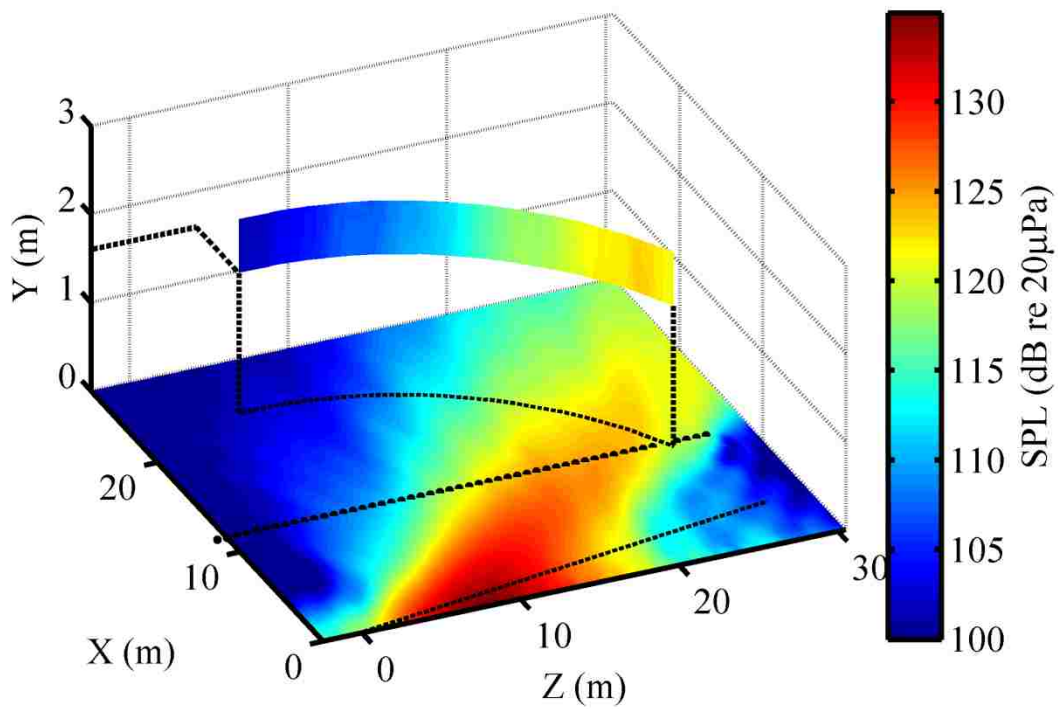
ss

Figure 5.4 Pressure reconstruction using cylindrical NAH at 105 Hz, military engine condition. Also shown is measured data along the 23-m arc. Dotted lines extending from arc are to show geometrical location.
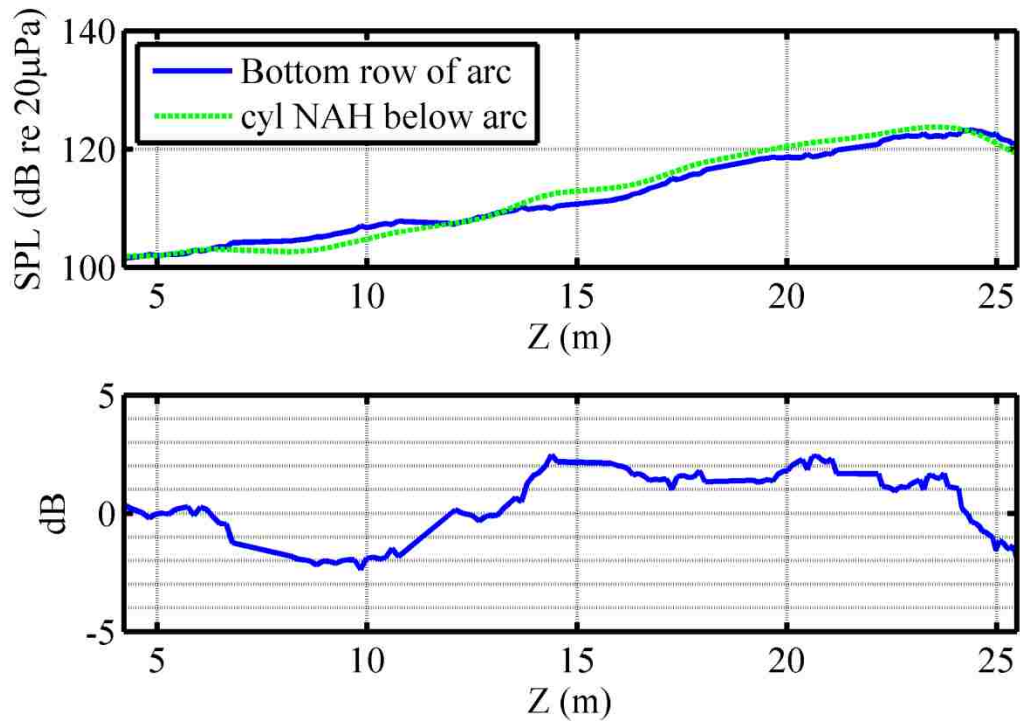
Figure 5.5 (Top) Comparison at 105 Hz, military engine condition, between the bottom row of microphones along the 23-m arc (160 cm off the ground) and cylindrical NAH reconstructions below the arc, 7.62 cm off the ground. (Bottom) The difference between the cylindrical NAH reconstruction and the bottom row along the 23-m arc.

Lastly for cylindrical NAH results for 105 Hz, military engine condition, we have estimated sound pressure levels along the shear layer 3 inches off the ground. A height of 3 inches off the ground was chosen so direct comparison may be made to levels estimated by planar NAH. Because the shear layer is angled with respect to the central axis and because each measurement position only propagates in the radial direction, Fig. 5.6 was obtained by propagating to multiple radii and the appropriate value picked off. Figure 5.6 shows cylindrical NAH reconstructs a maximum sound pressure level along the shear layer 7.62 cm (3 in) off the ground of 134 dB re 20 μPa at $z = 7.9$ m.

Numerical simulations of Chapter 4 showed the usable maximum source region along the shear layer is that which is within 2 dB of the maximum.For cylindrical NAH on measured jet

noise at 105 Hz, Figure 5.6 shows the usable maximum source region 5<Z<11m. The location and shape of the maximum source region is of particular interest as we compare to cylindrical NAH results at a higher frequency (210 Hz). The maximum source region will move upstream and narrow.
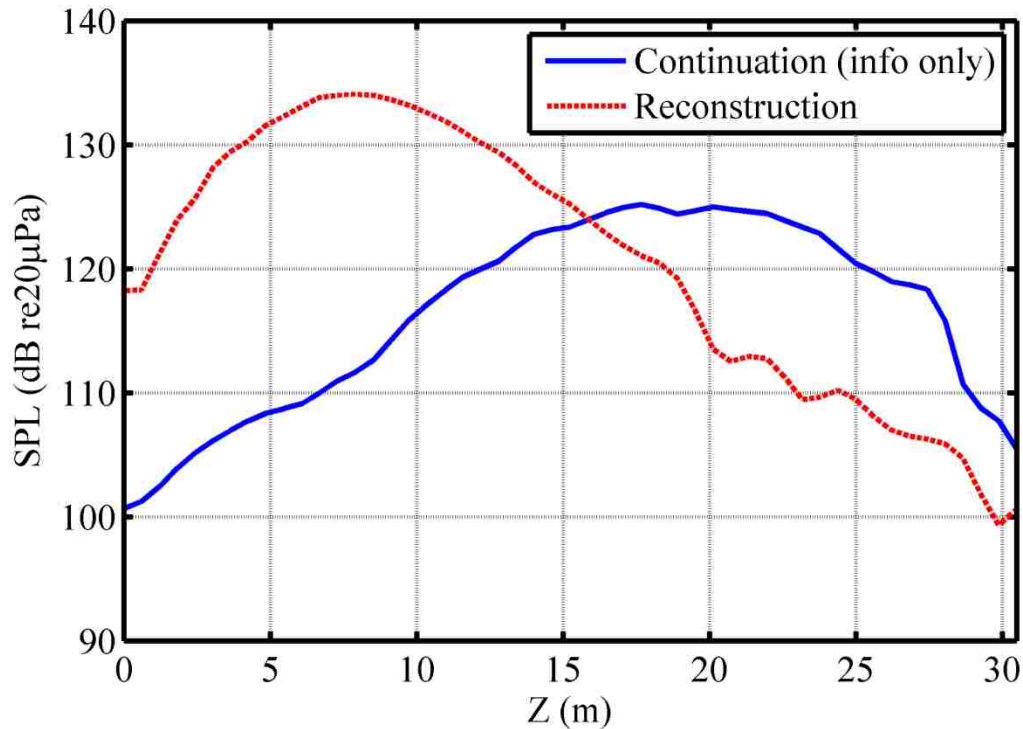
Figure 5.6 Pressure reconstructions along the shear layer and 7.62 cm (3 in) off the ground using cylindrical NAH at 105 Hz, military engine condition. Each point intersected the shear layer at different radii; consequently the continued pressures were propagated to the various radii and the appropriate value picked off. This shows cylindrical NAH estimates a maximum amplitude at z = 7.9 m along the shear layer at 105 Hz, military engine condition.

5.1.2 Cylindrical NAH results for 210 Hz, military power engine condition

Similar in format to Fig. 5.3 but at 210 Hz instead of 105 Hz, Fig. 5.7 shows the pressure reconstructions using cylindrical NAH. As described in Section 5.1, -2π was added to the unwrapped phase of the last four microphones prior to interpolation to account for the phase wrapping over the 1.524-m gap.
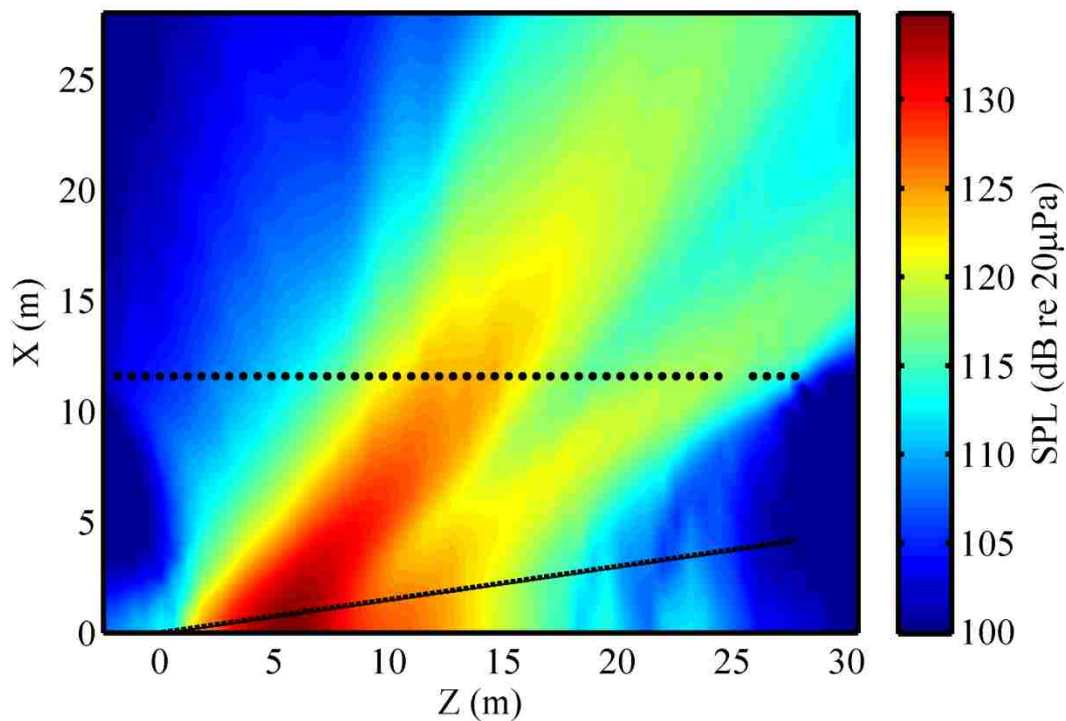
Figure 5.7 Pressure reconstructions using cylindrical NAH at 210 Hz, military engine condition.

Similar in format to Fig. 5.4 but for 210 Hz, Fig. 5.8 shows reconstructed sound pressure levels using cylindrical NAH in addition to measured data along the 23-m arc. The bottom row of the arc is compared with cylindrical NAH reconstructions in Fig. 5.9. As did reconstructions along the arc at 105 Hz, reconstructions at 210 Hz overestimate the radiated pressure.
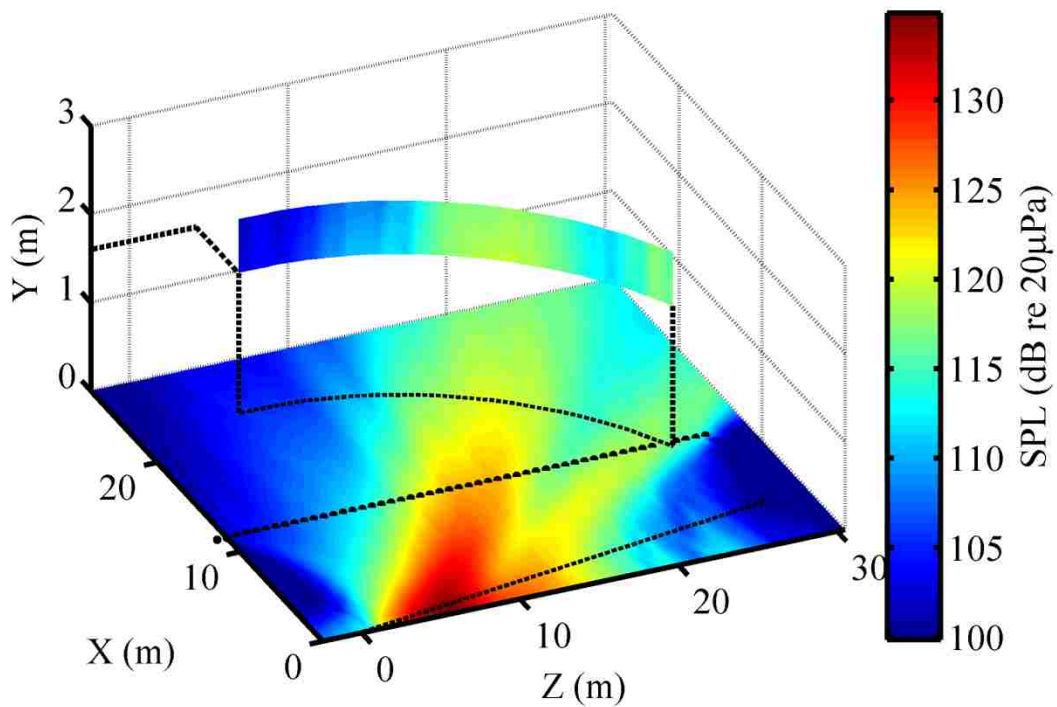
Figure 5.8 Pressure reconstructions using cylindrical NAH at 210 Hz, military engine condition. Also shown is measured data along the 23-m arc. Dotted lines extending from arc are to show geometrical location.

An interesting observation is that the reconstructions along the arc appear to be shifted slightly upstream. In other words, in Fig. 5.9 the measured pressures along the arc are more downstream than the reconstructions. Perhaps this is from a downstream mean air flow not accounted for in the cylindrical processing.
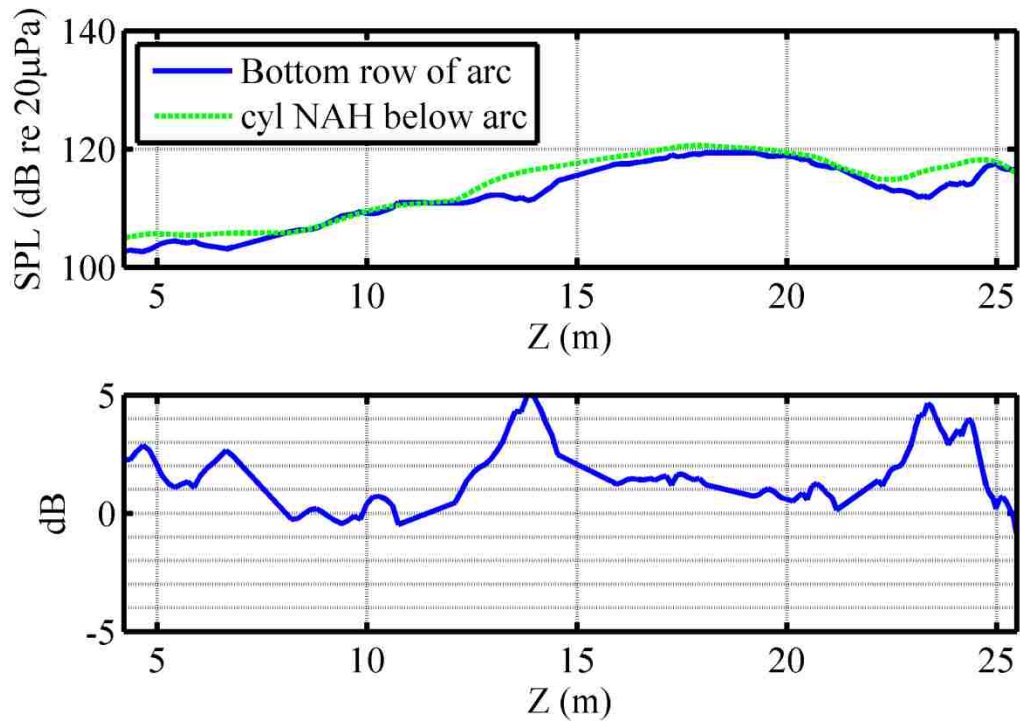
Figure 5.9 (Top) Comparison at 210 Hz, military engine condition, between the bottom row of microphones along the 23-m arc (1.60 m off the ground) and cylindrical NAH reconstructions below the arc, 7.62 cm off the ground. (Bottom) The difference between the cylindrical NAH reconstruction and the bottom row along the 23-m arc.

Finally for pressure reconstructions using cylindrical NAH at 210 Hz, military engine condition, we have the estimated sound pressure level along the shear layer 7.62 cm (3 in) off the ground in Fig. 5.10. Cylindrical NAH predicts a maximum sound pressure level along the shear layer of 134 dB re20 μPa at $z = 6$ m downstream. The usable maximum source region, as defined in Chapter 4 to be the region within 2 dB of the maximum, for cylindrical NAH at 210 Hz is 4<Z<8 m. The maximum pressure location has moved upstream from the cylindrical NAH results at 105 Hz. Furthermore, the shape of the maximum source pressure distribution has narrowed. The latter two qualitatively concur with directionality measurements of full-scale supersonic jets and model-scale jets.[49]
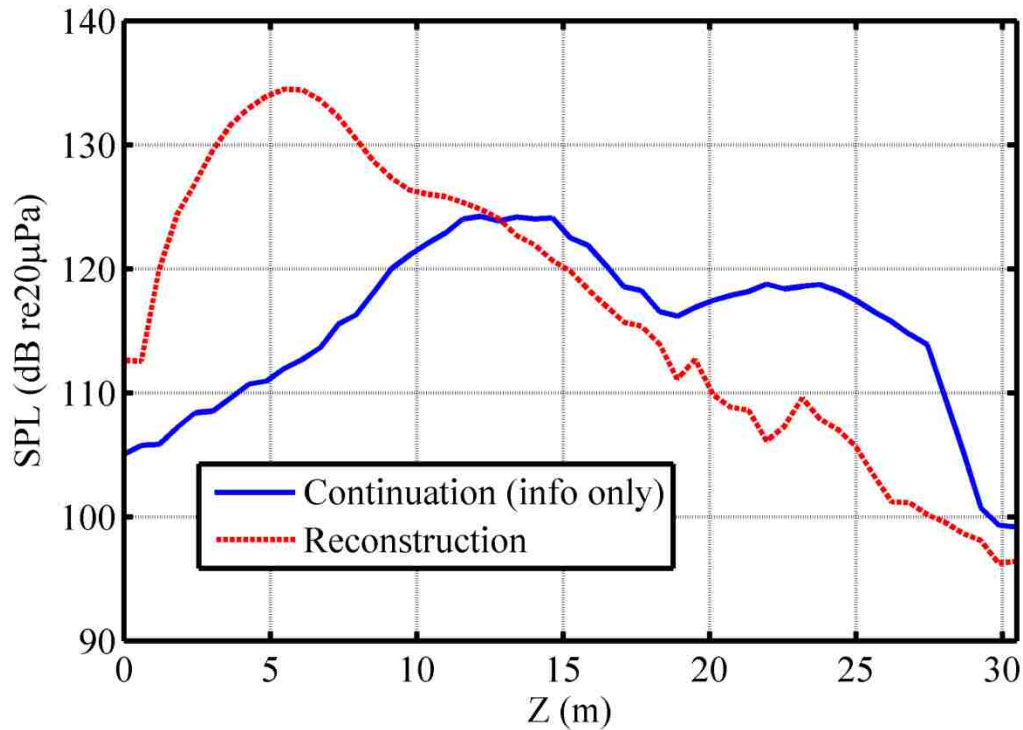
Figure 5.10 Pressure reconstructions along the shear layer and 7.62 cm (3 in) off the ground using cylindrical NAH at 210 Hz, military engine condition. This shows cylindrical NAH estimates a maximum sound pressure level on the shear layer at $z = 6.0$ m downstream.

5.2 Planar NAH pressure results from measured jet data

Now we move to pressure reconstructions using planar NAH. While there are multiple planes to choose from, we will constrain our analysis to measurement plane 2 shown in Fig. 2.2 as parallel to measurement plane 1 and the estimated shear layer. Using measurement plane 2 allows for direct comparison to measurement plane 1 from a back-propagation standpoint. As discussed in Section 3.4, the measured data was mirrored about the ground and interpolated such that there is a maximum level at the ground. Then analytic continuation was applied to extend

the measurement aperture. For the planar NAH results shown, the aperture was continued 10 m upstream, 30 m downstream, and 20 m both vertically up and down. Because analytic continuation has the ability to correctly interpolate the data within in the measurement aperture, the number of measured data points in the horizontal direction was reduced to include only four per wavelength. This resulted in reduced computation time for analytic continuation on the order of 10 minutes per continuation for 105 Hz and 20 minutes per continuation for 210 Hz.

Once extended pressures were obtained through analytic continuation, the pressures were propagated using equations (3.33) and (3.34). The following two subsections show pressure reconstructions using planar NAH at 105 Hz and 210 Hz military power engine condition.

