SPACE-FREQUENCY SAMPLING CRITERIA
FOR ELECTROMAGNETIC SCATTERING OF A FINITE OBJECT

By
Fredric Y. S. Fok

The Ohio State University
ElectroScience Laboratory
Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report No. 714190-11
Contract No. N00014-82-K-0037
August 1985

Department of the Navy
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited
NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.
This investigation concerns the sampling criteria in the wave number space for generating the spatial impulse response of a finite target. The impulse response of a finite target is important for target identification and imaging. The other purpose of this report is in the management of large amounts of data for potential application in the presentation of scattered field data and construction of such images. For clarity, monostatic impulse responses of up to two dimensions are considered.

A proper choice of canonical confinement for the target in space can greatly reduce the number of samples required to sufficiently characterize the target's spatial impulse response. One dimensional impulse responses generated using two basic types of confinement: 1) isotropic and 2) parallelepiped, are compared. The two approaches show competitive reconstructed results. Though a sampling lattice may be more efficient in the sense of a reduced number of measurement points, it may be less effective when digital processing is involved. Specifically, the time consuming interpolation step is required to put data presented in other types of sampling lattice into the cubic type.

Two dimensional impulse responses reconstructed from cubic sampled data are compared with those using Mensa et al.'s method. Two dimensional impulse responses obtained also indicate good potentials for image reconstruction via the spatial impulse response.
# TABLE OF CONTENTS

| LIST OF TABLES | .......... | iv |
| LIST OF ILLUSTRATIONS | .......... | v |

## CHAPTER

| I. INTRODUCTION | .......... | 1 |
| II. THEORY | .......... | 5 |
| III. PRACTICAL CONSIDERATIONS | .......... | 28 |
| IV. ONE DIMENSIONAL IMPULSE RESPONSES | .......... | 36 |
| V. TWO DIMENSIONAL IMPULSE RESPONSES | .......... | 81 |
| VI. CONCLUSIONS | .......... | 105 |
| VII. RECOMMENDATIONS | .......... | 113 |