5.2.1 Planar NAH results for 105 Hz, military power engine condition

Figure 5.11 shows pressure reconstructions for planar NAH at 105 Hz, military power engine condition. The solid black rectangle outlines the measurement plane. Cutting through reconstruction planes are the measured data along the 23-m arc. The outward propagation overestimates the pressure in some locations by up to less than 5 dB. The rippling levels at 110 dB around $(Z, X) = (0,25)$ m are artifacts of wrap-around error from the directional lobe leaving one end of the continued region and entering the other end. It could be lessened by continuing horizontally more points. The reconstructed points below the ground and more than 15 cm (6 in) above the measurement aperture were discarded
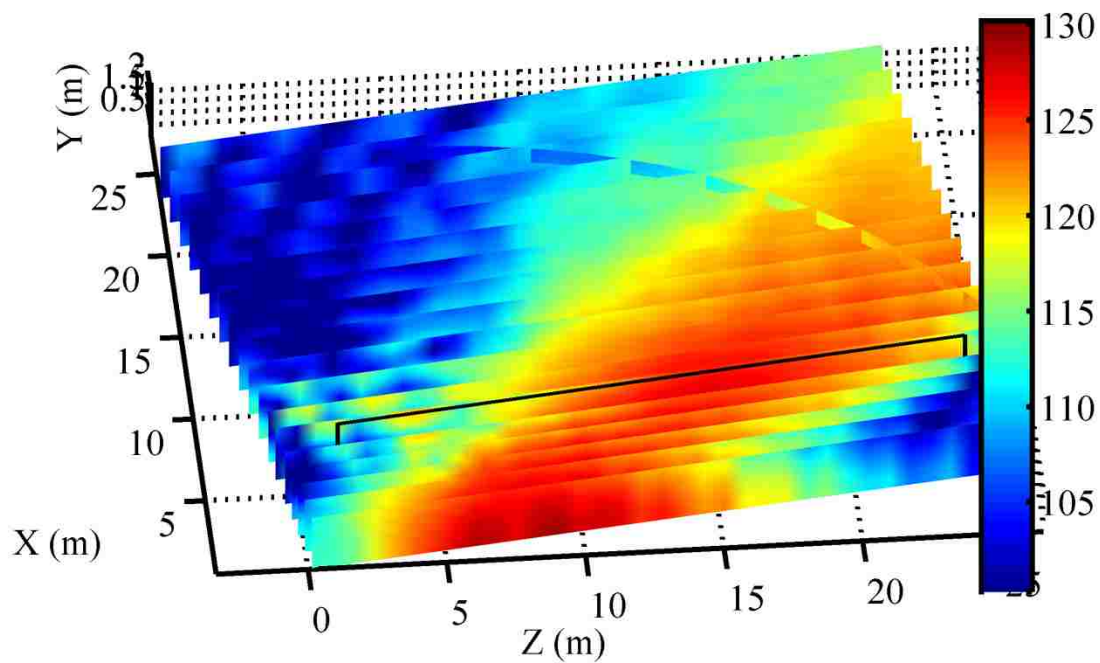
Figure 5.11 Pressure reconstruction planes using planar NAH at 105 Hz, military engine condition. The solid black rectangle outlines the measurement region. The arc cutting through measurement planes is measured data from the 23-m arc used as a benchmark. Reconstructions have been truncated from the full analytically continued region.

In Fig. 5.12 we have a comparison at 105 Hz, military engine condition, of (top) measured data at plane 1 and (bottom) back-propagation using planar NAH from plane 2 to the locations of plane 1. Notice how the planar NAH processing greatly smoothes the data.
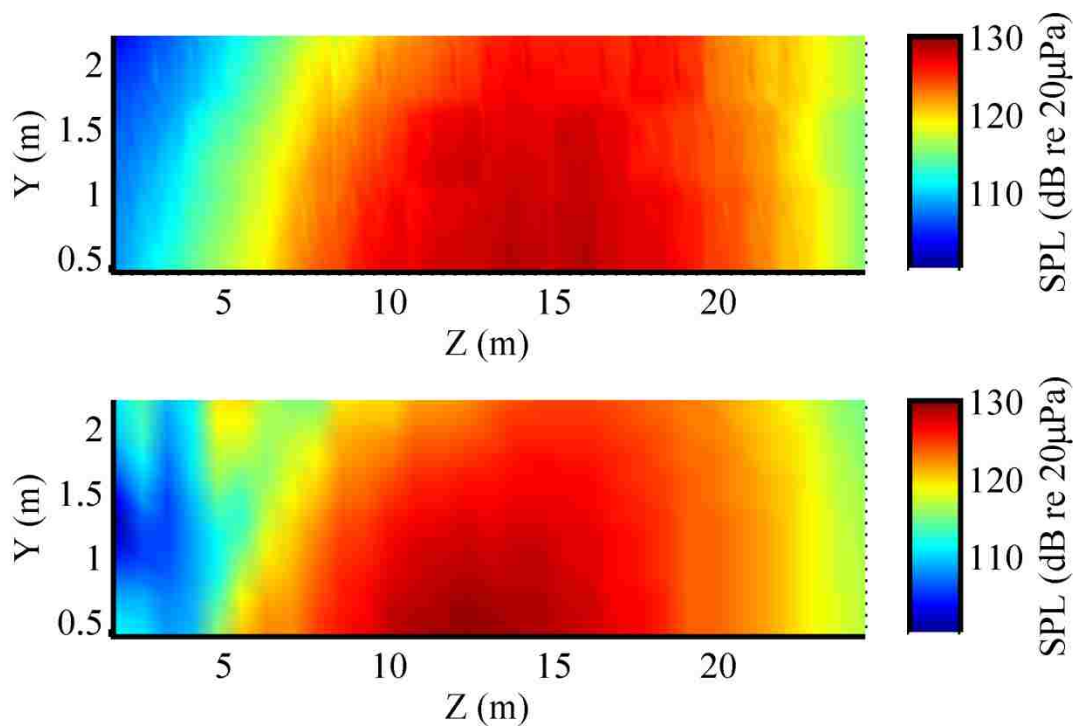
Figure 5.12 (Top) Benchmark measurement at 105 Hz, military engine condition, located at measurement plane 1. (Bottom) Reconstruction by back-propagation from plane 2 to plane 1. (See Figure 2.2 for measurement plane locations.)

Figure 5.13 shows reconstructed pressures along the shear layer 7.62 cm (3 in) off the ground shown in. A height 7.62 cm off the ground was chosen because the lowest row of microphones on the rig was 38 cm (1.25 ft), and 7.62 cm off the ground resulted from trying to maintain the 15-cm (6-in) rig microphone spacing while interpolating prior to analytic continuation. As discussed in Chapter 4, the maximum level is grossly underestimated and the maximum location resulting from planar NAH may not be accurate. Instead, planar NAH gives a maximum source region, the top 2 dB of the reconstructed response, within which the maximum location resides. Planar NAH predicts the maximum source region along the shear layer for 105 Hz military power engine condition is 6<Z<12 m and an underestimated maximum level of 128 dB re 20μPa.
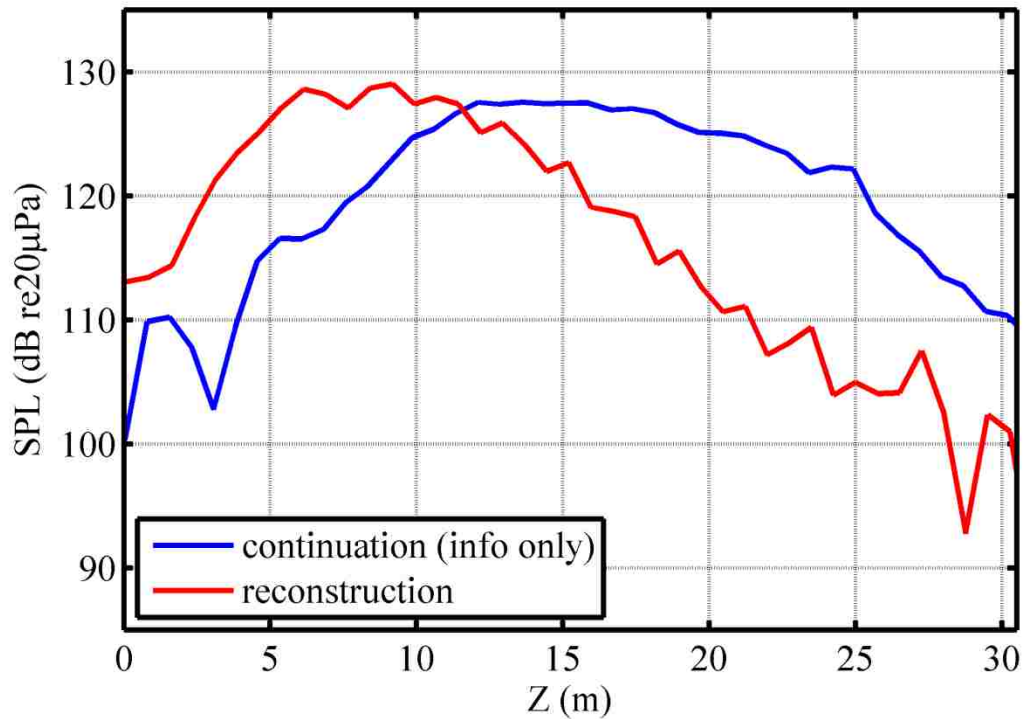
Figure 5.13 Pressure reconstructions along the shear layer and 7.62 cm (3 in) off the ground using planar NAH at 105 Hz, military engine condition. This shows planar NAH estimates a maximum amplitude at 7.7 m along the shear layer at 105 Hz, military engine condition.

Having propagated outwards along the 23-m arc, Fig. 5.14 shows pressure reconstructions compared to measured data along the bottom of the arc. Also shown is the difference between reconstruction and measured data. The horizontal axis is the locations along the 23-m arc projected onto the Z axis. Similar to the numerical simulation results in Fig. 4.8, planar NAH overestimates the pressure along the 23-m arc. As discussed in section 4.3 Discussion of numerical results, the outward propagations overestimate the pressure because only supersonic wave information was recorded, which the planar NAH pressure propagator, Eq. (3.17), propagates without reduction in level and the summation of waves at the arc locations has not reduced in level.
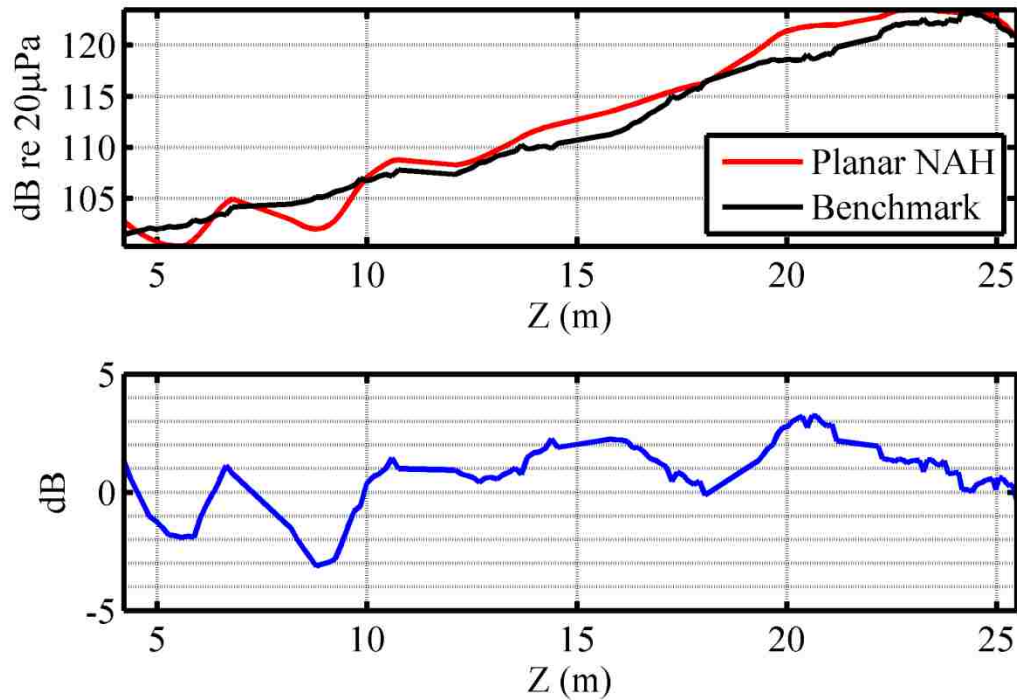
Figure 5.14 (Top) Pressure reconstruction and measured data along the bottom of the 23-m arc using planar NAH at 105 Hz, military engine condition. (Bottom) Difference between reconstruction and measured data.

5.2.2 Planar NAH results for 210 Hz, military power engine condition

We now look at planar NAH pressure reconstructions for 210 Hz, military power engine condition in Fig. 5.15, which is identical in layout to Fig. 5.11. Similar to 105 Hz, the rippling about 110 dB at $(Z, X) = (0,25)$ m is caused by wrap-around error and may be lessened by analytically continuing more points in the upstream direction. Notice the maximum pressure region along the shear layer has moved upstream compared to the 105-Hz results in Fig. 5.11. The movement upstream of the maximum pressure region along the shear layer will be discussed in greater detail in the conclusion section.
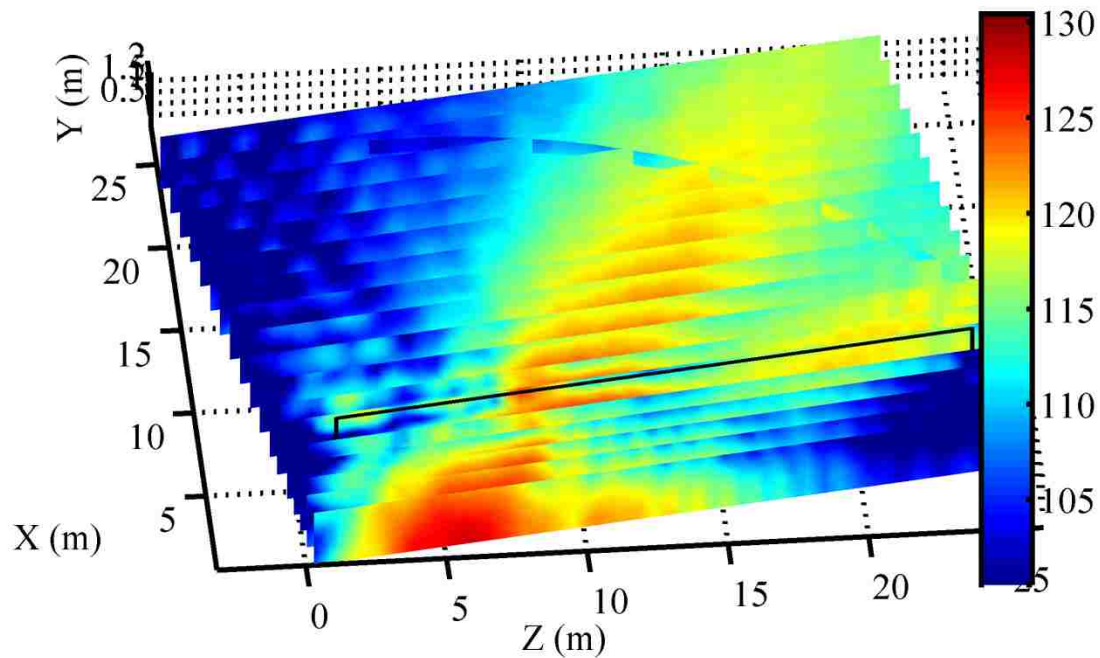
Figure 5.15 Pressure reconstruction planes using planar NAH at 210 Hz, military engine condition. The solid black rectangle outlines the measurement region. The arc cutting through measurement planes is measured data from the 23-m arc used as a benchmark. Reconstructions have been truncated from the full analytically continued region.

Having measured two planes of data parallel to each other (see Fig. 2.2 for measurement locations), we are able to back-propagate using planar NAH from plane 2 to plane 1 at 210 Hz, military engine condition as in Fig. 5.16. The top shows the measured data at plane 1 with a ground-reflection destructive-interference null running horizontally across the figure. The bottom shows the reconstruction via planar NAH. The maximum pressure region is reconstructed but the increase in amplitude above the ground interference null is not. This is likely due to how analytic continuation decreases the amplitude external to the measurement aperture; whereas in actuality, and as can be inferred from the top plot, the amplitude above the ground interference null would increase external to the measurement aperture.
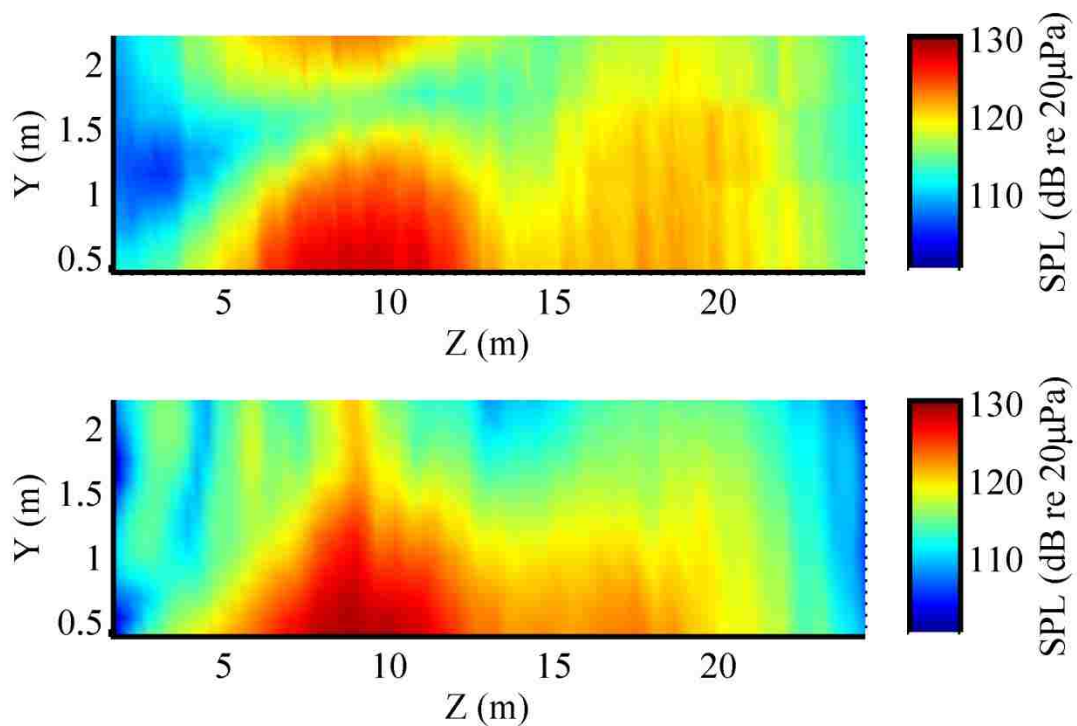
Figure 5.16 (Top) Benchmark measurement at 210 Hz, military engine condition, located at measurement plane 1. (Bottom) Reconstruction by back-propagation from plane 2 to plane 1.(See Figure 2.2 for measurement plane locations.)

Figure 5.17 shows planar NAH pressure reconstructions 7.62 cm (3 in) off the ground along the shear layer at 210 Hz, military engine condition. As discussed in Chapter 4, the maximum reconstructed pressure and location from planar NAH are not accurate, but the maximum location will be withing the region encompassing the top 2 dB. Planar NAH estimates the maximum source region at 210 Hz is 4<Z<7 m and an underestimated maximum level of 127 dB re 20µPa.
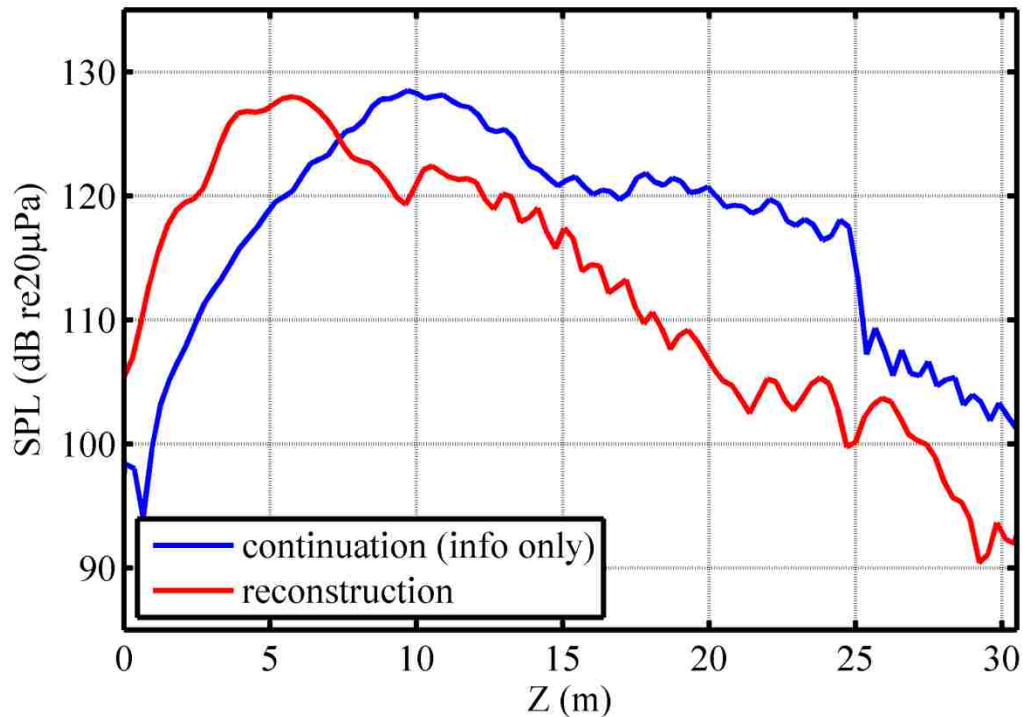
Figure 5.17 Pressure reconstructions along the shear layer and 7.62 cm (3 in) off the ground using planar NAH at 210 Hz, military engine condition. This shows planar NAH estimates a maximum sound pressure level 5.2 m on the shear layer at $z = 5.2$ m downstream.

Comparing reconstructions along the bottom of the 23-m arc we have Fig. 5.18. As in the numerical example, planar NAH reconstructions overestimate the pressure along the 23-m arc. Similar to reconstructions for 210 Hz using cylindrical NAH in Fig. 5.9, the reconstruction appears upstream of the measurements along the arc. Or rather, the measurements along the arc are more downstream than the reconstructions.
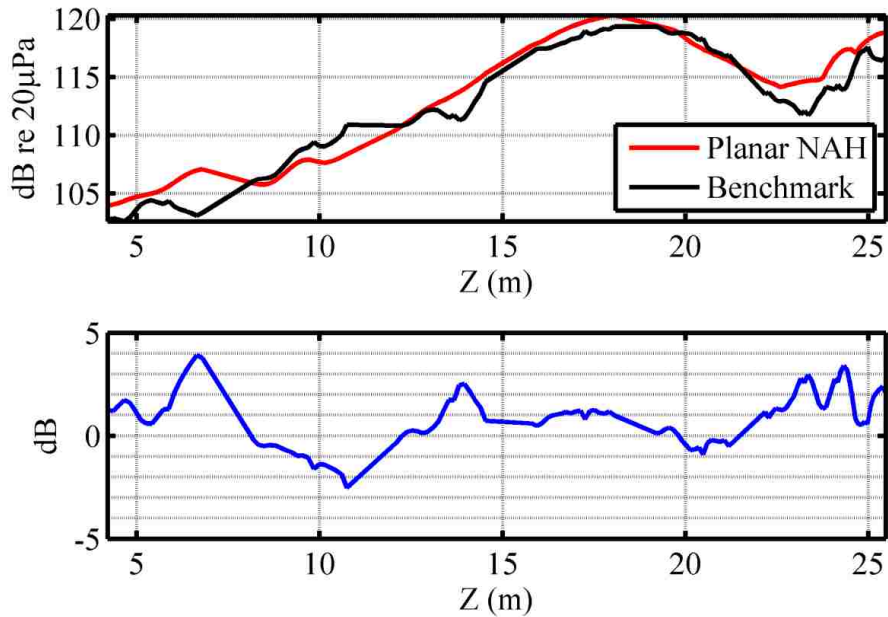
Figure 5.18 (Top) Pressure reconstruction and measured data along the 23-m arc using planar NAH at 210 Hz military power engine condition. (Bottom) Different between planar NAH and benchmark measured data.

Table 5.1 shows the maximum source region estimates along the shear layer 7.62 cm (3 in) above the ground. Because of the microphone spacing in the *z*-direction, the spatial resolution is limited to ±0.3 m for cylindrical NAH and ±0.2 m for planar NAH. Pinpointing a maximum source location with exact precision is difficult because the noise comes from turbulence that has some distributed volume. As discussed previously in Chapter 4, the maximum source levels reported by planar NAH are too low.

	f (Hz)	Z (m)	SPL (dB re20μPa)
Cylindrical NAH	105	5<Z<11	134
Planar NAH	105	6<Z<12	128
Cylindrical NAH	210	4<Z<8	134
Planar NAH	210	4<Z<7	127

Table 5.1 Maximum source region and maximum level estimates along the shear layer 7.62 cm (3 in) above the ground.

6 Intensity results

With the NAH processing and acoustic pressure and acoustic particle velocity propagation outlined in Chapter 3, we are now able to calculated estimated, time-averaged acoustic intensity and compare it to acoustic intensity measured with the tetrahedral intensity probe. Intensity estimates through NAH were calculated by propagating the components of intensity for each singular vector (cylindrical NAH) and adding as vectors the results from all the propagated singular vectors.

Because the intensity vectors have hugely varying magnitudes within a given plot and to make the intensity vectors easy to see, the vectors have been scaled on a logarithmic scale referenced to a small value that keeps the decibel magnitude values positive (i.e. the argument within the logarithm greater than 1). The resulting scaled magnitudes then compare only in a logarithmic sense. Short vectors can then be considered as much less prominent than long vectors. The direction of each intensity vector remains true to its pre-scaling direction.

6.1 Intensity from cylindrical NAH propagation of measured jet data

This section shows acoustic intensity results from cylindrical NAH for 105 Hz and 210 Hz, military power engine condition. Figure 6.1 shows intensity results from 105 Hz overlaid on the pressure results. Black dots show the ground microphone locations, and the angled line shows the estimated shear layer. Certain characteristics stand out. High vorticity near the end of the microphone array is likely a byproduct of analytic continuation. As seen in Fig. 3.6, analytic continuation reduces the pressure smoothly, but quite rapidly, exterior to the measurement edge;

93

more rapid, perhaps, than it would actually drop. This rapid reduction in amplitude manifests itself in vorticity of the intensity vectors. Another interesting characteristic is how the sound radiates upstream for locations upstream of the maximum source region. Intuitively this makes sense, that while there is obviously a preferred directionality, being a real system, the jet radiates some sound upstream, albeit at much lower levels.
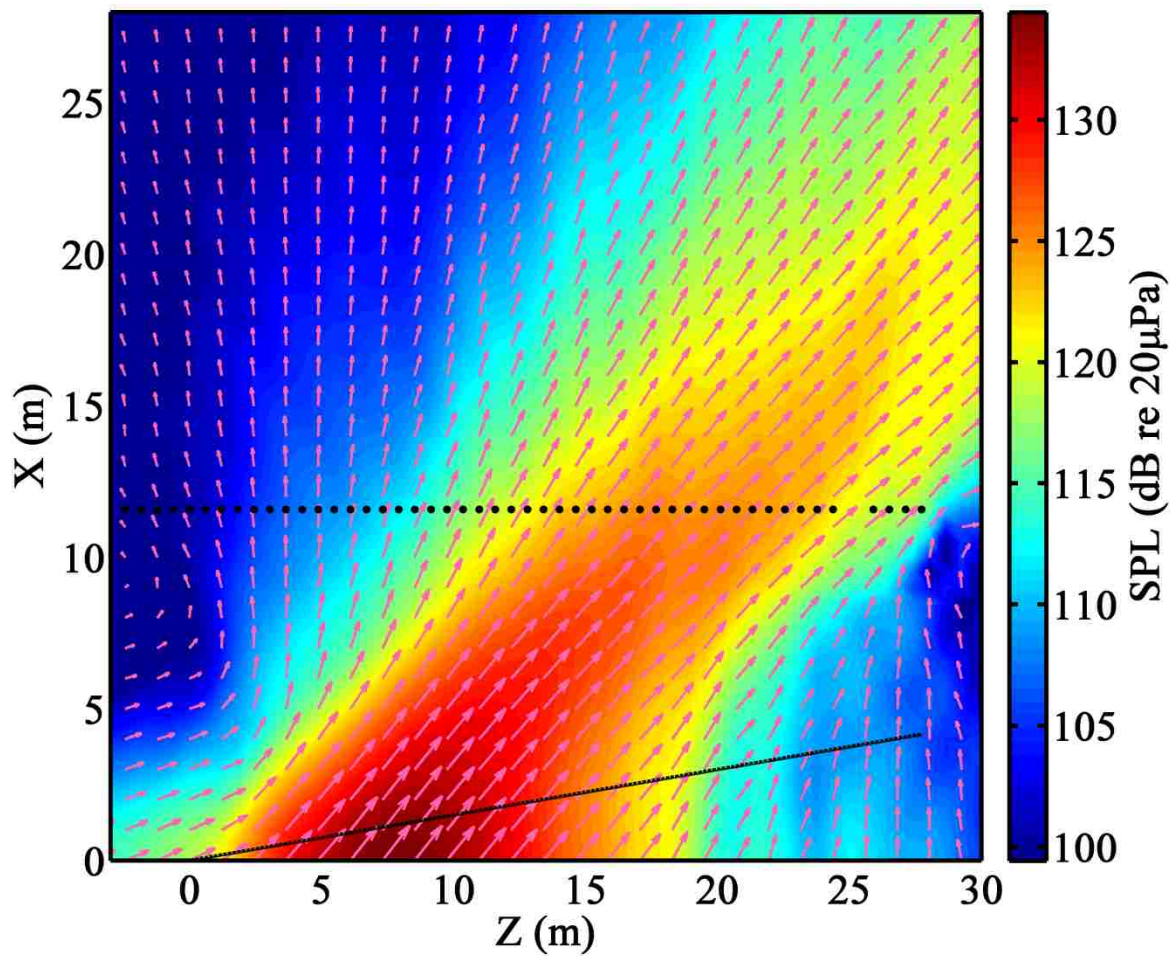
Figure 6.1 Acoustic intensity arrows overlaid on propagated pressure, both estimated through cylindrical NAH at 105 Hz, military engine condition.

Figure 6.2 shows cylindrical NAH intensity results at 210 Hz, military engine condition overlaid on propagated pressure. As before, the dots represent the ground microphone locations

and the dotted line the assumed shear layer location. Similar to Fig. 6.1, vorticity is seen near the microphone array edge, which is caused by the rapid reduction in amplitude from analytic continuation. Also seen are the two main lobe directions.
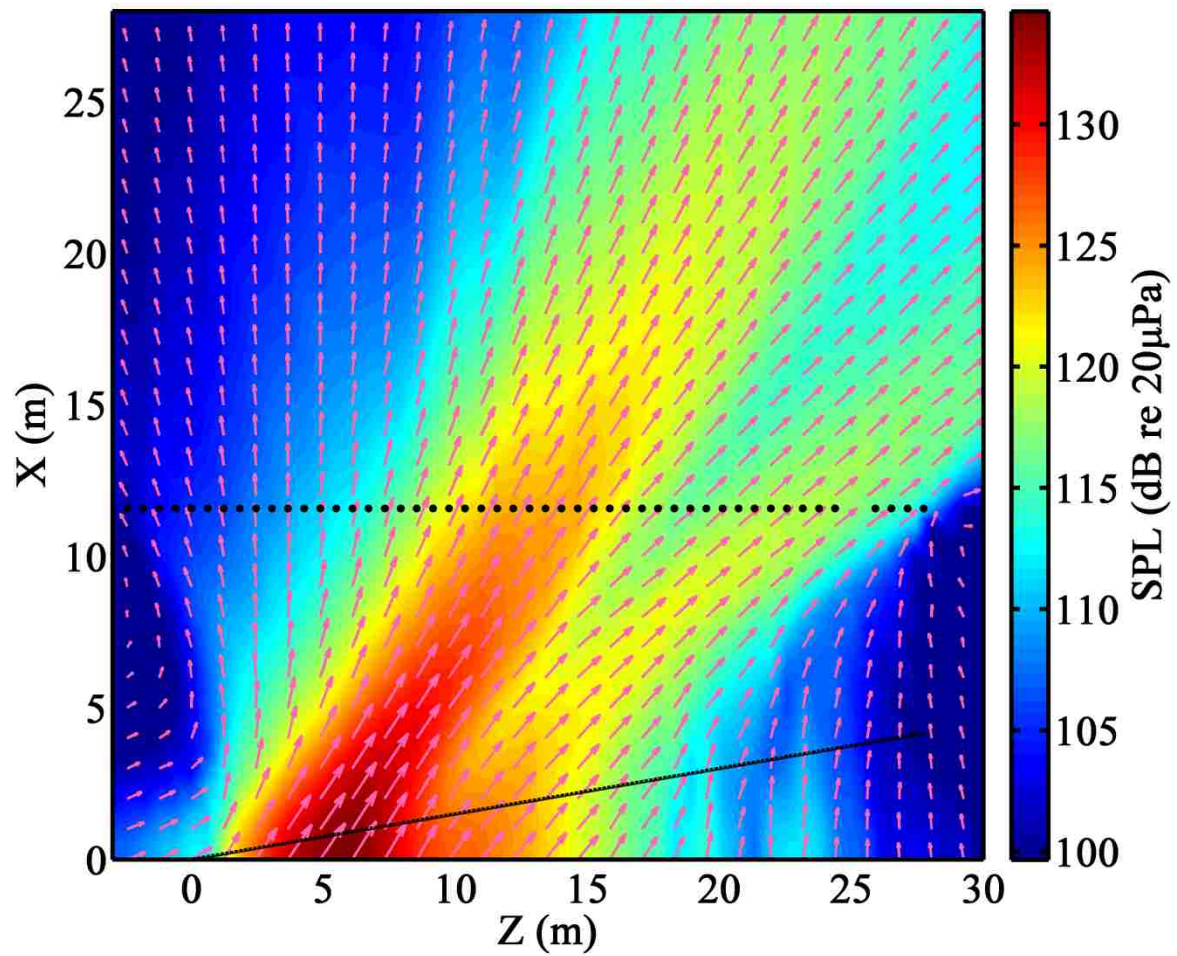
Figure 6.2 Acoustic intensity arrows overlaid on propagated pressure, both estimated through cylindrical NAH at 210 Hz, military engine condition.

6.2 Intensity from tetrahedral intensity probe

The tetrahedral intensity probe was attached to the top of the rig as shown in Fig. 2.1 and recorded data at every measurement scan except for those scans where the rig was turned vertically and are not considered in this thesis. The intensity probe uses the finite-difference method and Euler's equation, as discussed in Section 3.7.1, to calculate the acoustic intensity at the probe location. Some measurement planes have scans at different heights, when this occurred, the probe location closest to the ground was kept so as to be nearest to where the analysis for cylindrical and planar NAH occurred.

Figure 6.3 shows measured acoustic intensity vectors from the tetrahedral intensity probe at 105 Hz, military engine condition. The vector magnitudes are scaled on a logarithmic scale referenced to a small value so the argument of the logarithm is positive. The magnitudes can then be compared on a logarithmic basis where long vectors are much more prominent than short ones. The agreement between the various scans supports the processing technique and stationarity of the source at subsequent run-ups.
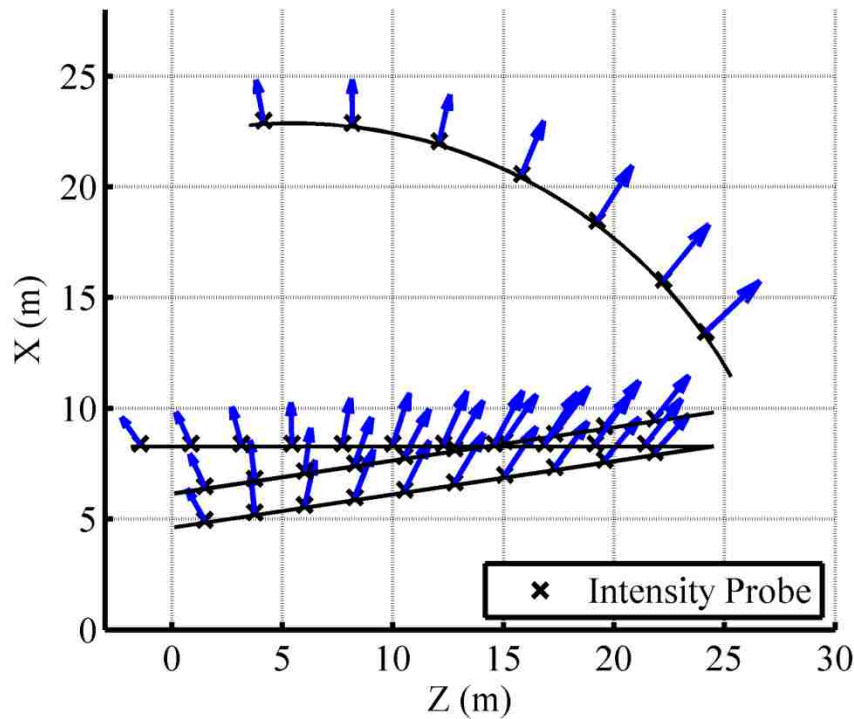
Figure 6.3 Benchmark intensities at 105 Hz, military engine condition, measured using tetrahedral intensity probe.

Figure 6.4 shows measured acoustic intensity vectors for 210 Hz, military engine condition. They are scaled in a manner similar those in Fig. 6.3. As evidenced along the 23-m arc, the dominant direction has moved upstream from where it was for 105 Hz, to about the middle of the arc. The most downstream vector on the arc is also slightly longer than the second-most downstream, showing the two-lobe directionality at 210 Hz. Also, the most downstream vectors are angled more downstream than their 105-Hz counterparts suggesting the source has moved upstream.
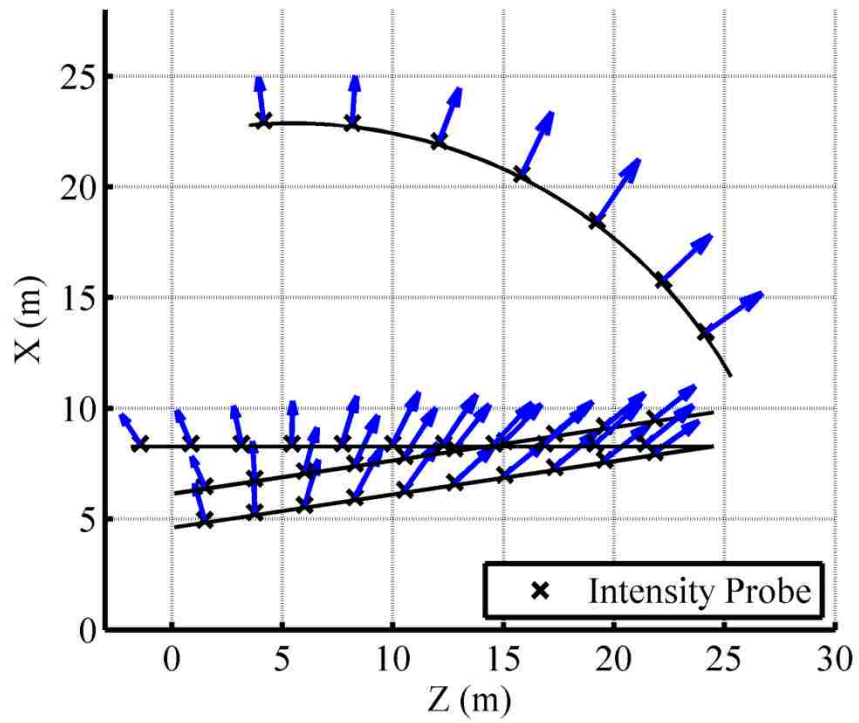
Figure 6.4 Benchmark intensities at 210 Hz, military engine condition, measured using tetrahedral intensity probe.

7 Conclusion

Acoustical measurements of a full-scaled military jet aircraft were performed, comprising over 6000 measurement points in the geometric near-field of the jet. To date, this data set constitutes the most dense of any yet performed on a full-scale military jet. A breadth of information can be mined from the data set. As an initial analysis, Fourier NAH was performed in both cylindrical and Cartesian coordinates. Theory for cylindrical and planar NAH was outlined. Multiple measurement scans along a measurement plane were pieced together using virtual coherence to obtain fully coherent partial fields. Certain obstacles related to ground reflections and the limited aperture size with respect to the source were considered. Ground reflections were addressed using knowledge of the sound field and mirroring about a ground reflecting plane. The limited aperture size and deleterious byproducts of the discrete Fourier transform such as wrap-around error and spatial aliasing were addressed using analytic continuation.

A simulated directional line array provided insight into the accuracy of the NAH processing. Initially we learn that if directional information is not recorded in the measurement, ie, the directional lobe does not pass through the measurement aperture, the processing does not well reconstruct that information. In other words, we can only reconstruct sources whose directionality passes through the measurement aperture.

Numerical simulations also show that along the estimated shear layer both cylindrical Fourier NAH and planar Fourier NAH yield similar the maximum source regions but they differ greatly in their maximum source level reconstructions. This can be attributed to the difference between the cylindrical and planar pressure propagators. When propagating towards the source, a

reconstructed pressure in cylindrical NAH can increase in amplitude independent of the existence of evanescent wave information. A reconstructed pressure in planar NAH, other other hand, depends on the evanescent wave information in order to increase in amplitude. Because measurements were taken in the acoustic far-field of the source, evanescent information was in the noise floor, fully regularized during the aperture extension, and did not yield an increase in pressure level upon planar NAH back-propagation.

This latter argument can additionally be applied to outward propagations along the 23-m arc. Outward propagations show an overestimation of the reconstructed soundfield, which is similarly attributable to the lack of evanescent information in the measurement.

Pressure reconstructions show an upstream movement of the maximum source region with increasing frequency. Also, the maximum source distribution narrows with increasing frequency. These two findings, in conjunction with the intensity results from both the cylindrical NAH processing and the tetrahedral intensity probe qualitatively agree with the popular two-source model for jet noise.[50]

Applying the NAH processes to actual measured data overestimation trends similar to numerical simulation results are seen along the 23-m arc. That then infers the important conclusion that the reconstructions along the shear layer behave as did the numerical simulation: the maximum source location is accurate, the maximum planar NAH pressure reconstruction is underestimated, and the maximum cylindrical NAH pressure reconstruction is accurate to within 2 dB.

Future work

One exciting area of future work is that of developing a numerical source model that when propagated outwards matches measured values and estimates from NAH. Initial qualitative work in this area was performed by Morgan[51]. Further refining the model through use of a least-squares or other optimization scheme could provide a quantitative best solution.

The success of cylindrical NAH, even though measuring in the acoustic far field, gives insight for new measurements. The frequency range and the linear measurement aperture could be greatly increased by putting the 90 microphones that were on the rig on the ground instead, either in a single, dense long line, or in multiple lines that could be compared to one another. The microphones on the rig also had a lower sensitivity than the linear ground array, allowing them to be placed closer to the rig. Results of this work could also assist in estimating how close to the jet the microphones could be placed.

One large assumption throughout this work was that of linearity and using linear theory. Acoustic pressure levels of interest in this work – on the order of 120-140 dB – are well above the linear acoustic realm. However, the complexity of nonlinear wave theory was beyond the scope of this attempt at military jet source localization. Incorporating nonlinear theory into the holography propagators could potentially better address the high noise levels.

The reconstructed and measured intensity of this work was qualitative in nature. Quantitative analysis will provide greater understanding of its performance. The usable frequency range of the tetrahedral intensity probe is much greater than that of cylindrical NAH. Higher frequencies could be investigated to see the directionality at higher frequencies. In addition, the tetrahedral probe was placed at different heights. Intensity maps in planes other than

the (Z,X) plane presented in this work could be investigated. Such analysis could address the axisymmetric assumption.

Lastly, the data set can be further mined to address sound propagation and its impact on communities surrounding the Air Force base. Work has been done on nonlinear wave propagation[52], and this data set can help increase understanding in that regard.

Appendix A One-step patch NAH

Through the course of this work an additional method for performing NAH that appeared computationally efficient was investigated. Developed by Lee and Bolton[29] and called one-step patch NAH, this method focuses on performing NAH over a limited area of the source, which results in finite, non-zero, amplitudes at the measurement edge. The process of extending the measured sound field and projecting the extended pressure onto the source surface occurs through use of a regularized least squares solution to the sampling and bandlimiting matrices.

What makes one-step patch NAH one step is that the measurement extension, bandlimiting, and propagation all happens in a single step. It does not require an iterative procedure to extend the measured region. The solution obtained, however, is not necessarily the best solution. It is one of many possible solutions because the cutoff wavenumber may or may not have been optimized. In their paper section III.B.3 the authors propose a method for choosing the optimized cutoff wave number. A method which ultimately is an iterative procedure.

While one-step patch NAH may have been computationally efficient given *a priori* knowledge of the necessary cutoff wavenumber, one-step patch NAH was abandoned for a couple reasons. First, extending the measured sound field prior to propagation was necessary for both this work and that performed by Wall, whose method did not have a built-in measurement extension, and whose results would be compared to those obtained by the author of this work. As such, an iterative procedure to extend the measured region would need to be performed, regardless of being built-in or not. Second, while it may be computationally expensive to iteratively extend the measurement aperture, once it has sufficiently converged, the extension can

be saved, never needing to be performed again; thus allowing a side-by-side comparison of the holography procedures.