## APPENDIXES

| A. THE DERIVATION OF EQUATION (2-13) | .......... | 117 |
| B. PROGRAMS DEVELOPED | .......... | 131 |

## REFERENCES

| .......... | 191 |
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>Efficiency comparison on 3 objects using 3 types of containment cells</td>
<td>43</td>
</tr>
<tr>
<td>4-2</td>
<td>The number of sampling points for 3 types of sampling on 3 objects over various frequency ranges</td>
<td>44</td>
</tr>
<tr>
<td>5-1</td>
<td>An example: Time estimation</td>
<td>91</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Block diagram depicting system analogy</td>
<td>6</td>
</tr>
<tr>
<td>2-2</td>
<td>Periodicity of the canonical unit: parallelogram (shaded) with its images (dotted lined) defined by $\tilde{v}_1$ and $\tilde{v}_2$</td>
<td>14</td>
</tr>
<tr>
<td>2-3</td>
<td>New $\tilde{v}_1$ and $\tilde{v}_2$ defining a similar periodicity as Figure 2-2 except for the extra guard band</td>
<td>15</td>
</tr>
<tr>
<td>2-4</td>
<td>New $\tilde{v}_1$ and $\tilde{v}_2$ defining a similar periodicity as Figure 2-2 except the images are rotated around the unit</td>
<td>16</td>
</tr>
<tr>
<td>2-5</td>
<td>A choice of periodicity for the parallelogrammatic containment unit</td>
<td>19</td>
</tr>
<tr>
<td>2-6</td>
<td>Sampling lattice defined by the choice of periodicity in Figure 2-5</td>
<td>20</td>
</tr>
<tr>
<td>2-7</td>
<td>A choice of periodicity for the circular containment cell</td>
<td>22</td>
</tr>
<tr>
<td>2-8</td>
<td>Sampling lattice defined by the choice of periodicity in Figure 2-7</td>
<td>23</td>
</tr>
<tr>
<td>3-1</td>
<td>Frequency information available</td>
<td>29</td>
</tr>
<tr>
<td>3-2</td>
<td>Frequency information available after assumption is made about the negative frequency samples</td>
<td>32</td>
</tr>
<tr>
<td>4-1</td>
<td>Cosine tapering weighting (Equation 4-2 with $N_1=16$)</td>
<td>45</td>
</tr>
<tr>
<td>4-2</td>
<td>Impulse response of a 6&quot; metallic sphere with frequency samples taken every 60 Mhz over the range of 0 to 12 Ghz</td>
<td>46</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>4-3</td>
<td>Impulse response of a 6&quot; metallic sphere with frequency samples taken every 125 MHz over the range of 0 to 12 GHz and interpolated using Equation (4-1)</td>
<td>47</td>
</tr>
<tr>
<td>4-3a</td>
<td>Error of Figure 4-3 from Figure 4-2 (Magnification=10X)</td>
<td>48</td>
</tr>
<tr>
<td>4-4</td>
<td>Similar description as Figure 4-3, except frequency samples are taken every 250 MHz</td>
<td>49</td>
</tr>
<tr>
<td>4-4a</td>
<td>Error of Figure 4-4 from Figure 4-2 (Magnification=10X)</td>
<td>50</td>
</tr>
<tr>
<td>4-5</td>
<td>Similar description as Figure 4-3, except frequency samples are taken every 500 MHz</td>
<td>51</td>
</tr>
<tr>
<td>4-5a</td>
<td>Error of Figure 4-5 from Figure 4-2 (No magnification)</td>
<td>52</td>
</tr>
<tr>
<td>4-6</td>
<td>Similar description as Figure 4-3, except frequency samples are taken every 750 MHz</td>
<td>53</td>
</tr>
<tr>
<td>4-6a</td>
<td>Error of Figure 4-6 from Figure 4-2 (No magnification)</td>
<td>54</td>
</tr>
<tr>
<td>4-7</td>
<td>Impulse response of a 6&quot; metallic sphere at 0° aspect angle with 1-D frequency response interpolated from 2-D isotropic lattice samples taken with a safety factor of 1 over the range of 0 to 12 GHz using Equations (4-3) and (2-13)</td>
<td>55</td>
</tr>
<tr>
<td>4-7a</td>
<td>Error of Figure 4-7 from Figure 4-2</td>
<td>56</td>
</tr>
<tr>
<td>4-8</td>
<td>Similar description as Figure 4-7, except the aspect angle is 72°</td>
<td>57</td>
</tr>
<tr>
<td>4-8a</td>
<td>Error of Figure 4-8 from Figure 4-2</td>
<td>58</td>
</tr>
<tr>
<td>4-9</td>
<td>Similar description as Figure 4-7, except the aspect angle is 1.44°</td>
<td>59</td>
</tr>
<tr>
<td>4-9a</td>
<td>Error of Figure 4-9 from Figure 4-2</td>
<td>60</td>
</tr>
<tr>
<td>4-10</td>
<td>Similar description as Figure 4-7, except the aspect angle is 30°</td>
<td>61</td>
</tr>
</tbody>
</table>
### LIST OF ILLUSTRATIONS (CONTINUED)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-10a</td>
<td>Error of Figure 4-10 from Figure 4-2</td>
<td>62</td>
</tr>
<tr>
<td>4-11</td>
<td>Impulse response of a 6″ metallic sphere at 0° aspect angle with 1-D frequency response interpolated from 2-D cubic lattice samples taken with a safety factor of 1 over the range of 0 to 12 Ghz using Equations (4-3) and (2-11).</td>
<td>63</td>
</tr>
<tr>
<td>4-11a</td>
<td>Error of Figure 4-11 from Figure 4-2</td>
<td>64</td>