Appendix B MATLAB code

FNAH_20120624

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DESCRIPTION: PERFORMS PLANAR FOURIER NAH WITH ANALYTIC CONTINUATION
%AUTHOR: DAVID W. KRUEGER/BYU 2012
%IN COLLABORATION WITH: ALAN WALL/BYU 2012
%SUBROUTINES:
%   read_pressures.m
%   read_benchmark_num_arc_pressures_20120118.m
%   coordinate_rotation_20110409.m
%   continued_geometry_20120214.m
%   read_benchmark_pressures_20120110.m
%   continuation_20120224.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program:
% Perfomrs planar Fourier near-field acoustical holography on actual and
% simulated jet noise data.
%
% It must be run from a directory or path containing the above listed
% subroutines.
%
%BIBLIOGRAPHY FOR PAPERS REFERENCED IN THE PROGRAM:
% E. G. Williams, 'Fourier Acoustics, Sound Radiation and Nearfield
% Acoustical Holography' (Academic Press, San Diego, 1999), pp. 115-180.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc; drawnow;

%Default figure parameters
set(0,'DefaultAxesFontName','Times New Roman');
set(0,'DefaultAxesFontSize',14);
set(0,'DefaultTextFontSize',14);
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',10);
set(0,'DefaultFigureUnits','inches');
set(0,'DefaultFigurePosition',[1 1 6 4]);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LOAD DATA
```

```matlab
%SUB
sub.processing.measure_extended = 0;%Choose 1 to measure the extended
                                    %aperture, 0 to measure limited aperture.
sub.numerical = 0;   %Choose 1 for a numerical source, 0 for real data.
sub.scan_based = 1; %Choose 1 for a scan-based numerical measurement.

%PARAMETER
parameter.meas_plane = 2;        %Always choose 2 for the
                                 % 'F22Holloman2009Geometry4.mat' geometry.
parameter.foi = 105;
parameter.eng_cond = 3;%Engine condition (1,2,3,4)=(idle,interm,mil,AB)
parameter.df = 2.9;%The band width of each narrow-band frequency value in
                   %the DFT. Choose from 1.5, 2.9, 5.9, 11.7Hz;

%DEFINE CONSTANTS
rho = 1.21;                      %Air desnity, kg/m^3.
c = 343;                         %Speed of sound, m/s;
k = 2*pi*parameter.foi/c;    %Wavenumber, m^-1;
ShearAngle = 8.53*(pi/180); %Measured from centerline

%READ LOCATIONS ALONG 75-ft ARC
cdtemp = cd;
cd('W:\NAH\CylindricalNAH');
list = list_generator;
file.complex_pres = ...
    'Z:\Students\Alan Wall\HollomanF22_2009\ComplexPressures_v';
load('Geometry\F22Holloman2009Geometry3.mat');
[Zarc,Xarc,Yarc,parc] = ...
    read_pressures(4,parameter,list,sub,file,F22Holloman2009Geometry);
cd(cdtemp);
if sub.numerical
    clear parc
    [parc] = read_benchmark_num_arc_pressures_20120118(parameter,Zarc,...
        Xarc,Yarc);
end
%Rotate arc geometries to get propagation distances
[R.arc,S.arc,T.arc] = ...
    coordinate_rotation_20110409(Xarc,Yarc,Zarc,ShearAngle);

%DEFINE RECONSTRUCTION DISTANCE
%AND START PROPAGATION DISTANCE FOR LOOP
load('F22Holloman_PropDist');
R_meas = X_PLN(parameter.meas_plane); %Meas. distance from shear(rot.geom)
% D = R.arc(1,:)-R_meas;%For locations along arc
D = [-R_meas,-4.5,-3.5,-2.5,X_PLN(1)-X_PLN(2),0,...
    X_PLN(22:30),X_PLN(30)+1.5071*(1:3)];
endt = numel(D);%
for t = 1:endt
d = D(t);                        %m propagation distance
disp(['d = ',num2str(d)]);
R_rcs = R_meas+d;                %Rcs. distance from shear

if t==1
%LIST
list = list_generator_20110628();
```

```matlab
%FILE
file.geometry = 'F22Holloman2009Geometry6.mat'; %Geometry file name
file.complex_pres_path = ...
 'Z:\Students\Alan Wall\HollomanF22_2009\Press_TEST\ProcessedPressures_v6';
                               %Complex pressure files location
% file.read = 'read_pressures_20110908';% data reader subroutine name
file.read = 'read_pressures_20110901';% data reader subroutine name
file.virt_coh = 'virtual_coherence_20110908';% virt. coh. subroutine name

%GRID RESOLUTION
% Chooses every nnth column and row to propagate; Ensures that there
% are at least 4 array microphones per wavelength
nn = max([floor(c/parameter.foi/4/(0.5*0.3048)) 1]);
dT = 0.5*0.3048;            %Microphone spacing (m)

%AC
ac.ad_hoc = 0.2; % Ad hoc constant for determining stringency of Iteration
                 % stopping condition. (See Ref. 8, Sec. III)
ac.iter_lim = 25;% Limit on number of iterations.  We have found that for
                  % this particualr  problem, the convergence described in
                  % Ref. 8, Sec. IIImay not be optimal.  For this case, a
                  % limit of 25 iterations is often sufficient.
ac.continue_length_z.up = 30;%3;%    % m Positive z direction (downstream)
ac.continue_length_z.dn = 30;%3;%    % m Negative z direction (upstream)
ac.continue_length_y.up = 20;%2;%    % m Positive y direction
ac.continue_length_y.dn = 20;%2;%    % m Negative y direction

%CONTINUED APERTURE SIZE PARAMETERS
ac.continue_points_y.up = ceil(ac.continue_length_y.up/dT);
ac.continue_points_y.dn = ceil(ac.continue_length_y.dn/dT);
ac.continue_points_z.up = ceil(ac.continue_length_z.up/dT);
ac.continue_points_z.dn = ceil(ac.continue_length_z.dn/dT);
ac.continue_points_reduced_y.up = ceil(ac.continue_points_y.up/nn);
ac.continue_points_reduced_y.dn = ceil(ac.continue_points_y.dn/nn);
ac.continue_points_reduced_z.up = ceil(ac.continue_points_z.up/nn);
ac.continue_points_reduced_z.dn = ceil(ac.continue_points_z.dn/nn);

%NUMBER OF ARRAY MICS
scan_size = ...
    list.plane.meas_h(parameter.meas_plane)*...
    list.plane.meas_v(parameter.meas_plane);

%REFERENCE MICROPHONES
ref_array = 1:52;
ref_num = length(ref_array);     %Number of reference microphones

%LOAD GEOMETRY FILE
load(file.geometry);

%COHERENCE CRITERION
parameter.coh_crit = 0.9;    %Choose 0.5, 0.6, 0.7, 0.8, 0.9, or 0.99.
parameter.coh_v = find(parameter.coh_crit == list.coh_crit); %Version

%READ MEASUREMENT PRESSURES AND GEOMETRY
tic
```

```matlab
disp('STARTING LOAD DATA');
[data.measured] = ...
    eval([file.read,'(parameter.meas_plane,parameter,list,'...
    'sub,file,F22Holloman2009Geometry,ac,nn)']);
disp('FINISHED LOADING DATA');
toc

%READ ORIGINAL MEASUREMENT PLANE CONTINUED GEOMETRY
X.meas = data.measured.X;
Y.meas = data.measured.Y;
Z.meas = data.measured.Z;
[X.cont,Y.cont,Z.cont] = ...
        continued_geometry_20120214(X.meas,Y.meas,Z.meas,ac,nn);

%ROTATED GEOMETRIES
[R.cont,S.cont,T.cont] = ...
    coordinate_rotation_20110409(X.cont,Y.cont,Z.cont,ShearAngle);
[R.cont_in,S.cont_in,T.cont_in] = ...
    coordinate_rotation_20110409(X.meas(:,1:nn:end),...
    Y.meas(:,1:nn:end),Z.meas(:,1:nn:end),ShearAngle);
end %end of first t==1

%Reconstructed Geometry
%%%Adds rcs dist and rotates back. Notice the +d and -ShearAngle.
[X.rcs(:,:,t),Y.rcs(:,:,t),Z.rcs(:,:,t)] = ...
    coordinate_rotation_20110409((R.cont+d),...
    S.cont,T.cont,(-ShearAngle));


%BENCHMARK PRESSURE
if sub.numerical
[benchmark] = read_benchmark_pressures_20120110(parameter,Z,X,Y,t);
end

if t==1 %start of second t==1
%END LOAD DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VIRTUAL COHERENCE

disp('Performing virtual coherence.');

vc.measured = eval([file.virt_coh,'(data.measured,parameter,'...
    'sub,list,ref_num,scan_size);']);

disp('Virtual coherence completed.')

%END VIRTURAL COHERENCE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%APERTURE EXTENSION

if ~sub.processing.measure_extended
%      if t==1
        disp('Performing aperture extension.');
```

```matlab
        %EXTEND APERTURE
        %The resulting geometry variables 'Zc' and 'Yc' are also part of
        %the rotated axes geometry.

        tic
        for i = 1:vc.measured.pf_num
            [ppfc(:,:,i),bb(i),Alpha(i),Lambda(:,:,i),Falpha1(:,:,1),...
                Fz,Fzinv,Fy,Fyinv,KZ,KY,KX] = ...
                continuation_20120224(vc.measured.p_pf(:,1:nn:end,i),...
                T.cont_in,S.cont_in,parameter.foi,...
                R_meas,R_rcs,file,i,ac,sub,nn);
        end
        toc

        clear i

        disp('Aperture extension completed.');

        %CALCULATE CONTINUED PARTIAL FIELD SUM
        ppfc_total = 0;
        for i = 1:vc.measured.pf_num
            ppfc_total = ppfc_total+abs(ppfc(:,:,i)).^2;
        end
        ppfc_total = sqrt(ppfc_total);
%     end
end %end of aperture extension
disp('Finished aperture extension');
end %end of second t==1
%END APERTURE EXTENSION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%FNAH
disp('Performing FNAH.');

%Propagator
GDinv = exp(-1i*KX*d);  % Pressure propagator
Guz = KZ/(rho*c*k).*exp(-1i*KX*d);% Z velocity propagatorWilliamsbook(2.60)
Guy = KY/(rho*c*k).*exp(-1i*KX*d);% Y velocity propagator
Gux = KX/(rho*c*k).*exp(-1i*KX*d);% X velocity propagator

%FourierTransformEachPf
for pp = 1:numel(ppfc(1,1,:));
    PPFC(:,:,pp) = (Fz*(Fy*ppfc(:,:,pp)).').';
end

%MultiplyEachPfByPropagator
for pp = 1:numel(ppfc(1,1,:));
    PPFR(:,:,pp) = (Falpha1.*PPFC(:,:,pp)).*GDinv;
end

%InverseTransformEachPf
for pp = 1:numel(ppfc(1,1,:));
    ppfr(:,:,pp) = Fyinv*(Fzinv*PPFR(:,:,pp)).').';
end
```

```matlab
%Sum reconstructed partial fields
pr = squeeze(sqrt(sum(abs(ppfr).^2,3)));    %Units: |Pa|

%Collect summed reconstructions for plotting
p_rcs_all(:,:,:,t)= ppfr;
p_rcs(:,:,t) = pr;
if sub.numerical
p_bench_all(:,:,t) = benchmark.p_bench;
end
disp('FNAH Completed.');


end
%END FNAH
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% PLOT RESULTS
clear rcs_up rcs_dwn rcs_top rcs_bot rcs_y3in;
Z.rcsplot = []; X.rcsplot = []; Y.rcsplot = []; p_rcsplot = [];
for t=1:endt
rcs_up(t) = find(Z.rcs(1,:,t)>0,1,'first');%ind of first Z.rcs column > 0
rcs_dwn(t) = find(Z.rcs(1,:,t)>26,1,'first');
rcs_top(t) = find(Y.rcs(:,1,t)>Y.meas(1,1)+.01,1,'last');%1strow above meas
rcs_bot(t) = find(Y.rcs(:,1,t)>0,1,'last');%ind of first row above ground
rcs_y3in(t) = find(Y.rcs(:,1,t)>0,1,'last');%which row is 3in off ground
NZplot = numel(rcs_up(1):rcs_dwn(1));


Z.rcsplot(:,:,t) = Z.rcs(rcs_top:rcs_bot,rcs_up:rcs_dwn,t);
X.rcsplot(:,:,t) = X.rcs(rcs_top:rcs_bot,rcs_up:rcs_dwn,t);
Y.rcsplot(:,:,t) = Y.rcs(rcs_top:rcs_bot,rcs_up:rcs_dwn,t);


p_rcsplot(:,:,t) = p_rcs(rcs_top(t):rcs_bot(t),...
    rcs_up(t):rcs_up(t)+NZplot-1,t);
if sub.numerical
bench_all(:,:,t) = p_bench_all(rcs_top(t):rcs_bot(t),...
    rcs_up(t):rcs_up(t)+NZplot-1,t);
end
end


ttt=1;
ppfcplot = 20*log10(abs(ppfc_total(rcs_top(ttt):rcs_bot(ttt),...
    rcs_up(ttt):rcs_up(ttt)+NZplot-1))/2e-5);
Z.contplot = Z.cont(rcs_top(ttt):rcs_bot(ttt),...
    rcs_up(ttt):rcs_up(ttt)+NZplot-1);
X.contplot = X.cont(rcs_top(ttt):rcs_bot(ttt),...
    rcs_up(ttt):rcs_up(ttt)+NZplot-1);
Y.contplot = Y.cont(rcs_top(ttt):rcs_bot(ttt),...
    rcs_up(ttt):rcs_up(ttt)+NZplot-1);


CAXIS = ([max(max(max(20*log10(abs(p_rcs)/2e-5))))-30,...
    max(max(max(20*log10(abs(p_rcs)/2e-5))))]);
if sub.numerical
CAXISB = [max(max(max(20*log10(abs(bench_all(:,:,1))/2e-5))))-30,...
    max(max(max(20*log10(abs(bench_all(:,:,1))/2e-5))))];
end
```

```matlab
% %Plot measured pressure
% figure;
% surf(Z.meas,X.meas,Y.meas,20*log10(abs(sum(data.measured.p,3))/2e-5));
% shading interp
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('measured');
% view([-7 24]); daspect([1,1,1]);
% colorbar;


% %Plot entire continued and reconstructed pressure
% figure;
% % ttt=1;
% subplot 121
% surf(Z.cont,X.cont,Y.cont,20*log10(abs(ppfc_total)/2e-5));
% hold on;
% surf(Z.rcs(:,:,ttt),X.rcs(:,:,ttt),Y.rcs(:,:,ttt),...
%     20*log10(abs(p_rcs(:,:,ttt))/2e-5));
% shading interp;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('reconst''d');
% view([-7 24]); daspect([1,1,1]);
% colorbar;
% % caxis(CAXIS);
% axis tight;
% subplot 122
% surf(Z.cont,X.cont,Y.cont,20*log10(abs(ppfc_total)/2e-5));
% hold on;
% surf(Z.rcs(:,:,ttt),X.rcs(:,:,ttt),Y.rcs(:,:,ttt),...
%     20*log10(abs(p_rcs(:,:,ttt))/2e-5));
% shading interp;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('continued');
% view([-143 8]); daspect([1,1,1]);
% colorbar;
% % caxis(CAXIS);
% axis tight;


% %Plot truncated continued and reconstructed pressure
% figure;
% ttt = 1;
% subplot 211;
% surf(Z.contplot,X.contplot,Y.contplot,ppfcplot);
% shading interp; colorbar;
% view([-7 24]); daspect([1,1,1]);
% xlim([-1 27]); ylim([0 10]);
% caxis=([max(max(ppfcplot))-30 max(max(ppfcplot))]);
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title(['f=',num2str(parameter.foi),...
%     ', continued, caxis= ',num2str(caxis,4)]);
% subplot 212;
% surf(Z.rcsplot(:,:,ttt),X.rcsplot(:,:,ttt),Y.rcsplot(:,:,ttt),...
%     20*log10(abs(p_rcsplot(:,:,ttt))/2e-5));
% shading interp; colorbar;
% view([-7 24]); daspect([1,1,1]);
% xlim([-1 27]); ylim([0 10]);
```

```matlab
% caxis=([max(max(20*log10(abs(p_rcsplot(:,:,ttt))/2e-5)))-30,...
%     max(max(20*log10(abs(p_rcsplot(:,:,ttt))/2e-5)))]);
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title(['f=',num2str(parameter.foi),...
%     ', reconstructed, caxis= ',num2str(caxis,4)]);

% %Plot entire reconstructed pressure
% figure;
% surf(Z.rcs,X.rcs,Y.rcs,20*log10(abs(ppfr)/2e-5));
% shading interp; colorbar;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('reconst''d');
% view([-7 24]); daspect([1,1,1]);
% caxis(CAXIS);
% axis tight;

% %Plot entire benchmark pressure
% figure;
% surf(Z.rcs,X.rcs,Y.rcs,20*log10(abs(benchmark.p_bench)/2e-5));
% shading interp; colorbar;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('benchmark');
% view([-7 24]); daspect([1,1,1]);
% % caxis(CAXISB);
% caxis(CAXIS);
% axis tight;

% %Plot NUMERICAL truncated reconstruction and benchmark
% if sub.numerical;
% figure;
% subplot 211;
% for t=1:endt
% surf(Z.rcsplot(:,:,t),X.rcsplot(:,:,t),Y.rcsplot(:,:,t),...
%     20*log10(abs(p_rcsplot(:,:,t))/2e-5));
% hold on;
% shading interp; cb=colorbar;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('From numerical data, reconstruction along shear');
% view([0 0]);%view([-7 24]);
% daspect([1,1,.5]);
% caxis(CAXISB);
% axis tight;
% ylabel(cb,'SPL dBre20\muPa');
% end
% subplot 212;
% for t=1:endt
% surf(Z.rcsplot(:,:,t),X.rcsplot(:,:,t),Y.rcsplot(:,:,t),...
%     20*log10(abs(bench_all(:,:,t))/2e-5));
% hold on;
% shading interp; cb=colorbar;
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% title('From numerical data, benchmark along shear');
% view([0 0]); %view([-7 24]);
% daspect([1,1,.5]);
% caxis(CAXISB);
% axis tight;
```

```matlab
% ylabel(cb,'SPL dBre20\muPa');
% end
% end

% %Plot ACTUAL DATA truncated reconstructions
% if ~sub.numerical
% figure;
% for t=1:endt
% surf(Z.rcsplot(:,:,t),X.rcsplot(:,:,t),Y.rcsplot(:,:,t),...
%     20*log10(abs(p_rcsplot(:,:,t))/2e-5));
% hold on;
% end
% shading interp;
% colorbar('location','east','yaxislocation','Right');
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% % title('From actual data, reconst''d');
% view([-5 36]); daspect([1,1,1]);
% caxis(CAXIS);
% axis tight;
% %Plot the 75-ft arc
% hold on;
% surf(Zarc,Xarc,Yarc,20*log10(mean(abs(parc),3)/2e-5));
% caxis(CAXIS);
% shading interp;
% % axis([-10 55 0 33 0 5]);
% axis tight;
% %Plot lines around the measured aperture
% plot3([Z.meas(1,1) Z.meas(1,end)],[X.meas(1,1) X.meas(1,end)],...
%     [Y.meas(1,1) Y.meas(1,end)],'k-');
% plot3([Z.meas(end,1) Z.meas(end,end)],[X.meas(end,1) X.meas(end,end)],...
%     [Y.meas(end,1) Y.meas(end,end)],'k-');
% plot3([Z.meas(1,1) Z.meas(end,1)],[X.meas(1,1) X.meas(end,1)],...
%     [Y.meas(1,1) Y.meas(end,1)],'k-');
% plot3([Z.meas(1,end) Z.meas(end,end)],[X.meas(1,end) X.meas(end,end)],...
%     [Y.meas(1,end) Y.meas(end,end)],'k-');
% end

% %Plot line plots recont and bench 3in above ground
% % figure;
% for tt = 1;%:endt;
% figure;
% plot(Z.cont(rcs_y3in(tt),:),...
%     20*log10(abs(ppfc_total(rcs_y3in(tt),:))/2e-5),'b');
% hold on;
% plot(Z.rcs(rcs_y3in(tt),:,tt),...
%     20*log10(abs(p_rcs(rcs_y3in(tt),:,tt))/2e-5),'r');
% if sub.numerical
% plot(Z.rcs(rcs_y3in(tt),:,tt),...
%     20*log10(abs(p_bench_all(rcs_y3in(tt),:,tt))/2e-5),'k');
% end
% xlabel('Z (m)'); ylabel('SPL (dB re20\muPa)');
% grid on;
% if ~sub.numerical && tt==1
% title(['f=',num2str(parameter.foi),...
%     ', from actual data, 3 in off ground along shear, d=',...
%     num2str(D(tt))]);
```

```matlab
% legend('continuation (info only)','reconstruction',...
%     'location','southwest');
% elseif ~sub.numerical
% title(['f=',num2str(parameter.foi),...
%     ', actual data, 3 in off ground, d=',num2str(D(tt))]);
% legend('continuation (info only)','reconstruction',...
%     'location','southeast');
% elseif sub.numerical && tt==1
% title(['f=',num2str(parameter.foi),...
%     ', num. data 3 in off ground along shear, d=',num2str(D(tt))]);
% legend('continuation (info only)','reconstruction','benchmark',...
%     'location','northeast');
% elseif sub.numerical
% title(['f=',num2str(parameter.foi),...
%     ', num. data 3 in off ground, d=',num2str(D(tt))]);
% legend('continuation (info only)','reconstruction','benchmark',...
%     'location','southeast');
% end
% % axis([-40 60 70 150]);
% if sub.numerical
% axis([0 30.5 105 155]);
% else
% axis([0 30.5 85 135]);
% end
% end


% %Plot diff. between recont and bench along shear
% for tt= 1;
% figure;
% set(gcf,'OuterPosition',[1 1 6 3]);
% plot(Z.rcs(rcs_y3in(tt),:,tt),...
%     20*log10(abs(p_rcs(rcs_y3in(tt),:,tt))/2e-5)-...
%     20*log10(abs(p_bench_all(rcs_y3in(tt),:,tt))/2e-5));
% axis([0 30.5 -15 15]);
% ylabel('dB');
% xlabel('Z (m)');
% grid on;
% set(gca,'YMinorGrid','on');
% set(gca,'GridLineStyle','-');
% end



% %Plot reconstructions 3in off ground (similar to cylNAH plots)
% figure;
% pcolor(squeeze(Z.rcs(rcs_y3in(1),:,:)),...
%     squeeze(X.rcs(rcs_y3in(1),:,:)),...
%     squeeze(20*log10(abs(p_rcs(rcs_y3in(1),:,:))/2e-5)));
% daspect([1,1,1]);
% xlabel('Z (m)'); ylabel('X (m)');
% colorbar; grid on;
% Caxis=caxis;
% caxis([Caxis(2)-30,Caxis(2)]);
% title('Reconstructions 3in off ground');


% %Plot line plots along bottom of arc(having used D from arc to get p_rcs)
```

```
% yarcind = find(Y.cont(:,1)>1.6,1,'last');
% ZrecArc(:,:) = Z.rcs(yarcind,:,:); %Pick off bottom col along arc
% XrecArc(:,:) = X.rcs(yarcind,:,:);%
% precArc(:,:) = p_rcs(yarcind,:,:);
% ZrecArcCol = reshape(ZrecArc.',[],1);%TriScatteredInterp req's col unputs
% XrecArcCol = reshape(XrecArc.',[],1);
% precArcCol = reshape(precArc.',[],1);
% [ZZarc,XXarc] = meshgrid(Zarc(1,:),Xarc(1,:));
% F = TriScatteredInterp(ZrecArcCol,XrecArcCol,precArcCol,'natural');
% qz = F(ZZarc,XXarc);
% %Plot reconstructions at bottom row of arc before interpolation
% figure;
% pcolor(ZrecArc,XrecArc,20*log10(abs(precArc)/2e-5));
% % shading interp;
% colorbar;
% % caxis = CAXIS;
% daspect([1,1,1]); axis([4 26 11 23]);
% xlabel('Z (m)'); ylabel('X (m)');
% title('Before interpolation (at height of bottom of arc)');
% %Plot interpolation
% figure;
% pcolor(ZZarc,XXarc,20*log10(abs(qz)/2e-5));
% colorbar;
% % caxis = CAXIS;
% daspect([1,1,1]);
% xlabel('Z (m)'); ylabel('X (m)');
% title('After interpolation (at height of bottom of arc)');
% %Plot arc for comparison
% figure;
% surf(Zarc,Xarc,Yarc,20*log10(mean(abs(parc),3)/2e-5));
% shading interp; colorbar; daspect([1,1,1]); axis tight;
% % caxis = CAXIS;
% %Plot line plots at bottom row of arc
% figure;
% subplot 211;
% plot(Zarc(1,:),20*log10(abs(diag(qz))/2e-5),'r');
% hold on;
% plot(Zarc(1,:),20*log10(mean(abs(parc(end,:,:)),3)/2e-5),'k');
% axis tight; grid on;
% xlabel('Z (m)'); ylabel('dB re 20\muPa');
% % if sub.numerical
% % title(['f=',num2str(parameter.foi),', bottom of arc, numerical data']);
% % elseif ~sub.numerical
% % title(['f=',num2str(parameter.foi),', bottom of arc, actual data']);
% % end
% legend('Planar NAH','Benchmark','location','southeast');
% %Plot difference between benchmark and recontruction
% % figure;
% % set(gcf,'OuterPosition',[1 1 6 3]);
% subplot 212;
% plot(Zarc(1,:),20*log10(abs(diag(qz))/2e-5).' -...
%     20*log10(mean(abs(parc(end,:,:)),3)/2e-5));
% grid on; axis([Zarc(1,1) Zarc(1,end) -5 5]);%tight;
% xlabel('Z (m)');
% ylabel('dB');
% set(gca,'YMinorGrid','on');
% % title('difference between recont and bnch along bottom of arc');
```

```matlab
% set(gcf,'paperpositionmode','auto');
% print('-dtiff','-r400','PlanarNAH_210Hz_RealDATA_planes_20120624');


% %%%To plot the jet model
% cdtemp = cd;
% cd(['Z:\Students\Alan Wall\HollomanF22_2009\'...
%     'model3d_for_BYU\model3d_for_BYU']);
% plot_3ds_model_empty;%(h_fig,'MeasSchematic_20110429.fig');
% cd(cdtemp);
% % axis([-10 55 -2 35 0 5]);



% %%%plot reconsruction and benchmark at PLANE 1
% if ~sub.numerical
% figure;
% cdtemp = cd;
% cd('W:\NAH\CylindricalNAH');
% list = list_generator;
% file.complex_pres = ...
%     'Z:\Students\Alan Wall\HollomanF22_2009\ComplexPressures_v';
% load('Geometry\F22Holloman2009Geometry3.mat');
% [Zplane1,Xplane1,Yplane1,pplane1] = ...
%     read_pressures(1,parameter,list,sub,file,F22Holloman2009Geometry);
% cd(cdtemp);
% %Interpolate over bad meas mics
% BAD = [5,20,35,50,65,80,95,110,125,140];%Cols of row 2 with bad data
% PP = mean(abs(pplane1),3);
% PP(2,BAD) = sqrt((PP(1,BAD-1).^2+PP(3,BAD+1).^2+...
%     PP(3,BAD-1).^2+PP(1,BAD+1).^2)/4);
% %Plot Benchmark plane 1
% subplot 211;
% surf(Zplane1,Xplane1,Yplane1,20*log10(PP/2e-5));
% shading interp;
% CBp1 = colorbar;
% ylabel(CBp1,'SPL (dB re 20\muPa)');
% caxis(CAXIS);
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% view([0 0]);
% xlim([Zplane1(1,1) Zplane1(1,end)]);
% zlim([Yplane1(end,1) Yplane1(1,1)]);
% %Find reconstruction indices corresponding to plane 1 geometries
% Dind = 5;
% %index of 1st point below meas plane
% bot_pl1 = find(Y.rcs(:,1,Dind)<Yplane1(end,1),1,'first');
% %index of 1st pt above meas plane
% top_pl1 = find(Y.rcs(:,1,Dind)>Yplane1(1,1),1,'last');
% %index of 1st point just upstream of meas plane
% up_pl1 = find(Z.rcs(1,:,Dind)<Zplane1(1,1),1,'last');
% %index of 1st point just downstream of meas plane
% down_pl1 = find(Z.rcs(1,:,Dind)>Zplane1(1,end),1,'first');
% %Plot reconstruction at plane 1 from plane 2
% subplot 212;
% %PLOT RECONSTRUCTED FIELD
% surf(Z.rcs(top_pl1:bot_pl1,up_pl1:down_pl1,Dind),...
%     X.rcs(top_pl1:bot_pl1,up_pl1:down_pl1,Dind),...
```

```
%      Y.rcs(top_pl1:bot_pl1,up_pl1:down_pl1,Dind),...
%      20*log10(abs(p_rcs(top_pl1:bot_pl1,up_pl1:down_pl1,Dind))/2e-5));
% shading interp;
% CBp1rcs = colorbar;
% ylabel(CBp1rcs,'SPL (dB re 20\muPa)');
% caxis(CAXIS);
% xlabel('Z (m)'); ylabel('X (m)'); zlabel('Y (m)');
% view([0 0]);
% xlim([Zplane1(1,1) Zplane1(1,end)]);
% zlim([Yplane1(end,1) Yplane1(1,1)]);
% end
```

read_pressures

The "read_pressures.m" function is used in both planar and cylindrical NAH. In planar NAH, its purpose is to call the measurement locations and load measured data along the arc and plane 1. Simulated measurements along the arc for planar NAH are generated in "read_benchmark_num_arc_pressures_20120118.m". In cylindrical NAH, it calls both the arc locations and data for both measured and simulated data..

```
% FUNCTION [Z,X,Y,p,Pref] =
%     read_pressures(plane,parameter,list,sub,file,F22Holloman2009Geometry,
%     geometry)
%
% Description:
%
%  This function generates the
%  measured pressure map and geomery
%  for a specified plane.
%
% Inputs:
%
%   plane       :    the number of the plane of interest
%   parameter   :    the variable contining all the parameters for this case
%   list        :    the variable containing possible parameter lists, used
%                    for indexing purposes
%   sub         :    all subroutine parameter (boolean)
%   file        :    subroutine and other file names
%   F22Holloman2009Geometry :   measurement plane geometry variable
%
% Outputs:
%
%   Z       :    data points in Z
%   X       :    data points in X
%   Y       :    data points in Y
%   p       :    measured pressures
%   Pref    :    measured pressures at reference microphones; include in
%                output only if plane < 3.
%
```

```matlab
% Author: Alan Wall (alantwall@gmail.com)
%
% Date: 09/13/2010
%

function [Z,X,Y,p,Pref] = ...
    read_pressures(plane,parameter,list,sub,file,F22Holloman2009Geometry)

%FIND RUNS OF GIVEN PLANE
runsOI = eval(['list.runs.plane_',int2str(plane)]);

indy = 1;
for r = 1:length(runsOI)

    if ~sub.numerical

        %SELECT APPROPRIATE COMPLEX PRESSURE VERSION
        if runsOI(r) == 1 || parameter.eng_cond == 4
            blocks = ...
                list.cp_v.blocks2(find(list.cp_v.df == parameter.df)-1);
            cpv = list.cp_v.version(find(list.cp_v.df == parameter.df)-1);
        else
            blocks = list.cp_v.blocks(list.cp_v.df == parameter.df);
            cpv = list.cp_v.version(list.cp_v.df == parameter.df);
        end

        %READ COMPLEX PRESSURE FILE
        filename = ['CPv',int2str(cpv),'_',...
            num2str(parameter.foi,'%04i'),'Hz_Run',...
            num2str(runsOI(r)),'_',list.eng_cond{parameter.eng_cond},...
            '_',num2str(blocks,'%03i'),'blocks.mat'];
        eval(['load(''',file.complex_pres,int2str(cpv),'\',filename,''')'])

        %REFERNCE PRESSURES
        if plane < 4
        Pref(:,:,:,r) = ...
            permute(CPrefs(:,:,1:list.cp_v.blocks(...
            list.cp_v.df == parameter.df)),[3 2 1]);
        end
    end

        indz = 1;
        for h = 1:list.plane.scans_h(plane)

            %FIND RUN NUMBER FOR GIVEN HEIGHT
            rind = ...
                eval(['find(runsOI(r) == list.runs_order.plane_',...
                int2str(plane),')']);

            %GENERATE GEOMETRY
            Xtemp = F22Holloman2009Geometry{plane}(:,:,rind,h,1);
            Ytemp = F22Holloman2009Geometry{plane}(:,:,rind,h,2);
            Ztemp = F22Holloman2009Geometry{plane}(:,:,rind,h,3);

            if h == list.plane.scans_h(plane) && r == length(runsOI)
```

```matlab
        X(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = Xtemp;
        Y(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = Ytemp;
        Z(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = Ztemp;
    elseif h == list.plane.scans_h(plane)
        X(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = ...
            Xtemp(1:list.plane.stepsize_v(plane),:);
        Y(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = ...
            Ytemp(1:list.plane.stepsize_v(plane),:);
        Z(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.meas_h(plane)-1)) = ...
            Ztemp(1:list.plane.stepsize_v(plane),:);
    elseif r == length(runsOI)
        X(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Xtemp(:,1:list.plane.stepsize_h(plane));
        Y(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Ytemp(:,1:list.plane.stepsize_h(plane));
        Z(indy:indy+(list.plane.meas_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Ztemp(:,1:list.plane.stepsize_h(plane));
    else
        X(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Xtemp(1:list.plane.stepsize_v(plane),...
            1:list.plane.stepsize_h(plane));
        Y(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Ytemp(1:list.plane.stepsize_v(plane),...
            1:list.plane.stepsize_h(plane));
        Z(indy:indy+(list.plane.stepsize_v(plane)-1),...
            indz:indz+(list.plane.stepsize_h(plane)-1)) = ...
            Ztemp(1:list.plane.stepsize_v(plane),...
            1:list.plane.stepsize_h(plane));
    end
    clear Xtemp Ytemp Ztemp

    %GENERATE PRESSURE MAP
    for blk = 1:list.cp_v.blocks(list.cp_v.df == parameter.df)
        if ~sub.numerical
            phsqtemp = reshape(CPrig(:,h,blk),...
                [list.plane.meas_v(plane) ...
                list.plane.meas_h(plane)]);

            if h == list.plane.scans_h(plane) && ...
                    r == length(runsOI)
                p(indy:indy+(list.plane.meas_v(plane)-1),...
                    indz:...
                    indz+(list.plane.meas_h(plane)-1),blk) = ...
                    phsqtemp;
            elseif h == list.plane.scans_h(plane)
```

```matlab
                        p(indy:indy+(list.plane.stepsize_v(plane)-1),...
                            indz:...
                            indz+(list.plane.meas_h(plane)-1),blk) = ...
                            phsqtemp(1:list.plane.stepsize_v(plane),:);
                    elseif r == length(runsOI)
                        p(indy:indy+(list.plane.meas_v(plane)-1),...
                            indz:indz+...
                            (list.plane.stepsize_h(plane)-1),blk) = ...
                            phsqtemp(:,1:list.plane.stepsize_h(plane));
                    else
                        p(indy:indy+(list.plane.stepsize_v(plane)-1),...
                            indz:indz+...
                            (list.plane.stepsize_h(plane)-1),blk) = ...
                            phsqtemp(1:list.plane.stepsize_v(plane),...
                            1:list.plane.stepsize_h(plane));
                    end
                end

                clear phsqtemp
            end
            indz = indz+list.plane.stepsize_h(plane);
        end
    indy = indy+list.plane.stepsize_v(plane);
end


if sub.numerical

    %DEFINE CONSTANTS
    rho = 1.21; %Air desnity, kg/m^3.
    c = 343; %Speed of sound, m/s;
    k = 2*pi*parameter.foi/c; %Wavenumber, m^-1;

    %GENERATE SOURCE NUMERICALLY
    Q = 1;
    xs = 0;
    ys = 6*0.3048;
    ysi = -ys;
    zs = 7;
    SNRdB = 60; % dB Signal to noise ratio
    SNR = 10^(SNRdB/20);

    R = sqrt((Z-zs).^2+(X-xs).^2+(Y-ys).^2);
    Ri = sqrt((Z-zs).^2+(X-xs).^2+(Y-ysi).^2);
    p = 1i*rho*c*k*Q/4/pi./R.*exp(-1i*k.*R)+1i*rho*c*k*Q/4/pi./Ri.*exp(-
1i*k.*Ri);
    noise = sqrt(norm(p)/SNR/sqrt(numel(p)))*...
        sqrt(1/2)*(randn(size(p))+1i*randn(size(p)));
    p = p+noise;
end
```

read_benchmark_num_arc_pressures_20120118

```matlab
% FUNCTION [dataout] =
%         read_benchmark_num_arc_pressures_20120118(parameter,Z,X,Y)
%
% Description:
%
%  This function generates the measured pressures along the 23-m arc.
%
% Inputs:
%
%   plane        :    the number of the plane of interest
%   parameter    :    the variable contining all the parameters for this case
%   list         :    the variable containing possible parameter lists, used
%                     for indexing purposes
%   sub          :    all subroutine parameters (boolean)
%
% Outputs:
%
%   dataout.blocks  :    number of blocks for averaging
%   dataout.Z       :    data points in Z
%   dataout.X       :    data points in X
%   dataout.Y       :    data points in Y
%   dataout.p       :    measured pressures; Include in output only if plane
%                        < 5 orpressures are numerically generated.
%   dataout.p_ref   :    measured pressures at reference microphones;
%                        Include in output only if plane < 3.
%
% Author: David Krueger (dvdkrueger@gmail.com)
%   revised from code by: Alan Wall (alantwall@gmail.com)
%
% Date: 01/18/2012
%

function [dataout] = ...
    read_benchmark_num_arc_pressures_20120118(parameter,Z,X,Y)


%DEFINE CONSTANTS
rho = 1.21; %Air desnity, kg/m^3.
c = 343; %Speed of sound, m/s;
k = 2*pi*parameter.foi/c; %Wavenumber, m^-1;

%GENERATE DIRECTIONAL SOURCE NUMERICALLY
Nsources = 20;%number of partially correlated numerical sources
dsource = .4; %m distance between numerical sources
sourcedirection = 130;%[deg] sourcedirection angle meas from DOWNstream
phi = 2*pi*parameter.foi*dsource*...
    (0:Nsources-1)*cosd(sourcedirection)/c;%source phases
Q = 1;
SNRdB = 60; % dB Signal to noise ratio
SNR = 10^(SNRdB/20);
xs = 0;
ys = 6*0.3048;
```

```matlab
zs = 2:dsource:2+(Nsources-1)*dsource;

%LOOP THROUGH SOURCES
for nn = 1:Nsources
    R(:,:,nn) = sqrt((Z(:,:)-zs(nn)).^2+...
        (X(:,:)-xs).^2+(Y(:,:)-ys).^2);
end

    %LOOP THROUGH BLOCKS
    for blk = 1%:dataout.blocks

        %GENERATE VARIATION FACTOR
        variation = 1;%(1+vmag*(randn))*exp(1i*2*pi*rand);

        %LOOP THROUGH SOURCES
        for nn = 1:Nsources

            %PERFORM SIMULTANEOUS NUMERICAL MEASUREMENT
            p_num_all(:,:,nn) = ...
                1i*rho*c*k*Q/4/pi./R(:,:,nn).*exp(-1i*k.*R(:,:,nn))*...
)*...
                exp(1i*phi(nn))*variation;

            noise_num_all(:,:,nn) = ...
                sqrt(norm(p_num_all(:,:,nn)))/...
                SNR/sqrt(numel(p_num_all(:,:,nn))))*...
                sqrt(1/2)*(randn(size(p_num_all(:,:,nn)))+...
                1i*randn(size(p_num_all(:,:,nn)))));

            p_num_noise_all(:,:,nn,blk) =
p_num_all(:,:,nn)+noise_num_all(:,:,nn);
        end
    end
    dataout = squeeze(sum(p_num_noise_all,3));

end
```

coordinate_rotation_20110409

```matlab
% FUNCTION [R,S,T] =
%       coordinate_rotation_20110409(X,Y,Z,theta)
%
% Description:
%
%  This function transforms the original
%  coordinates to a rotated, conformal
%  coordinate system.
%
% Inputs:
```

```
%
%   X, Y, Z     :   Original coordinate matrices
%   theta       :   Angle of rotation
%
% Outputs:
%
%   R, S, T     :   Respective, rotated coordinate matrices
%
% Author: Alan Wall (alantwall@gmail.com)
%
% Date: 04/09/2011
%

function [R,S,T] = ...
    coordinate_rotation_20110409(X,Y,Z,theta)

R = -Z*sin(theta)+X*cos(theta);
S = Y;
T = Z*cos(theta)+X*sin(theta);

end
```

continued_geometry_20120214

```
% FUNCTION [Xc,Yc,Zc] =
%       continuation_20100915(X,Y,Z,ac,nn)
%
% Description:
%
% This function extends the
% meaurement aperture through
% mirroring, interpolating, and
% continuing analytically.
%
% Inputs:
%
%   X, Y, Z     :   Original geometry
%   ac          :   analytic continuation parameters
%   nn          :   frequency of column and row selection
%
% Outputs:
%
%   Xc, Yc, Zc  :   Extended geometry
%
% Author:   Alan Wall (alantwall@gmail.com) 09/24/2010
% Revised:  DAvid Krueger (dvdkrueger@gmail.com) 02/14/2012
%
% Date:     03/14/2012
%
```

```matlab
function [Xc,Yc,Zc] =  continued_geometry_20120214(X,Y,Z,ac,nn)

dx = X(1,2)-X(1,1);
dy = Y(1,1)-Y(2,1);
dz = Z(1,2)-Z(1,1);

%CONTINUATION GEOMETRY
xc = X(1,1)-dx*nn*ac.continue_points_reduced_z.dn:...
    nn*dx:...
    X(1,1)+dx*nn*floor((length(Z(1,:))+ac.continue_points_z.up)/nn)+0.00001;

yc = Y(1,1)+dy*ac.continue_points_y.up:...
    -dy:...
    -Y(1,1)-dy*ac.continue_points_y.dn-0.00001; %mirrored

zc = Z(1,1)-dz*nn*ac.continue_points_reduced_z.dn:...
    nn*dz:...
    Z(1,1)+dz*nn*floor((length(Z(1,:))+ac.continue_points_z.up)/nn)+0.00001;

[~,Xc] = ndgrid(yc,xc);
[Yc,Zc] = ndgrid(yc,zc);
```

read_benchmark_pressures_20120110

```matlab
% FUNCTION [dataout] =
%       read_benchmark_pressures_20120110(parameter,Z,X,Y,t)
%
% Description:
%
%  This function generates the
%  measured pressure map and geomery
%  for a specified plane.
%
% Inputs:
%
%   plane       :   the number of the plane of interest
%   parameter   :   the variable contining all the parameters for this case
%   list        :   the variable containing possible parameter lists, used
%                   for indexing purposes
%   sub         :   all subroutine parameters (boolean)
%
% Outputs:
%
%   dataout.blocks  :   number of blocks for averaging
%   dataout.Z       :   data points in Z
%   dataout.X       :   data points in X
%   dataout.Y       :   data points in Y
%   dataout.p       :   measured pressures; Include in output only if plane
%                       < 5 orpressures are numerically generated.
```

```matlab
%   dataout.p_ref    :   measured pressures at reference microphones;
%                        Include in output only if plane < 3.
%
% Author: Alan Wall (alantwall@gmail.com) 03/29/2011
% REvised David Krueger (dvdkrueger@gmail.com) 01/10/2012
%
% Date: 01/10/2012
%

function [dataout] = ...
    read_benchmark_pressures_20120110(parameter,Z,X,Y,t)
%DEFINE CONSTANTS
rho = 1.21; %Air desnity, kg/m^3.
c = 343; %Speed of sound, m/s;
k = 2*pi*parameter.foi/c; %Wavenumber, m^-1;

%GENERATE DIRECTIONAL SOURCE NUMERICALLY
Nsources = 20;%number of partially correlated numerical sources
dsource = .4; %m distance between numerical sources
sourcedirection = 130;%[deg] sourcedirection angle meas from DOWNstream
phi = 2*pi*parameter.foi*dsource*...
    (0:Nsources-1)*cosd(sourcedirection)/c;%source phases
Q = 1;
SNRdB = 60; % dB Signal to noise ratio
SNR = 10^(SNRdB/20);
xs = 0;
ys = 6*0.3048;
zs = 2:dsource:2+(Nsources-1)*dsource;

%LOOP THROUGH SOURCES
for nn = 1:Nsources
    R(:,:,nn) = sqrt((Z.rcs(:,:,t)-zs(nn)).^2+...
        (X.rcs(:,:,t)-xs).^2+(Y.rcs(:,:,t)-ys).^2);
end

% if sub.processing.measure_extended

    %LOOP THROUGH BLOCKS
    for blk = 1%:dataout.blocks

        %GENERATE VARIATION FACTOR
        variation = 1;%(1+vmag*(randn))*exp(1i*2*pi*rand);

        %LOOP THROUGH SOURCES
        for nn = 1:Nsources
            %PERFORM SIMULTANEOUS NUMERICAL MEASUREMENT
            p_num_all(:,:,nn) = ...
                1i*rho*c*k*Q/4/pi./R(:,:,nn).*exp(-1i*k.*R(:,:,nn))*...
)*...
                exp(1i*phi(nn))*variation;

            noise_num_all(:,:,nn) = ...
                sqrt(norm(p_num_all(:,:,nn)))/...
                SNR/sqrt(numel(p_num_all(:,:,nn))))*...
                sqrt(1/2)*(randn(size(p_num_all(:,:,nn)))+...
```

```
                1i*randn(size(p_num_all(:,:,nn))));

            p_num_noise_all(:,:,nn,blk) =
p_num_all(:,:,nn)+noise_num_all(:,:,nn);
        end
    end
    dataout.p_bench = squeeze(sum(p_num_noise_all,3));

end
```

continuation_20120224

```
% FUNCTION [pc,bb,alpha,Lambda,Falpha1,Fz,Fzinv,Fy,Fyinv,KZ,KY,KX] = ...
%     continuation_20120224(pin,Zin,Yin,foi,x_meas,x_rcs,...
%     file,pfi,ac,sub,nn)
%
% Description:
%
% This function extends the
% meaurement aperture through
% mirroring, interpolating, and
% continuing analytically.
%
% Inputs:
%
%   pin     :    partial fields
%   Zin,Yin :    original measurement geometry
%   foi     :    frequency of interest
%   x_meas  :    perpendicular measurement distance from shear layer
%   x_rcs   :    perpendicular reconstruction distance from shear layer
%   Cz,Cy   :    number of points to continue
%   figbn   :    data-saving base name
%   file    :    directories of important files
%   pfi     :    partial field index
%   ac      :    analytic continuation parameters
%   nn      :    every nn-th column ad row are kept in the final pressure
%
% Outputs:
%
%   pc      :    extended partial fields
%   Zc,Yc   :    extended geometry
%   bb      :    number of iterations
%
% Author:   Alan Wall (alantwall@gmail.com) 09/24/2010
% Revised:  David Krueger (dvdkrueger@gmail.com) 02/24/2102
%

function [pc,bb,alpha,Lambda,Falpha1,Fz,Fzinv,Fy,Fyinv,KZ,KY,KX] = ...
    continuation_20120224(pin,Zin,Yin,foi,x_meas,x_rcs,...
    file,pfi,ac,sub,nn)
```

```matlab
%INITIALIZE PARAMETERS
fig = 0; % Choose 0 to supress plotting, 1 otherwise.