</tr>
<tr>
<td>4-12</td>
<td>Similar description as Figure 4-11, except the aspect angle is 1.19°</td>
<td>65</td>
</tr>
<tr>
<td>4-12a</td>
<td>Error of Figure 4-12 from Figure 4-2</td>
<td>66</td>
</tr>
<tr>
<td>4-13</td>
<td>Similar description as Figure 4-11, except the aspect angle is 2.39°</td>
<td>67</td>
</tr>
<tr>
<td>4-13a</td>
<td>Error of Figure 4-13 from Figure 4-2</td>
<td>68</td>
</tr>
<tr>
<td>4-14</td>
<td>Similar description as Figure 4-11, except the aspect angle is 45°</td>
<td>69</td>
</tr>
<tr>
<td>4-14a</td>
<td>Error of Figure 4-14 from Figure 4-2</td>
<td>70</td>
</tr>
<tr>
<td>4-15</td>
<td>Impulse response of a 6″ metallic sphere at 0° aspect angle with 1-D frequency response interpolated from 2-D isotropic lattice samples taken with a safety factor of 3 over the range of 0 to 12 Ghz using Equations (4-3) and (2-13)</td>
<td>71</td>
</tr>
<tr>
<td>4-15a</td>
<td>Error of Figure 4-15 from Figure 4-2 (Magnification=10X)</td>
<td>72</td>
</tr>
<tr>
<td>4-16</td>
<td>Similar description as Figure 4-15, except the aspect angle is 0.35°</td>
<td>73</td>
</tr>
<tr>
<td>4-16a</td>
<td>Error of Figure 4-16 from Figure 4-2 (Magnification=10X)</td>
<td>74</td>
</tr>
<tr>
<td>4-17</td>
<td>Similar description as Figure 4-15, except the aspect angle is 30°</td>
<td>75</td>
</tr>
<tr>
<td>4-17a</td>
<td>Error of Figure 4-17 from Figure 4-2 (Magnification=10X)</td>
<td>76</td>
</tr>
</tbody>
</table>
## LIST OF ILLUSTRATIONS (CONTINUED)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-18</td>
<td>Dimensional perspective among the sphere, cube and sphere cap cylinder used in the example. (Diameter of sphere = 6 inches)</td>
<td>77</td>
</tr>
<tr>
<td>4-19</td>
<td>The number of samples specified by the different types of sampling lattices on a sphere cap cylinder described in Figure 4-18.</td>
<td>78</td>
</tr>
<tr>
<td>4-20</td>
<td>The number of samples specified by two types of sampling lattices on a 6&quot; sphere.</td>
<td>79</td>
</tr>
<tr>
<td>4-21</td>
<td>The number of samples specified by two types of sampling lattices on the cube described in Figure 4-18.</td>
<td>80</td>
</tr>
<tr>
<td>5-1</td>
<td>2-D impulse response of a finite metallic circular cylinder with 6&quot; in length and 3&quot; in diameter. Frequency samples are taken at the cubic sampling lattice over the range of 2 to 5 GHz and Fourier transformed discretely into the spatial domain. Data are taken over the half plane: $0 &lt; \theta &lt; \pi$.</td>
<td>92</td>
</tr>
<tr>
<td>5-1a</td>
<td>Contour plot of Figure 5-1.</td>
<td>93</td>
</tr>
<tr>
<td>5-2</td>
<td>2-D impulse response of a finite metallic circular cylinder with 6&quot; in length and 3&quot; in diameter. Frequency samples are taken over 6 rings: 2.5, 3.5, 4, 4.5, 5 GHz and transformed into the spatial domain via Mensa et al.'s method. Data are taken over the half plane: $0 &lt; \theta &lt; \pi$.</td>
<td>94</td>
</tr>
<tr>
<td>5-2a</td>
<td>Contour plot of Figure 5-2.</td>
<td>95</td>
</tr>
<tr>
<td>5-3</td>
<td>Similar description as Figure 5-2 with 10 additional frequency rings: 5.5, 6.5, 7.5, 8, 8.5, 9.5, 10 GHz.</td>
<td>96</td>
</tr>
<tr>
<td>5-3a</td>
<td>Contour plot of Figure 5-3.</td>
<td>97</td>
</tr>
<tr>
<td>5-4</td>
<td>Similar description as Figure 5-3, except the data are taken over the full 360° plane: $0 &lt; \theta &lt; 2\pi$.</td>
<td>98</td>
</tr>
<tr>
<td>5-4a</td>
<td>Contour plot of Figure 5-4.</td>
<td>99</td>
</tr>
<tr>
<td>5-5</td>
<td>The plot of the absolute value of the amplitude in Figure 5-4.</td>
<td>100</td>
</tr>
<tr>
<td>5-5a</td>
<td>Contour plot of Figure 5-5.</td>
<td>101</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5-6</td>
<td>2-D impulse response of a 6&quot; metallic sphere with its centre situated at (3&quot;,0). Frequency samples are taken at the cubic sampling lattice over the range of 0 to 12 GHz. The data are cosine tapering weighted before Fourier transformed discretely into the spatial domain. Data are taken over the half plane: $\pi &lt; \theta &lt; 2\pi$.</td>
<td>102</td>
</tr>
<tr>
<td>5-6a</td>
<td>Contour plot of Figure 5-6.</td>
<td>103</td>
</tr>
<tr>
<td>5-7</td>
<td>Path length of a creeping wave on a metallic sphere.</td>
<td>104</td>
</tr>
<tr>
<td>A-1</td>
<td>Hexagonal integration limit.</td>
<td>118</td>
</tr>
<tr>
<td>B-1</td>
<td>Flow chart showing the relationship among the written programs.</td>
<td>132</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