%CONSTANTS
d = abs(x_meas-x_rcs); % m
c=343; % m/s
rho = 1.21; % kg/m^3
k = 2*pi*foi/c;

%DEFINE HOLORAM AND RECONSTRUCTION GEOMETRY
% (See Ref. 6)
dz = abs(Zin(1,2)-Zin(1,1)); % m
Ninz = length(pin(1,:));
Lz = (Ninz-
1+ac.continue_points_reduced_z.up+ac.continue_points_reduced_z.dn)*dz;
Nz = Ninz+ac.continue_points_reduced_z.up+ac.continue_points_reduced_z.dn;
qinz = -floor(Ninz/2):floor((Ninz-1)/2);
qz = -floor(Nz/2):floor((Nz-1)/2);
zin = qinz*dz;
z = qz*dz;

dy = abs(Yin(2,1)-Yin(1,1)); % m
Niny = length(pin(:,1));
Ly = 2*Yin(1,1)+dy*(ac.continue_points_y.up+ac.continue_points_y.dn);
% Ly = (Yin(1,1)-
Yin(end,1))+dy*(ac.continue_points_y.up+ac.continue_points_y.dn);
Ny = round(Ly/dy+1);
qiny = -floor(Niny/2):floor((Niny-1)/2);
qy = -floor(Ny/2):floor((Ny-1)/2);
yin = qiny*dy;
y = qy*dy;

%MIRROR OVER GROUND
pin_mir = [pin; flipud(pin)];

%INTERPOLATE BETWEEN MEASURED AND REFLECTED
Nybelow = floor(Yin(end,1)/dy);

indiny = [ac.continue_points_y.up+1:ac.continue_points_y.up+Niny ...

ac.continue_points_y.up+Niny+2*Nybelow+1:ac.continue_points_y.up+Niny+2*Nybel
ow+Niny];
indiny2 =
[ac.continue_points_y.up+1:ac.continue_points_y.up+Niny+2*Nybelow+Niny];
indinz = ac.continue_points_reduced_z.dn+(1:Ninz);

yint1 = y(indiny);
yint2 = y(indiny2);
[Zt1,Yt1] = meshgrid(zin,yint1);
[Zt2,Yt2] = meshgrid(zin,yint2);

ptemp1 = interp2(Zt1,Yt1,pin_mir,Zt2,Yt2,'spline');

%PLOT MIRRORED AND INTERPOLATED DATA
```

```matlab
if fig
    figure
    pcolor(Zt2,Yt2,20*log10(abs(ptemp1)/20e-6));
    hold on
    contour(Zt2,Yt2,20*log10(abs(ptemp1)/20e-6),20,'k');
    shading interp
    h = colorbar;
    ylabel('dB re 20 \muPa')
    xlabel('Z (m)')
    ylabel('Y (m)')
    title('Interpolated SPL')

    figure
    plot(yint1,20*log10(pin_mir(:,15)/20e-6),'x')
    hold on
    plot(yint2,20*log10(ptemp1(:,15)/20e-6));
    xlabel('Y (m)')
    ylabel('dB re 20 \muPa')
    grid on
    title('Interpolated SPL, One Column')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ANALYTIC CONTINUATION

%ZERO-PAD
p = padarray(padarray(ptemp1,...
    [ac.continue_points_y.up ac.continue_points_reduced_z.dn],0,'pre'),...
    [ac.continue_points_y.dn ac.continue_points_reduced_z.up],0,'post');

%PLOT ZERO-PADDED DATA
if fig
    figure
    pcolor(Z,Y,20*log10(abs(p)/20e-6));
    shading interp
    daspect([1 1 1]);
    title({'Measured SPL (Numerical)',[int2str(foi),' Hz, X =
',num2str(x_meas,2),' m']});
    xlabel('Z (m)');
    ylabel('Y (m)');
    h=colorbar;
    ylabel(h,['dB re 20 \muPa']);
end

%PERFORM FFT, USING WILLIAMS' FFT FUNCTION
dkz = 2*pi/Lz; % See Ref. 6, (1.54)
mz = qz; % See Ref. 6, (1.53)
kz = mz*dkz; % See Ref. 6, (1.53)

% See Ref. 6, (1.55, 1.62)
for mm = 1:Nz
    Fz(mm,:) = Lz/Nz*exp(-1i*2*pi*mz(mm)*qz/Nz); % Forward DFT Operator
    Fzinv(:,mm) =1/Lz*exp(1i*2*pi*mz(mm)*qz/Nz); % Inverse DFT Operator
end
```

```matlab
dky = 2*pi/Ly;
my = qy;
ky = my*dky;

for mm = 1:Ny;
    Fy(mm,:) = Ly/Ny*exp(-1i*2*pi*my(mm)*qy/Ny); % Forward DFT Operator
    Fyinv(:,mm) =1/Ly*exp(1i*2*pi*my(mm)*qy/Ny); % Inverse DFT Operator
end
[KZ,KY] = meshgrid(kz,ky);

%GENERATE LAMBDAS FOR FILTER
% (See Ref. 7, Sec. III.2)
for aaa = 1:Ny
    for bbb = 1:Nz
        if KZ(aaa,bbb)^2 + KY(aaa,bbb)^2 <= k^2
            KX(aaa,bbb) = sqrt(k^2-KY(aaa,bbb)^2-KZ(aaa,bbb)^2);
        else
            KX(aaa,bbb) = -1i*sqrt(KY(aaa,bbb)^2+KZ(aaa,bbb)^2-k^2);
        end
    end
end

Lambda = rho*c*k*exp(-sqrt(KY.^2+KZ.^2-k^2)*d)./...
    (1i*sqrt(KY.^2+KZ.^2-k^2));

%INITIALIZE REGULARIZATION
reg_alpha = logspace(-10,90,200);%DWK20111222

%ITERATE THROUGH FILTERS
% (See Ref. 6, Eqs. I1-I4)

p2 = p;
ptemp = zeros(size(p));
cflag = 0;

%LOOP THROUGH ITERATIONS
for bb = 1:ac.iter_lim

    disp([num2str(foi),' Hz, Iteration ',int2str(bb)]);
    clear P sigma Falpha1 Pf

    %TAKE DFT
    P = (Fz*(Fy*(p2)).').';

    if bb == 1
        %PLOT 2D DFT F OR FIRST ITERATION
        range2 = 20*log10([min(min([abs(P)])) max(max([abs(P)]))]/20e-6);
        dif2 = (range2(2)-range2(1))/100;
        Caxis2 = [range2(1)-dif2 range2(2)+dif2];

        if fig
            figure
            pcolor(KZ,KY,20*log10(abs(P)/20e-6));
            hold on
            t = 0:0.01:2*pi;
```

```matlab
        polar(t,k*ones(size(t)),'k'); % Radiation circle
        h = colorbar;
        daspect([1 1 1])
        ylabel(h,'dB re 20 \muPa');
        xlabel('k_z (m^{-1})')
        ylabel('k_y (m^{-1})')
        title({'Angular Spectrum',['2-D FFT of P, ',int2str(foi),' Hz']})
        shading interp
        caxis(Caxis2)
    end
end


%ESTIMATE THE NOISE VARIANCE USING MDP
% In this process, we must assume that the "Fourier coefficients of p2
% have dropped off sufficiently into the noise so that we may estimate
% the variance % (See Ref. 7, Sec.  III)

%A POSSIBLE ALTERNATIVE METHOD FOR CHOOSING sigma
% Sigma is difficult to estimate for our given problem.  This solution
% looks for a pseudo-convergence of sigma, as the minimum wavenumber
% assumed to beassociated with noise only is varied.

%LOOP THROUGH NUMBER OF CONTINUED MEASUREMENT POINTS
for i = 1:ceil(Ny/2)

    %ESTIMATE MINIMUM NOISE-RELATED WAVENUMBER
    km = ky(floor(Ny/2)+i);

    % Initialize index and number of basis vectors
    Q = 0;
    clear Up

    %LOOP THROUGH WAVENUMBERS
    for a = 1:Ny
        for b = 1:Nz

            %SELECT ASSUMED NOISE-RELATED WAVENUBERS
            if sqrt(KY(a,b)^2+KZ(a,b)^2) > km
                Q = Q+1;
                Up(Q) = P(a,b);
            end

        end
    end

    %RETURN A sigma
    SIGMA(i) = norm(Up)/sqrt(Q);
end


%PSEUDO-CONVERGENCE TEST
% I'm not sure if the following method of estimating sigma is the best,
% but it seems to get close in many cases.
[temp2,ind] = find(min(abs(diff(SIGMA)))==abs(diff(SIGMA)));
sigma = SIGMA(ind);
```

```matlab
    %PLOT sigma VALUES
    if bb == 1
        if fig
            figure
            plot(ky(floor(Ny/2)+1:end),SIGMA);
            xlabel('k_{max}')
            ylabel('\sigma')
            title(['\sigma = ',num2str(sigma)]);
        end
    end

    %FROM VARIANCE, SOLVE FOR alpha
    % We vary alpha, filter P accordingly, and thake the IDFT.  We then use
    % Ref. 6, Eq. 8.  Iterate this through alphas, and find
    % the appropriate one.

    % (See Ref. 6, (7))
    for i = 1:length(reg_alpha);
        J(i) =
mdp2D(reg_alpha(i),Lambda,Fyinv,Fzinv,P,p2,Ny,Nz,sigma,k,KY,KZ);
    end

    [low,ind]=min(J);
    alphalow = reg_alpha(ind);

    %SELECT alpha
    %(See Ref. 6, (7))
    alpha = fminbnd('mdp2D',.001*alphalow,1000*alphalow,...

optimset('Display','off'),Lambda,Fyinv,Fzinv,P,p2,Ny,Nz,sigma,k,KY,KZ);
    Alpha(bb) = alpha;
    disp(' ');
    disp(['alpha = ',num2str(alpha,5)])
    disp(['max(|lambda|^2) = ',num2str(max(max(abs(Lambda).^2)),5)]);
    disp(' ');

    %PLOT alpha VALUES
    if bb == 1
        if fig
            figure
            semilogx(reg_alpha,J)
            hold on
            plot([alpha alpha],[min(J) max(J)],'r--');
            title({['\alpha = ',num2str(alpha,3),', First Iteration'],'Cost
function to be minimized'});
            xlabel('\alpha');
            ylabel('J')
        end
    end

    %GENERATE FILTER
    Falpha1 = (abs(Lambda).^2./...
        (abs(Lambda).^2+alpha*(alpha./(alpha+abs(Lambda).^2)).^2));

    %LOOP THROUGH WAVENUMBERS
```

```matlab
for a = 1:Ny
    for b = 1:Nz

        %CORRECT FOR FILTER CENTER
        % (See Ref. 7)
        if sqrt(KY(a,b)^2+KZ(a,b)^2) < k
            Falpha1(a,b) = 1;
        end

    end
end

%FIRST-ITERATION ALPHA
if bb == 1
    Falpha11 = Falpha1;
end

%FILTER DATA, AND TAKE INVERSE DFT
Pf = Falpha1.*P;
pf = Fyinv*(Fzinv*Pf.').';

%TO STOP THE ITERATION:
% (See Ref.6, (13))
Sigma(bb) = sigma;
Change(bb) = norm(pf-p2)/sqrt(Ny*Nz);

if norm(pf-p2)/sqrt(Ny*Nz) < ac.ad_hoc*sigma && bb > 10
    cflag = 1;
    break
end

%REPLACE REAL DATA
clear ptemp
ptemp = pf;
if bb ~= ac.iter_lim
    pf(indiny,indinz) = [pin; flipud(pin)];%pin;%
end

if bb == 1

    %PLOT FILTERED 2D DFT
    if fig

        %COLOR AXIS
        Caxis =20*log10([min(min(abs(pin))) max(max(abs(pin)))]/20e-6);

        figure
        pcolor(KZ,KY,20*log10(abs(Pf)/20e-6));
        hold on
        t = 0:0.01:2*pi;
        polar(t,k*ones(size(t)),'k'); % Radiation circle
        h = colorbar;
        daspect([1 1 1])
        ylabel(h,'dB re 20 \muPa');
        xlabel('k_z (m^{-1})')
```

```matlab
            ylabel('k_y (m^{-1})')
            title({'Filtered Angular Spectrum',['2-D FFT of Pf, ',...
                int2str(foi),' Hz']})
            shading interp
            caxis(Caxis2)

            figure
            pcolor(Z,Y,20*log10(abs(ptemp)/20e-6));
            shading interp
            daspect([1 1 1]);
            title({'Filtered SPL ,First Iteration',[int2str(foi),...
                ' Hz, X = ',num2str(x_meas,2),' m']});
            xlabel('Z (m)');
            ylabel('Y (m)');
            h=colorbar;
            ylabel(h,'dB re 20 \muPa');
            caxis(Caxis);
        end
    end

    %PREPARE FOR NEXT ITERATION
    clear p2
    p2 = pf;

end

%FINAL CONTINUED PRESSURE
pc = pf;

if fig
    %PLOT ALPHA OVER ITERATIONS
    figure
    semilogy(Alpha)
    xlabel('iteration')
    ylabel('\alpha')

    %PLOT CONVERGENCE PARAMETERS OVER ITERATIONS
    figure
    plot(ac.ad_hoc*Sigma,'r')
    hold on
    plot(Change)
    xlabel('Iteration');
    legend([num2str(ac.ad_hoc),'\sigma^i'],'Eq. 13')

    %PLOT CONTINUED PARTIAL FIELD
    figure
    pcolor(Z,Y,20*log10(abs(pf)/20e-6));
    hold on
    shading interp
    daspect([1 1 1]);
    if cflag
        title({['Filtered Partial Field # ',int2str(pfi)],...
            ['Convg. at ',int2str(bb)],[int2str(foi),' Hz, X = ',...
            num2str(x_meas,2),' m']});
    else
```

```matlab
        title({['Filtered Partial Field # ',int2str(pfi)],...
            ['No Convergence, ',int2str(bb),' Iters.'],[int2str(foi),...
            ' Hz, X = ',num2str(x_meas,2),' m']});
    end
    xlabel('Z (m)');
    ylabel('Y (m)');
    h=colorbar;
    ylabel(h,'dB re 20 \muPa');
    caxis(Caxis);

    %PLOT FILTERED PARTIAL FIELD IN WAVENUMBER SPACE
    figure
    pcolor(KZ,KY,20*log10(abs(Pf)/20e-6));
    hold on
    t = 0:0.01:2*pi;
    polar(t,k*ones(size(t)),'k'); % Radiation circle
    h = colorbar;
    daspect([1 1 1])
    ylabel(h,'dB re 20 \muPa');
    xlabel('k_z (m^{-1})')
    ylabel('k_y (m^{-1})')
    title({' Filtered Angular Spectrum',['2-D FFT of Pf, ',...
        int2str(foi),' Hz']})
    shading interp
    caxis(Caxis2)
end

end
```

STSF_cylFFT_20120614

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DESCRIPTION: PERFORMS CYLINDRICAL FOURIER NAH WITH ANALYTIC CONTINUATION
%AUTHOR: DAVID W. KRUEGER/BYU
%SUBROUTINES: mdpCYL.m, NumericalSource_20110606.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program:
% Loads complex pressures from ground reference microphones.
% Computes cross-spectral matrix (CSM) and singular value decomposition.
% Chooses appropriate singular values and singular vectors.
% Applies analytic continuation on scaled singular vectors.
% Propagates singular vector in cylindrical coordinates.
% Sums propagated singular vectors energetically.
%
% It must be run from a directory or path containing:
% mdpCYL.m and NumericalSource_20110606.m
%
%BIBLIOGRAPHY FOR PAPERS REFERENCED IN THE PROGRAM:
% D. Long, J. Peters, M. Anderson, "Evaluating Turbofan Exhaust Noise and
% Source Characteristics from Near Field Measurements," AIAA 2009-3214,
% 15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics
```

```matlab
% conference 11-13 May 2009, Miami, Florida.
%
% E. G. Williams, 'Fourier Acoustics, Sound Radiation and Nearfield
% Acoustical Holography' (Academic Press, San Diego, 1999), pp. 115-180.
%
% E. G. Williams, "Regularization methods for near-field acoustical
% holography," J. Acoust. Soc. Am. 110,1976-1988 (2001).
%
% E. G. Williams, "Continuation of acoustic near-fields," J. Acoust. Soc.
% Am. 113, 1273-1281 (2003).
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc; drawnow;
%#ok<*SAGROW>


%INITIALIZE PARAMETERS
sub.numerical = 0; % a boolean. 1 numerical source, 0 for measured data.
pfdB = 25;   % Retain all the singular vectors whose singular value is
             % within this many dB down from first singular vector.
% N = 46;      % Number of equally spaced microphones to be used
foi = 105;  % Frequency of interest
pathname = 'Z:\Students\Alan Wall\';    % Where the data is stored
cont = 1;    % Choose 0 to supress analytic continuation
adhoc = 0.1;% Williams2003,(Eq.13) Factor used to define regularization
             % break point.  0.2-0.4 is a typical range.
alphdiff = 1e-4;% If regularization parameter alpha doesn't change by this
                % percent, break.
date = '20110609';
figdir = ['W:\NAH\figures\Figs',date]; % Where to save figures
mdir = 'W:\NAH\CylindricalNAH';         % Where current m file is.
savbasename = ['cylNAH_',date,'_'];    % Figure base name
fn = 1;      % Figure number, appends to the figure base name and increments
             % by one when saving figures.
xprop = 16.5;%.5;%100;% [m] Distto propagate outwards to from meas mics.
xin = 11.5824;%.5;%11.5;    % [m] Dist to propagate inwards toward source.
B = 50;      % Set max number of iterations
Czpts = 100;%10;% How many points to analytically continue on each side.
plotarc=0;  % A boolean. (1 plot 22.9m arc, 0 don't plot 22.9m arc)
savearcfig=0;% A boolean.(1 save plot with 22.9m arc, 0 don't save plot)
rho = 1.21; % [kg/m^3] Density of air
c = 343;    % [m/s] Speed of sound in air
k = 2*pi*foi/c; % [1/m] Acoustic wave number



savefig = 0;%input('Do you want to save figures? (0 no or 1 yes) : ');%
            % A boolean (1 save figures, 0 don't save)
Savevar = 0;%input('Do you want to save variables? (0 no or 1 yes) : ');%
            % A boolean (1 save variables, 0 don't save)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LOAD DATA
% Even though we are only concerned with the refernce microphone
% microphones on the ground, the program is set up to load in a full plane
% of calculated complex pressure data.
plane = 2;  % Plane of interest
runOI = 5;  % Measurement run of interest
```

```matlab
scan = 1;    % Scan number of interest
cond = 3;    % 1 for Idle, 2 for Intermediate, 3 for Mil, 4 for AB


cpv = 6;     % Complex Pressure Version, cpv and CPV help choose the correct
             % block size.
CPV = 4:7;


%CHOOSE OTHER PARAMETERS
BLOCKS = [75 607 151 303];
BLOCKS2 = [39 319 79 159];
CONDS = {'Idle','Interm','Mil','AB'};
Z = [-10:2:80 85:2:91 ]*0.3048; % [m] Z location of ground references mics
X = 38*0.3048*ones(size(Z));    % [m] X location of ground reference mics
Y = zeros(size(Z));             % [m] Y location of ground reference mics


%IDENTIFY RUNS ASSOCIATED WITH PLANES OF INTEREST
% This information may be found in the file "F22_Holloman_MicMapPlan_v2.m"
if plane == 1
    Runs = [1 2 6];   % Runs of interest in each plane from top to bottom
    ORSruns = [1 2 6];% Order of runs as they apear in
                      % F22Holloman2009Geometry
elseif plane == 2
    Runs = [9 10 5];
    ORSruns = [3 4 5 9 10];
elseif plane == 3
    Runs = [8 7];
    ORSruns = [7 8];
end


%LOAD DATA
indy = 1;
runind = find(Runs==runOI);    % Run number of interest
if cond == 4
    blocks = BLOCKS2(CPV==cpv);
else
    blocks = BLOCKS(CPV==cpv);
end

if runind == 1 && cond ~= 4
    cpv_temp = cpv;
    cplist = [5 7 6 4];
    cpv =  cplist(find(cpv == cplist)-1);

    blocks_temp = blocks;
    blocks = BLOCKS2(CPV==cpv);
end

pathname2 = [pathname,...
    'HollomanF22_2009\ComplexPressures_v',...
    int2str(cpv),'\']; % Directory for extracting complex pressures
filename2 = ['CPv',int2str(cpv),'_',num2str(foi,'%04i'),'Hz_Run',...
    num2str(runind),'_',CONDS{cond},'_',num2str(blocks,'%03i'),...
    'blocks.mat']; % Filename of complex pressures
eval(['load(''',pathname2,filename2,''')'])
```

```matlab
if runind == 1 && cond ~= 4
    cpv = cpv_temp;
    blocks = blocks_temp;
end

% This pulls out the ground reference microphones from the correct plane
% and puts them in order along Z.  Note: it pulls in 52 measurement
% positions.  The first 46 are equally spaced, then there's a break between
% the next 4. The last 2 (CH's 51and52) can be ignored since they are
% under the plane.
p = permute(CPrefs(1,[46:50,1:41,42:45],1:blocks),[3 2 1]);
% return

%INTERPOLATE REFERENCE MICROPHONES TO INCLUDE THE LAST FOUR
Zint = [Z(1:46),(82:2:90)*0.3048];
Xint = X(1,1)*ones(size(Zint));
% pmag = spline(Z,abs(p),Zint);
% pang = spline(Z,angle(p),Zint);
% pint = pmag.*exp(-1i*pang);

pswap = [p(:,1:46) -p(:,46+1:50)];%The polarity must be swapped on last4

for ii = 1:numel(pswap(:,1))
    pswapangtemp(ii,:) = unwrap(angle(pswap(ii,:)));
end
if foi>=210
    pswapangtemp(:,47:50) = pswapangtemp(:,47:50)-2*pi;%Correctfor4ft gap
end

pswapmag = interp1(Z,abs(pswap.'),Zint,'spline').';
pswapang = interp1(Z,pswapangtemp.',Zint,'spline').';
pswapint = pswapmag.*exp(1i*pswapang);

% figure;
% for ii = 1:4
% subplot(2,2,ii)
% plot(Z,unwrap(angle(pswap(ii+4,:))),'gp','Markersize',8); hold on;
% plot(Zint,unwrap(pswapang(ii+4,:)),'.-','Markersize',8);
% xlim([Z(1)-1 Z(end)+1]); title(['data block ',num2str(ii+4)]);
% ylim([-25 10]); xlabel('Z (m)'); ylabel('radians'); grid on;
% legend('measured','interpolated','location','SouthWest');
% end

% for ii = 1:numel(pswap(:,1))
% UN(ii,:) = unwrap(angle(pswap(ii,:)));
% end
% UN = unwrap(angle(pswap.'));
% figure;
% for ii = 1:4
% subplot(2,2,ii)
% plot(Z(1:46),UN(ii+8,1:46),'gp','Markersize',8); hold on;
% plot(Z(47:50),UN(ii+8,47:50)-2*pi,'gp','Markersize',8);
% end

% xlabel('Z (m)'); ylabel('unwrapped phase (radians)');
```

```matlab
% title('''spline'' interpolation of phase');
% legend('measured','interpolated');


%END LOAD DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE JET CENTERLINE AND RECONSTRUCTION GEOMETRY
% clear N;
N = numel(Zint);%51;
%JET GEOMETRY
Zj = Zint;
Xj = zeros(size(Zj));
Yj = 75/12*0.3048*ones(size(Zj));   % Defined in this manner, the origin is
                                     % on the ground below the jet.
%PROPAGATED GEOMETRY
Za = Zint;
Ya = 3/12*.3048;%6*.3048;%     % [m] 3 in off ground
Xa = X(1,1)-xin:.5:X(1,1)+xprop;    % Define all the reconstruction points.
ap = sqrt((Yj(1,1)-Ya)^2+Xa.^2);% Define all the reconstruction distances.
rh = sqrt(X(1,1)^2+Yj(1,1)^2);   % Distance from jet to ref. mic. array.

dZ=abs(Zint(2)-Zint(1));   %Spacing of each ref. microphone.
if cont==1
    Cz = Czpts;   % How many points to continue out to on each side of meas
elseif cont==0
    Cz = 0;
end
NN = N+2*Cz;     % Number of points in continued measurement.
ZZ = Zint(1)-Cz*dZ:dZ:Zint(N)+Cz*dZ;   % Where cont'd pts are located in Z.



if sub.numerical==0
elseif sub.numerical ==1
    clear pswapint;
    tic
    [Y,p,p_bench,p_noimage,blocks] = NumericalSource_20110606(foi,...
        Zint,Xint,Xa,ZZ);
    toc
    pswapint = p;
 end

tic

%END DEFINE GEOMETRY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULATE SVD OF CROSS-SPECTRAL MATRIX (TO GET SINGULAR VECTORS)
K = pswapint'*pswapint/blocks;% See Long2009, Eq.1 build cross-spect matrix
[C,Lamb,C2] = svd(K);   % See Long2009, Eq. 2 compute SVD
lamb = sqrt(diag(Lamb));% See Long2009, Eqs. 2 and 3

for n = 1:length(Zint)
    Sini(:,n) = C(:,n)*lamb(n); % Long 2009, Eq. 3 scale each sing. vect.
```

```matlab
end

S = Sini(:,1:N);     % Only keep first N singular vectors since the ref mic
                     % array has four mics geometrically separated.


% pfs is how many singular values are within the user defined pfdB dB of
% the largest singular value.
pfs= find(diag(10*log10(Lamb))>diag(10*log10(Lamb(1)))-pfdB,1,'last');

% Plot angular spectrum of first singular value
Nint = numel(Zint);
Lint = abs(Zint(end)-Zint(1));
dkzint = 2*pi/Lint;
qint = -floor(Nint/2):floor((Nint-1)/2);
kzint = qint*dkzint;
figure;
subplot 211;
plot(Zint,20*log10(abs(S(:,1))/2e-5));
xlabel('Z (m)'); ylabel('SPL (dB re20\muPa)');
title('First singular vector before anlyt.cont.');
subplot 212;
Skspace = 20*log10(abs(fftshift(fft(S(:,1)))));
plot(kzint,Skspace);
hold on;
plot([k,k],[min(Skspace) max(Skspace)],'k--');
plot([-k,-k],[min(Skspace) max(Skspace)],'k--');
xlabel('k_z'); ylabel('dB re 1');
title('Angular spectrum of first singular vector before anlyt.cont.');
ylim([min(Skspace)-.5 max(Skspace)+.5]);
legend('first sing. vect.','k','location','west');

%END CALCULATE SVD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%START ANALYTIC CONTINUATION
%Create 2D FFT operators in cyl. coord. see William1999(Eq5.27,pg155)
L = abs(ZZ(NN)-ZZ(1));  % Length of continued array.
dz = L/NN;       % Williams1999,(5.28)
dphi = 2*pi/NN;  % Williams1999,(5.28)
dkz = 2*pi/L;    % Williams1999,(5.28)
q = -floor(NN/2):floor((NN-1)/2);
zqp = q*dz;      % Williams1999,(5.28)
phiqp = q*dphi;  % Williams1999,(5.28)
kzmp = q*dkz;    % Williams1999,(5.28)


qp = 0:NN-1;     % Williams1999,(5.28)
mp = qp;         % Williams1999,(5.29)
pp = 0;          % Assuming axisymmetry, pp (meaning p prime,
np = pp;         % Williams1999,(5.27)) can be zero.


if np~=pp
    display('*Canceled since pp~=np.  Modify code accordingly.');
    return
end
```

```matlab
for mm = 1:numel(mp)% Williams1999,(5.27 and 5.29)
                    % Note: the coefficients L/N and 1/L better match
                    % Williams1999(Eq.1.55 and 1.62). Matlab's fft must be
                    % scaled by L/NN and ifft must be scaled by NN/L,
                    % though this code does not use Matlab's fft.
    Fz(mm,:) = L/NN*exp(-1i*(2*pi*mp(mm)*qp)/NN).*(-1).^(qp+mp(mm));
    Fzinv(mm,:) = 1/L*exp(1i*(2*pi*mp(mm)*qp)/NN).*(-1).^(qp+mp(mm));
end
if pp ~= 0
    for nn = 1:numel(np)
        Fphi(nn,:) = exp(-1i*(2*pi*np(nn)*pp)/N).*(-1).^(pp+np(nn));
        Fphiinv(nn,:) = exp(1i*(2*pi*np(nn)*pp)/N).*(-1).^(pp+np(nn));
    end
    [Ss,temp] = meshgrid(S(:,pfnum));
elseif pp == 0
    Fphi = exp(-1i*(2*pi*np*pp)/NN).*(-1).^(pp+np);
    Fphiinv = exp(1i*(2*pi*np*pp)/NN).*(-1).^(pp+np);
end


for aa = 1:NN;  % krmp = sqrt(k^2-kzmp.*conj(kzmp)); Ensure correct sign of
                % imaginary part of evanescent wave numbers. krmp is used
                % in Lambda and the propagator G.
    if kzmp(aa)*conj(kzmp(aa)) <= k^2
        krmp(aa) = sqrt(k^2-kzmp(aa)*conj(kzmp(aa)));
    else
        krmp(aa) = 1i*sqrt(kzmp(aa)*conj(kzmp(aa))-k^2);
    end
end

% For Lambda, see Williams 2001, Eq.16
Lambda = 1i*rho*c*k./(krmp).*besselh(np,1,krmp(mm)*ap(1))./...
    besselh(np,1,krmp(mm)*rh);
reg_alpha = logspace(-60,70,200);   % Initialize some regularization alpha
                                    % array within which the correct one
                                    % will be found.

%ITERATE THROUGH SINGULAR VECTORS
for ff = 1:pfs
    pfnum = ff;
    clear ppad G Pr;
    ppad = padarray(S(:,pfnum),Cz); % initially zero pad singular vector

    %%%%%%%%%%%%%%%%%%%%%
    if ff == 1
        figure;   %PLOT INITALLY ZERO-PADDED SINGULAR VECTOR
        plot(ZZ,abs(ppad),'b-',...
            [Zint(1) Zint(1)],[min(abs(ppad)) max(abs(ppad))],'k--',...
            [Zint(end) Zint(end)],[min(abs(ppad)) max(abs(ppad))],'k--');
        xlabel('Z (m)'); ylabel('|Pa|');
        axis([ZZ(1) ZZ(end) -2 max(abs(ppad))+2]);
        title('Zero-padded first singular vector');
        legend('1^{st} singular vector',...
            'Measurement ends','location','West');
        if savefig==1
            cd(figdir)
            set(gcf,'paperpositionmode','auto');
```

```matlab
            print('-dtiff','-r300',[savbasename,int2str(foi),'Hz_Cz',...
                num2str(Cz),'_xprop',num2str(xprop),'_',...
                num2str(fn,'%02i'),'_ZeroPad_color'])
            fn = fn+1;
            cd(mdir)
        end
    end


%ITERATE THROUGH FILTERS
p2 = ppad;
cflag = 0;  % A boolean, used to signal if iteration loop broke because
            % convergence criteria (see Williams2003,(Eq.13)) was met.
aflag = 0;  % A boolean, used to signal if iteration loop broke because
            % alpha didn't change by a certain percentage between
            % iterations. (0 no, 1 yes).


for bb = 1:B;
    clear sigma Falpha1 Pf

    % Take spatial fft of singular vector
    if pp ~= 0
        P2 = (Fz*(Fphi*Ss).').';
    elseif pp == 0
        P2 = (Fz*(Fphi*p2.').').';
    end

    %ESTIMATE THE NOISE VARIANCE USING MorozovDiscrepancyPrinciple
    % see Williams2003(Eqs.7-10)
    for ii = 1:ceil((NN-1)/2)
        km = kzmp(floor(NN/2)+ii);
        Q = 0; % Initialize index and number of basis vectors
        clear Up
        for cc = 1:NN
            if sqrt(kzmp(cc)^2) > km
                Q = Q+1;
                Up(Q) = P2(cc);
            end
        end
        SIGMA(ii) = norm(Up)/sqrt(Q); % An estimate of the noise variance
                                      % see Williams2003,(Eq.10)
    end

    [temp,ind] = find(min(abs(diff(SIGMA)))==abs(diff(SIGMA)));
    sigma = SIGMA(ind); % choose the first minimum of SIGMA to be the noise
                        % variance

%GENERATE FILTER
    % FROM VARIANCE, SOLVE FOR alpha
    % We vary alpha, filter P accordingly, and take the IDFT.  We then use
    % Williams2003,(Eq.8).  Iterate this through alphas, and find
    % the appropriate one.

    for i = 1:length(reg_alpha);     % To find the minimum of a function, we
                                     % use Matlab's fminbnd which requires
```

```matlab
                                        % the function to be minimized be a
                                        % separate function.
        J(i) = mdpCYL(reg_alpha(i),Lambda,Fphiinv,...
            Fzinv,P2,p2,NN,sigma,k,kzmp);
end

[low,ind]=min(J);
alphalow = reg_alpha(ind);


alpha = fminbnd('mdpCYL',.001*alphalow,1000*alphalow,...
    optimset('Display','off'),Lambda,Fphiinv,Fzinv,P2,p2,NN,sigma,...
    k,kzmp);
Alpha(bb) = alpha;  % Keep track of alphas so they may be plotted later

% Build the regularization filter with the optimized alpha and replace
% the values inside the radiation circle |kzmp|<|k| with unity.
Falpha1 = (abs(Lambda).^2./...
    (abs(Lambda).^2+alpha*(alpha./(alpha+abs(Lambda).^2)).^2));
for cc = 1:NN
    if sqrt(kzmp(cc)^2) < k
        Falpha1(cc) = 1;
    end
end

%FILTER DATA, AND TAKE INVERSE DFT (Williams2003,(Eq.I3))
Pf = Falpha1.*P2;
%%%%%%%%%%%%%%%%%%%%
if ff==1
if bb==1
    figure;    %PLOT ANGULAR SPECTRUM BEFORE AND AFTER FILTER
    subplot 311;
    semilogy(kzmp,abs(P2),[-k -k],[min(abs(P2)) max(abs(P2))],'k--',...
        [k k],[min(abs(P2)) max(abs(P2))],'k--');
    title('Cylindrical FFT, angular wave spectrum');ylabel('|Pa/m|');
    axis([kzmp(1) kzmp(end) min(abs(Pf)) max(abs(P2))]);
    set(gca,'Xtick',ceil(kzmp(1)):floor(kzmp(end)));
    subplot 312;
    plot(kzmp,Falpha1,[-k -k],[0 1],'k--',[k k],[0 1],'k--');
    xlim([kzmp(1) kzmp(end)]); ylabel('F^\alpha'); title('F^\alpha');
    set(gca,'Xtick',ceil(kzmp(1)):floor(kzmp(end)));
    subplot 313;
    semilogy(kzmp,abs(Pf),[-k -k],[min(abs(P2)) max(abs(P2))],'k--',...
        [k k],[min(abs(P2)) max(abs(P2))],'k--');
    xlabel('k_{zm''}');   ylabel('|Pa/m|');
    title('Filtered first iteration, angular wave spectrum');
    axis([kzmp(1) kzmp(end) min(abs(Pf)) max(abs(P2))]);
    set(gca,'Xtick',ceil(kzmp(1)):floor(kzmp(end)));
    if savefig==1
        cd(figdir)
        set(gcf,'paperpositionmode','auto');
        print('-dtiff','-r300',[savbasename,int2str(foi),'Hz_Cz',...
            num2str(Cz),'_xprop',num2str(xprop),'_',...
                num2str(fn,'%02i'),'_FiltPFirstIter_color'])
        fn = fn+1;
        cd(mdir)
    end
```

```matlab
    end
    end

    pf = (Fzinv*(Fphiinv.'*Pf).');

    %TO STOP THE ITERATION
    Sigma(bb) = sigma;  % Keep track of sigma so it may be plotted later.
    Change(bb) = norm(pf-p2)/sqrt(NN); % The change in the continued sing.
                                       % vect. (see Williams2003,(Eq.13))
    if bb>=10    %This sets a minimum number of iterations.
        if norm(pf-p2)/sqrt(NN) < adhoc*sigma
            cflag = 1;
            disp('Broke because continuation criteria was met.');
            break
        end
    end

    %REPLACE CENTER WITH REAL DATA (See Williams2003,(I4)
    clear ptemp
    ptemp = pf;

    if bb ~= B
        pf(NN/2+1-N/2:NN/2+1+N/2-1) = S(:,pfnum);
    end

    %%%%%%%%%%%%%%%%%%%%

    clear p2
    p2 = pf;


end
Pfall(:,ff) = Pf;

if bb==B
    disp(['Stopped at max number of iterations. ',...
        'Continuation criteria not met.']);
end

disp(['Finished singular vector ',num2str(ff),' of ',num2str(pfs)]);
end

%%%END ANALYTIC CONTINUATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for dd = 1:numel(ap);    % Reconstruct one distance at a time.

a = ap(dd); % Choose current reconstruction distance.
dr = rh-a;  % Distance between reconstruction and ref. mic. array.

for ff = 1:pfs
%CREATE PROPAGATOR
clear Gp Gur Guz;
for mm = 1:numel(mp); %Williams1999,(5.29)
```

```matlab
    %%%PRESSURE
    Gp(:,mm) = besselh(np,1,krmp(mm)*a)./besselh(np,1,krmp(mm)*rh);
    %%%VELOCITY
    if np==0
    Gur(:,mm) = (-1i*krmp(mm)/(rho*c*k))*...
        (-besselh(1,1,krmp(mm)*a)/besselh(np,1,krmp(mm)*rh));
    %Zcomponent of Williams eq 4.67pg132
    Guz(:,mm) = (kzmp(mm)/(rho*c*k))*...
        (besselh(np,1,krmp(mm)*a)./besselh(np,1,krmp(mm)*rh));
    else
    disp('Redefine velocity propagator for np~=0.');
    return
    end
end


%%%PROPAGATE CONTINUED SINGULAR VECTOR
Pr = Gp.*Pfall(:,ff).';
Urr = Gur.*Pfall(:,ff).';    %Velocity_reconstructed_radialComponent
Urz = Guz.*Pfall(:,ff).';    %Velocity_reconstructed_axialComponent

%TAKE INVERSE FFT TO GET RECONSTRUCTED SINGULAR VECTOR
pfbig(:,ff) = (Fzinv*(Fphiinv.'*Pfall(:,ff).').');
pa(:,ff) = (Fzinv.'*(Fphiinv.'*Pr).');   % Keep track of all the
uar(:,ff) = (Fzinv.'*(Fphiinv.'*Urr).');% reconstructed singular vectors to
uaz(:,ff) = (Fzinv.'*(Fphiinv.'*Urz).');% be added in the next step.

end
%%%ADD UP THE RECONSTRUCTED SINGULAR VECTORS
pfbigtemp = zeros(size(pfbig(:,1)));
prtemp = zeros(size(pa(:,1)));
Intrtemp = zeros(size(uar(:,1)));
Intztemp = zeros(size(uaz(:,1)));
for ff = 1:pfs
pfbigtemp = pfbigtemp+abs(pfbig(:,ff)).^2;
prtemp = prtemp+abs(pa(:,ff)).^2;
Intrtemp(:,ff) = real(pa(:,ff).*conj(uar(:,ff)));
Intztemp(:,ff) = real(pa(:,ff).*conj(uaz(:,ff)));
end
pfall(:,dd) = sqrt(pfbigtemp);
pr(:,dd) = sqrt(prtemp);
Intr(:,dd) = sum(Intrtemp,2);
Intz(:,dd) = sum(Intztemp,2);
disp(' ');
disp(['Finished reconstruction distance ',num2str(dd),' of ',...
    num2str(numel(ap))]);
disp(' ');

end

toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SAVE VARIABLES
if Savevar==1;
    cd(figdir)
    savename = [savbasename,num2str(foi),'Hz_Cz',num2str(Cz),'_xout',...
        num2str(xprop),'m_xin',num2str(xin),'m.mat'];
```

```matlab
        save(savename);
        cd(mdir)
    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

set(0,'DefaultAxesFontName','Times New Roman');
set(0,'DefaultAxesFontSize',14);
set(0,'DefaultTextFontSize',14);
set(0,'DefaultAxesLineWidth',2);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',10);
set(0,'DefaultFigureUnits','inches');
set(0,'DefaultFigurePosition',[1 1 6 4]);

%MAKE PLOTS OF RECONSTRUCTION AND ORIGINAL SVD

%PLOT INTENSITY RESULT
% figure;
% [f22,map,alphajet] =imread('F22_transparent.png');%Jetw/transp.background
% hold on;
% sf = 0.3048;
% J = image([-54 8]*sf,[20 -24]*sf,f22,'alphadata',alphajet);
% A = imread('F22_Image.jpg');       %For jet image with white background
% hold on;
% sf = 0.3048;
% J = image([-54 8]*sf,[20 -24]*sf,A);


figure;
scaling = 4000;
I_mag = sqrt(Intz.^2+Intr.^2);
I_mag_scaled = 10*log10(I_mag*scaling);
ang = atan2(Intr,Intz);
Intz_scaled = I_mag_scaled.*cos(ang);
Intr_scaled = I_mag_scaled.*sin(ang);
quiver(ZZ(2:2:end),Xa(2:2:end),...
    Intz_scaled(2:2:end,2:2:end).',Intr_scaled(2:2:end,2:2:end).',1,...
    '-','linewidth',1.5,'color',[255,105,180]/256);
xlabel('Z (m)');
ylabel('X (m)');
% title(['Reconstructed intensity, f = ',num2str(foi),' Hz']);
axis([-3 30 0 28]);
grid on;
drawnow;
hold on;
figure; %comment out this line to plot the quiver on the pressures
%PLOT CYL NAH RESULT
[ZZplot,aaplot]=meshgrid(ZZ,fliplr(Xa));
up = find(ZZplot(1,:)>-4,1,'first');%index of first pt just pos of z=-3.
down = find(ZZplot(1,:)>30,1,'first');%index of first pt just pos of z=30.
% [ZZbplot,aabplot]=meshgrid(ZZ(Cz+1:end-Cz/2),fliplr(ap));
Prplot = abs(flipud(pr.'));
Pplota = 20*log10(Prplot/2e-5);
% Pplotb = 20*log10(abs(flipud(pr(1+Cz:end-Cz/2,:).'))/2e-5);
Caxis = [max(max(Pplota))-35 max(max(Pplota))];
```

```matlab
% Caxis = [103 138];
% figure;
% subplot 211;
%PLOT RECONSTRUCTED PRESSURE
pcolor(ZZplot(:,up:down),aaplot(:,up:down),Pplota(:,up:down));
% pcolor(ZZplot,aaplot,Pplota);
hold on;
shading interp;
%MICS AND SHEAR LAYER
plot(Z,X,'k.',...
     [0,Z(end)],[0,Z(end)*tan(8.53*pi/180)],'k--');
% plot([ZZ(1+Cz) ZZ(1+Cz)],[ap(1) ap(end)],'k--',...
%      [ZZ(end-Cz) ZZ(end-Cz)],[ap(1) ap(end)],'k--',...
%      [ZZ(1+Cz) ZZ(end-Cz)],[ap(2*xin+1) ap(2*xin+1)],'k--'); hold off;
colormap jet;
h = colorbar;
caxis(Caxis);
xlabel('Z (m)');
ylabel('X (m)');
ylabel(h,'SPL (dB re 20\muPa)');
axis([-3 30 0 28]);

% %%%Plot numerical benchmark
% if sub.numerical
% figure;
% pbench = flipud(squeeze(sqrt(mean(p_bench.^2,1)))');
% PBENCH = 20*log10(abs(pbench)/2e-5);
% pcolor(ZZplot,aaplot,PBENCH);
% shading interp;
% axis([-3 30 0 28]);
% CB = colorbar;
% hold on;
% plot(Z,X,'k.',...
%      [0,Z(end)],[0,Z(end)*tan(8.53*pi/180)],'k--');%MICS AND SHEAR LAYER
% xlabel('Z (m)');
% ylabel('X (m)');
% ylabel(CB,'SPL (dB re 20\muPa)');
% title('numerical benchmark');
% caxis(Caxis);
% end

% %%%To plot interpolated num pressure along shear layer
% %%%Interpolate the reconstructed pressures to get the reconstructed
% %%%pressures along the shear layer.
% Xsh = (Zint(6):abs(Zint(2)-Zint(1)):abs(Zint(2)-
Zint(1))*50)*tan(8.53*pi/180);
% Zsh = ZZ(106:156);
% Psh = interp1(Xa,flipud(Pplota),Xsh,'spline'); %interp rcs pressures
% if sub.numerical
% Psh_bench = interp1(Xa,flipud(PBENCH),Xsh,'spline'); %interp bench
pressures
% end
% figure;
% % subplot 211;
% plot(Zsh,20*log10(pfall(106:156,1)/2e-5));
% hold on;
```

```matlab
% plot(Zsh,diag(Psh(:,106:156)),'r--');
% legend('Continuation (info only)','Reconstruction','location','northeast');
% if sub.numerical
% hold on;
% plot(Zsh,diag(Psh_bench(:,106:156)),'k');
% legend('continuation (info
only)','reconstruction','benchmark','location','south');
% end
% axis tight;
% ylim([105 155]);
% xlabel('Z (m)'); ylabel('SPL (dB re20\muPa)');
% grid on;
% % if sub.numerical
% % title(['f=',num2str(foi),', from numerical data, along shear layer']);
% % else
% % title(['f=',num2str(foi),', from actual data, along shear layer']);
% % end

% %%%Plot difference between benchmark and reconstruction
% if sub.numerical
% figure;
% set(gcf,'OuterPosition',[1 1 6 3]);
% plot(Zsh,diag(Psh(:,106:156))-diag(Psh_bench(:,106:156)));
% axis([Zsh(1) Zsh(end) -5 5]);
% xlabel('Z (m)'); ylabel('dB');
% grid on;
% % grid minor;
% set(gca,'YMinorGrid','on');
% end

% set(gcf,'paperpositionmode','auto');
% print('-dtiff','-r400','NumData_210Hz_shearlayerLinePlot_20120601')


% %%%To plot the 75-ft arc
% cdtemp = cd;
% cd('W:\NAH\CylindricalNAH');
% parameter.foi = foi;
% parameter.eng_cond = cond;
% parameter.df = 2.9;
% list = list_generator;
% file.complex_pres = ...
%     'Z:\Students\Alan Wall\HollomanF22_2009\ComplexPressures_v';
% load('Geometry\F22Holloman2009Geometry3.mat');
% [Zarc,Xarc,Yarc,parc] = ...
%     read_pressures(4,parameter,list,sub,file,F22Holloman2009Geometry);
% cd(cdtemp);
% if sub.numerical %Generate numerical meas along bottom row of arc
%     clear parc
%     arcmeas=1;
%     [Yarc1,~,parc,parc_noimage,blocksarc] =...
%     NumericalSource_20120321(foi,Zint,Xint,Xarc(end,:),...
%     Zarc(end,:),sub,arcmeas);
%     Pplotarc = 20*log10(sqrt(squeeze(mean(abs(parc).^2,1))))/2e-5;
% end
```