Discrete data sampling at the Nyquist rate (or better) for the reconstruction of continuous waveforms is well known. However, the application of the sampling theorem [1,2] is often limited to one dimensional problems. This report will explore an application of the N dimensional sampling theorem to inverse scattering. More specifically, the application is on the reconstruction of the spatial impulse response of a finite object using different sampling criteria. The different aspects of the N dimensional sampling theorem are investigated to produce efficient sampling criteria in the wave number space such that the spatial impulse response of a finite object is sufficiently characterized. In addition, the impulse response concept is extended to create possibly an image of the object.

The impulse response [3,4] is important both in target identification and imaging. The impulse response concept, as most people understand, is a far field one dimensional time response concept. This time response waveform when applied in scattering can be plotted on a distance abscissa after the factor due to the speed of the wave is
accounted for. If one can obtain the frequency responses of a finite
object for all aspect angles over the $4\pi$ solid angle, the inverse
Fourier transform of the total frequency response into the spatial
domain forms an image which is called the spatial impulse response.

i.e., the impulse response of a finite object at $\tilde{x}$ is

$$f(\tilde{x}) \equiv \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} F(\vec{k}) e^{j(\vec{k} \cdot \vec{x})} d\vec{k} \tag{1-1}$$

where $F(\vec{k})$ = the far field frequency response of a finite
object at $\vec{k}$ in the space-frequency domain

$$\tilde{x} = (x_1, x_2, x_3)$$
$$\vec{k} = (k_1, k_2, k_3)$$
$$d\vec{k} = dk_1 \, dk_2 \, dk_3$$

The three dimensional function $f(\tilde{x})$ for all $\tilde{x}$'s in the $x$-space
constitutes an image called the spatial impulse response.

The spatial impulse response can be divided into two types - the
monostatic and the bistatic. The monostatic spatial impulse response
uses monostatic frequency responses of all aspect angles. The bistatic
spatial impulse response uses the bistatic frequency responses of all
receiving aspect angles. The spatial, two dimensional, or one
dimensional impulse responses discussed hereafter will all be referring
to the monostatic case unless otherwise stated. The data used are also
monostatic data. However, the theory to be discussed is also applicable
to bistatic cases. Furthermore, only two dimensional frequency data are
used in the discussion, so this report will emphasize the two
dimensional impulse response only.

In the introduction, the theme and purpose of this report are
derined. A brief description on each of the following chapters is also
cluded. In Chapter II, the impulse response, sampling theorem and
their relationship are discussed. The basics of the one dimensional
impulse response concept and the one dimensional sampling theorem are
first reviewed. Using the idea of the settling time, one can relate the
impulse response to the sampling theorem. Then the one dimensional
impulse response concept is extended to the three dimensional space.
Lewis and Bojarski's [5,6] work in the area is also briefly contrasted.
A decision rule based on Petersen and Middleton's [7] definition of
sampling efficiency is introduced so that one can decide on the more
efficient sampling grid. There is also a discussion on Petersen and
Middleton's N dimensional sampling theorem and its different forms which
depend on the different types of sampling techniques. Then Mensa
et al.'s [8] polar transformation and single frequency approach to two
dimensional Fourier transform is rederived. Their approach presents a
new perspective to the Nyquist sampling criteria.