146

```matlab
% % figure;
% if ~sub.numerical
% Pplotarc = 20*log10(mean(abs(parc),3)/2e-5);
% surf(Zarc,Xarc,Yarc,Pplotarc);
% hold on;
% shading interp;
% % caxis(Caxis);
% zlabel('Y (m)');
% view([-24 46]);
% box off;
% grid on; hold on;
% %Plot guide lines to locate arc in space
% plot3([Zarc(1,end),Zarc(1,end)],...
%     [Xarc(1,end),Xarc(1,end)],[0,Yarc(end,1)],'k--');
% plot3(Zarc(1,:),Xarc(1,:),zeros(size(Yarc(1,:))),'k--');
% plot3([Zarc(1,1),Zarc(1,1)],[Xarc(1,1),Xarc(1,1)],...
%     [0,Yarc(end,1)],'k--');
% plot3([Zarc(1,1),Zarc(1,1)],[Xarc(1,1),aaplot(1,1)],...
%     [Yarc(end,1),Yarc(end,1)],'k--');
% plot3([ZZplot(1,up),Zarc(1,1)],[aaplot(1,1),aaplot(1,1)],...
%     [Yarc(end,1),Yarc(end,1)],'k--');
%
% else
% plot(Zarc(end,:),diag(Pplotarc).');
% end


% %%%To interpolate reconstruction and compare with arc
% %%%Be sure to previously load the arc data
% [ZZarc,XXarc] = meshgrid(Zarc(1,:),Xarc(1,:));
% Pintarc = interp2(ZZplot,aaplot,Pplota,ZZarc,XXarc,'spline');
% figure; pcolor(ZZarc,XXarc,Pintarc);
% figure;
% subplot 211;
% if ~sub.numerical
% plot(Zarc(1,:),Pplotarc(end,:),'b-');%bottom row of arc
% else
% plot(Zarc(end,:),diag(Pplotarc).');
% end
% axis tight; ylim([100 140]);
% hold on;
% plot(Zarc(1,:),diag(Pintarc),'g--');%cylNAH results below arc
% xlabel('Z (m)'); ylabel('SPL (dB re 20\muPa)'); grid on;
% legend('Bottom row of arc','cyl NAH below arc','location','northwest');
% subplot 212;
% if~sub.numerical
% plot(Zarc(1,:),diag(Pintarc).'-Pplotarc(end,:));
% else
% plot(Zarc(end,:),diag(Pintarc).'-diag(Pplotarc).');
% end
% xlabel('Z (m)'); ylabel('dB');
% axis tight;
% ylim([-5 5]);
% grid on;
% set(gca,'YMinorGrid','on');
```

```matlab
% %PLOT SINGULAR VECTORS
% [SingV,SingZ]=meshgrid(1:numel(Z),Z(1:N));
% figure;
% pcolor(SingV,SingZ,20*log10(abs(S)/20e-6))
% h= colorbar;
% ylabel(h,'dB re 20 \muPa');
% caxis([20*log10(max(max(abs(S)))/20e-6)-50.5 20*...
%     log10(max(max(abs(S)))/20e-6)]);
% xlabel('Singular vector #');
% ylabel('Z (m)'); grid on; axis('xy');
% colormap(gray);
% title({'SVD of Reference Array';['Plane ',num2str(plane),', ',...
%     CONDS{cond},', f = ',num2str(foi),' Hz']});


% %PLOT SINGULAR VALUES
% figure;
% stem(diag(10*log10(Lamb))); xlim([0 N+.5]); hold on;
% plot([pfs+.5 pfs+.5],[min(diag(10*log10(Lamb))) max(diag(10*...
%   log10(Lamb)))],'k--'); hold off;
% ylabel('10\cdotlog_{10}(\lambda)  (dB re 1)');
% xlabel('Singular Value');
% title([num2str(pfs),' singular values within ',num2str(pfdB),...
%   ' dB of largest']);
% if savefig==1
%     cd(figdir)
%     set(gcf,'paperpositionmode','auto');
%     print('-dtiffn','-r300',[savbasename,int2str(foi),'Hz_Cz',...
%         num2str(Cz),'_xprop',num2str(xprop),'_',num2str(fn,'%02i'),...
%         '_SingVals_color'])
%     fn = fn+1;
%     cd(mdir)
% end
```

mdpCYL

```matlab
function J = mdpCYL(alph,L,Fphiinv,Fzinv,P,p,Nz,sig,k,kzmp)

Falpha1 = (abs(L).^2./...
    (abs(L).^2+alph*(alph./(alph+abs(L).^2)).^2));

    for cc = 1:Nz % Once the regularization alpha is computer, the elements
                  % within the radiation circle |kzmp|<|k| are replaced with
                  % unity. See Williams2001(discussion above Eq.33).
        if sqrt(kzmp(cc)^2) < k
            Falpha1(cc) = 1;
        end
    end
ptemp = (Fzinv.'*(Fphiinv.'*(Falpha1.*P)).');
J = (norm(ptemp-p)/sqrt(numel(p))-sig)^2;
```

NumericalSource_20110606

```matlab
function [Y,p,p_bench,p_noimage,blocks] =
NumericalSource_20110606(foi,Z,X,Xa,ZZ)
%This numerical source propagates from a directional line array along the
%jet centerline to the array of ground reference microphones.
%
%Inputs:
%foi is the frequency of interest
%Zint are the Z locations of the reference mics
%X are the X locations of the reference mics
%Xa is the reconstruction locations in X
%ZZ is the continued locations in Z
%
%Outputs:
%Z,Y,X are the locations of the reference microphones
%p is pressures at the ref mic locations using an image source
%p_bench is the benchmark pressure at every reconstruction distance using
%    an image source
%p_noimage is pressures at the ref mic locations NOT using an image source
%blocks is the number of blocks = 151
%The size of p and p_noimage is [blocks,num mics]=[151,50]
%The size of p_bench is [blocks,Z,Xa]

%DEFINE CONSTANTS
rho = 1.21; %Air desnity, kg/m^3.
c = 343; %Speed of sound, m/s;
k = 2*pi*foi/c; %Wavenumber, m^-1;

%GENERATE DIRECTIONAL SOURCE NUMERICALLY
Nsources = 20;%number of partially correlated numerical sources
dsource = .4; %m distance between numerical sources
sourcedirection = 130;%[deg] sourcedirection angle meas from DOWNstream
phi = 2*pi*foi*dsource*...
    (0:Nsources-1)*cosd(sourcedirection)/c;%source phases
Q = 1;
xs = 0;
ys = 6.25*0.3048;
ysi = -ys;
zs = 2:dsource:2+(Nsources-1)*dsource;
SNRdB = 60; % dB Signal to noise ratio
SNR = 10^(SNRdB/20);

for blcks = 1:151;
    for aa = 1:numel(Xa);
    for nn = 1:Nsources
        R(:,nn) = sqrt((Z(1,:)-zs(nn)).^2+...
            (X(1,:)-xs).^2+(0-ys).^2);
        Ri(:,nn) =sqrt((Z(1,:)-zs(nn)).^2+...
```

```matlab
                    (X(1,:)-xs).^2+(0-ysi).^2);
            Rbench(:,nn) = sqrt((ZZ-zs(nn)).^2+...
                (Xa(aa)-xs).^2+(0-ys).^2);
            Ribench(:,nn) = sqrt((ZZ-zs(nn)).^2+...
                (Xa(aa)-xs).^2+(0-ysi).^2);
            p_num_all(:,nn) = ...
                (1i*rho*c*k*Q/4/pi./R(:,nn).*exp(-1i*k.*R(:,nn))+...
                1i*rho*c*k*Q/4/pi./Ri(:,nn).*exp(-1i*k.*Ri(:,nn)))*...
                exp(1i*phi(nn));
            p_num_noimage(:,nn) = ...
                (1i*rho*c*k*Q/4/pi./R(:,nn).*exp(-1i*k.*R(:,nn)))*...
                exp(1i*phi(nn));
            p_num_bench(:,nn) = ...
                (1i*rho*c*k*Q/4/pi./Rbench(:,nn).*exp(-1i*k.*Rbench(:,nn))+...
                1i*rho*c*k*Q/4/pi./Ribench(:,nn).*exp(-1i*k.*Ribench(:,nn)))*...
                exp(1i*phi(nn));
            noise_num_all(:,nn) = ...
                sqrt(norm(p_num_all(:,nn))/...
                SNR/sqrt(numel(p_num_all(:,nn))))*...
                sqrt(1/2)*(randn(size(p_num_all(:,nn)))+...
                1i*randn(size(p_num_all(:,nn))));
            noise_num_noimage(:,nn) = ...
                sqrt(norm(p_num_noimage(:,nn))/...
                SNR/sqrt(numel(p_num_noimage(:,nn))))*...
                sqrt(1/2)*(randn(size(p_num_noimage(:,nn)))+...
                1i*randn(size(p_num_noimage(:,nn))));
            noise_num_bench(:,nn) = ...
                sqrt(norm(p_num_bench(:,nn))/...
                SNR/sqrt(numel(p_num_bench(:,nn))))*...
                sqrt(1/2)*(randn(size(p_num_bench(:,nn)))+...
                1i*randn(size(p_num_bench(:,nn))));


        p_num_noise_all(blcks,:,nn) = p_num_all(:,nn)+noise_num_all(:,nn);
        p_num_noise_noimage(blcks,:,nn) =
p_num_noimage(:,nn)+noise_num_noimage(:,nn);
        p_num_noise_bench(blcks,:,nn,aa) = p_num_bench(:,nn) +
noise_num_bench(:,nn);
    end
    end
end

Y = zeros(size(Z(1,:)));
p = sum(p_num_noise_all,3);
% p = (diag(W)*sum(p_num_noise_all,3).').';
p_noimage = sum(p_num_noise_noimage,3);
p_bench = squeeze(sum(p_num_noise_bench,3));
blocks = blcks;
```

8 References

[1] M. J. Lighthill, "On Sound Generated Aerodynamically. I. General Theory," Proc. of the Royal Soc. of London. A, Mathematical and Physical Sciences **211**, 564-587 (1952).

[2] H. V. Fuchs, "On the Application of Acoustic Mirror , Telescope and Polar Correlation Techniques to Jet Noise Source Location," J. Sound Vib. **58**, 117-126 (1978).

[3] J. Billingsley, and R. Kinns, "The acoustic telescope," J. Sound Vib. **48**, 485-510 (1976).

[4] J. Billingsley, "A comparison of the source location techniques of the acoustic telescope and polar correlation," J. Sound Vib. **61**, 419-425 (1978).

[5] M. J. Fisher, M. Harper-Bourne, and S. A. L. Glegg, "Jet engine noise source location: The polar correlation technique," J. Sound Vib. **51**, 23-54 (1977).

[6] B. D. Van Veen, and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," IEEE ASSP magazine **5**, 4-24 (1988).

[7] D. Papamoschou, P. J. Morris, and D. K. McLaughlin, "Beamformed flow-acoustic correlations in a supersonic jet," AIAA J. **48**, 2445-2453 (2010).

[8] W. M. Humphreys, W. W. Hunter, K. R. Meadows, and T. F. Brooks, "Design and use of microphone directional arrays for aeroacoustic measurements," in *AIAA 98-0471* (Reno, Nevada, 1998), pp. 98--0471.

[9] E. G. Williams, and J. D. Maynard, "Holographic Imaging without the Wavelength Resolution Limit," Phys. Rev. Lett. **45**, 554-557 (1980).

[10] M. Lee, and J. S. Bolton, "Source characterization of a subsonic jet by using near-field acoustical holography," J. Acoust. Soc. Am. **121**, 967-977 (2007).

[11] D. Long, J. Peters, and M. Anderson, "Evaluating turbofan exhaust noise and source characteristics from near field measurements," in *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference), May 11, 2009 - May 13, 2009* (American Institute of Aeronautics and Astronautics Inc., Miami, FL, United states, 2009).

[12] M. Lee, J. S. Bolton, and L. Mongeau, "Application of cylindrical near-field acoustical holography to the visualization of aeroacoustic sources," J. Acoust. Soc. Am. **114**, 842-858 (2003).

[13] M. D. Gardner, "Scan-based Near-Field Acoustical Holography on Partially Correlated Sources," Thesis, Brigham Young University, Provo, Utah, (2009).

[14] A. Michalke, "Some remarks on source coherence affecting jet noise," J. Sound Vib. **87**, 1-17 (1983).

[15] J. Hald, "Basic theory and properties of statistically optimized near-field acoustical holography," J. Acoust. Soc. Am. **125**, 2105-2120 (2009).

[16] T. Semenova, and S. F. Wu, "On the choice of expansion functions in the Helmholtz equation least-squares method," J. Acoust. Soc. Am. **117**, 701-710 (2005).

[17] E. G. Williams, *Fourier Acoustics Sound Radiation and Nearfield Acoustical Holography*, (Academic Press, London, 1999).

[18] K. B. Ginn, and J. Hald, "STSF-practical instrumentation and applications," Technical review - Bruel and Kjaer English ed., 1-27 (1989).

[19] J. Hald, "STSF - a unique technique for scan-based Near-field Acoustic Holography without restrictions on coherence," Technical review - Bruel and Kjaer English ed., 1-50 (1989).

[20] M. Lee, and J. Stuart Bolton, "Scan-based near-field acoustical holography and partial field decomposition in the presence of noise and source level variation," J. Acoust. Soc. Am. **119**, 382-393 (2006).

[21] K. L. Gee, V. W. Sparrow, M. M. James, J. M. Downing, C. M. Hobbs, T. B. Gabrielson, and A. A. Atchley, "The role of nonlinear effects in the propagation of noise from high-power jet aircraft," J. Acoust. Soc. Am. **123**, 4082-4093 (2008).

[22] E. G. Williams, "Continuation of acoustic near-fields," J. Acoust. Soc. Am. **113**, 1273-1281 (2003).

[23] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," Proc. IEEE **66**, 51-83 (1978).

[24] E. G. Williams, "Regularization methods for near-field acoustical holography," J. Acoust. Soc. Am. **110**, 1976-1988 (2001).

[25] P. C. Hansen, *Rank-deficient and discrete ill-posed problems*, (SIAM, Philadelphia, Pennsylvania, 1998).

[26] Y. Kim, and P. A. Nelson, "Optimal regularisation for acoustic source reconstruction by inverse methods," J. Sound Vib. **275**, 463-487 (2004).

[27] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," Technometrics **21**, 215-223 (1979).

[28] K. L. Gee, J. H. Giraud, J. D. Blotter, and S. D. Sommerfeldt, "Near-field vector intensity measurements of a small solid rocket motor," J. Acoust. Soc. Am. **128**, EL69-EL74 (2010).

[29] M. Lee, and J. S. Bolton, "A one-step patch near-field acoustical holography procedure," J. Acoust. Soc. Am. **122**, 1662-1670 (2007).

[30] A. T. Wall, K. L. Gee, M. M. James, and M. D. Gardner, "Application of near-field acoustical holography to high-performance jet aircraft noise," J. Acoust. Soc. Am. **127**, 1879 (2010).

[31] A. T. Wall, K. L. Gee, M. M. James, K. A. Bradley, S. A. McInerny, and T. B. Neilsen, "Near-field noise measurements of a high-performance military jet aircraft," Noise Control Eng J (accepted 2012).

[32] M. R. Bai, "Application of BEM (boundary element method)-based acoustic holography to radiation analysis of sound sources with arbitrarily shaped geometries," J. Acoust. Soc. Am. **92**, 533-549 (1992).

[33] M. Y. Yang, H. Vold, and P. Shah, "Measurement and Propagation of Supersonic Aeroacoustic Noise Sources Using Continuous Scanning Measurement Technologies and the Fast Multipole Boundary Element Method," Proceedings of Meetings on Acoustics **14**, Acoust. Soc. of America (2011).

[34] H.-S. Kwon, and J. S. Bolton, "Partial field decomposition in nearfield acoustical holography by the use of singular value decomposition and partial coherence procedures," in *Proceedings of the 1998 National Conference on Noise Control Engineering. Part 1 (of 3), April 5, 1998 - April 9, 1998* (Inst Noise Control Eng, Ypsilanti, MI, USA, 1998), pp. 649-654.

[35] H.-S. Kwon, Y.-J. Kim, and J. S. Bolton, "Compensation for source nonstationarity in multireference, scan-based near-field acoustical holography," J. Acoust. Soc. Am. **113**, 360-368 (2003).

[36] E. G. Williams, B. H. Houston, and P. C. Herdic, "Fast fourier transform and singular value decomposition formulations for patch nearfield acoustical holography," J. Acoust. Soc. Am. **114**, 1322-1333 (2003).

[37] A. D. Pierce, *Acoustics*, (Acoustical Society of America, Melville, 1989).

[38] J. Laufer, R. Schlinker, and R. E. Kaplan, "Experiments on Supersonic Jet Noise," AIAA J **14**, 489-497 (1976).

[39] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*, 4th Ed., (John Wiley & Sons, Inc.).

[40] B. E. Anderson, "Grating lobe reduction in transducer arrays through structural filtering of supercritical plates," Dissertation, The Pennsylvania State University, (2006).

[41] P. A. Nelson, and S. J. Elliott, *Active Control of Sound*, (Academic Press, London, 1992).

[42] J. Tichy, and S. D. Sommerfeldt, "Active Noise and Vibration Control," (The Pennsylvania State University, 1993).

[43] G. Rasmussen, "Measurement of vector fields," in *Proceedings of the 2nd International Congress on Acoustic Intensity* (1985), pp. 53-58.

[44] S. M. Jaeger, and C. S. Allen, "Two-dimensional sound intensity analysis of jet noise," in *15th AIAA Aeroacoustics Conference* (Long Beach, CA, 1993).

[45] F. J. Fahy, *Sound Intensity*, 2nd Edition, (Chapman & Hall, London, 1995).

[46] J.-C. Pascal, and J.-F. Li, "A systematic method to obtain 3D finite-difference formulations for acoustic intensity and other energy quantities," J. Sound Vib. **310**, 1093-1111 (2008).

[47] D. C. Thomas, "Theory and Estimation of Acoustic Intensity and Energy Density," Thesis, Brigham Young University, Provo, Utah, (2008).

[48] T. K. Moon, and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, (Prentice Hall, Upper Saddle River, New Jersey, 2000).

[49] R. H. Schlinker, S. A. Liljenberg, D. R. Polak, K. A. Post, C. T. Chipman, and A. M. Stern, "Supersonic jet noise source characteristics and propagation: Engine and model scale," AIAA J., 2007-3623 (2007).

[50] C. K. W. Tam, K. Viswanathan, K. K. Ahuja, and J. Panda, "The sources of jet noise: Experimental evidence," J Fluid Mech **615**, 253-292 (2008).

[51] J. Morgan, K. L. Gee, T. Neilsen, and A. T. Wall, "A simple-source model of military jet aircraft noise," J. Acoust. Soc. Am. **129**, 2442-2442 (2011).

[52] K. L. Gee, T. B. Gabrielson, A. A. Atchley, and V. W. Sparrow, "Preliminary analysis of nonlinearity in military jet aircraft noise propagation," AIAA J **43**, 1398-1401 (2005).