Some practical aspects of the theory in Chapter II are considered
in Chapter III. Even though the discussion is in one dimension, it is
also applicable to two and three dimensional analyses. The real signal
requirement on Fourier transform is rederived. This helps to clarify
the problem of non-zero imaginary parts during data processing. Using
the sampling theorem interpolation scheme, the frequency bandwidth's
relationship to target size and spatial resolution is considered. A common problem that may be easily overlooked in data processing is the periodicity of the Fourier series representation to solve the Fourier integral. This is also restated in the last part of Chapter III.

In order to build some confidence in the different forms of Petersen and Middleton's sampling criteria, results of interpolating one dimensional impulse responses are presented in Chapter IV. The interpolation scheme used is the same as in the sampling theorem, except the infinite summation is a finite summation. The interpolated results using one dimensional sampled data appear first in the chapter to preview the interpolation via two dimensional sampled data. Later, an example is chosen to compare the efficiency of different sampling grids. Two dimensional impulse responses computed using discrete two dimensional Fourier transform are compared with results using Mensa et al.'s approach to Fourier transformation in Chapter V. The potential of using the spatial impulse response for target imaging is explored in the last part of the report.
CHAPTER II
THEORY

Some of the basic concepts on impulse response and one dimensional sampling theorem are individually reviewed. Then the two concepts are combined and extended to three dimensional space. The different aspects of N dimensional sampling theorem are introduced. Finally, Mensa et al.'s approach to two dimensional Fourier transform and sampling is also discussed.

If an impulsive electric field is incident on a target, the normalized far zone time domain scattering will be the impulse response of the target at that particular aspect angle and polarization. The approach to be used herein to obtain the impulse response is similar to deconvolution. First, the scattered field (a complex function of frequency) is divided by the spectrum of the incident wave. This result is then inverse Fourier transformed to produce the desired impulse response.
Let $f(t)$ be the input signal in time
$F(\omega)$ be the input frequency spectrum
$h(t)$ be the impulse response of the target
$H(\omega)$ be the frequency response of the target
$c(t)$ be the output signal in time
$C(\omega)$ be the output frequency spectrum

Figure 2-1. Block diagram depicting system analogy
By the convolution theorem of Fourier transform theory [2]

\[ C(\omega) = F(\omega)H(\omega) \quad (2-1) \]

\[ \Leftrightarrow \quad H(\omega) = \frac{C(\omega)}{F(\omega)} \quad (2-2) \]

Fourier transformed,

\[ h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{C(\omega)}{F(\omega)} e^{j\omega t} d\omega \quad (2-3) \]

According to Kennaugh and Moffatt [3], the impulse response will decay exponentially for large values of \( t - \frac{r}{c} \)

where \( t \) -time

\( r \) -distance between observation point and the origin

\( c \) -speed of light

Therefore, one can define a settling time when the impulse response has its amplitude embedded in the noise level. For all practical purposes, the settling time will be the end of the impulse response. Thus, the impulse response is a time limited signal. From the sampling theorem [1,2]: "A time limited signal can be reproduced from its discrete frequency values, if it is sampled over the complete frequency domain using the Nyquist rate." As a consequence, the impulse response may be reproduced by measuring its frequency spectrum at the Nyquist sampling rate (\( f_s \)). (i.e., \( f_s < 1/2T \); where the signal is limited in time to \( \pm T \))
Or, if

\[ f(t) = 0 \quad |t| > T \]

then

\[
F(\omega) = \sum_{n=-\infty}^{\infty} F(\frac{n\pi}{T}) \frac{\sin(\omega T - n\pi)}{\omega T - n\pi}
\]  \hspace{1cm} (2-4)

where the samples are taken at \( \omega = n\pi/T \), but \( n \) is taken from \( -\infty \) to \( \infty \).

Now, let's consider the concept of marching in time. An impulsive magnetic field incident upon a solid conducting body, sets up current \( J \) on the surface. As a result, \( J = \hat{n} \times \hat{A} \) will start generating a scattering field in all directions. After the wave passes over the target, the current created decays exponentially. The scattered field behaves similarly. At one aspect angle, the time, where the onset of the scattered waveform is observed, corresponding to the time required for the wave to reach the initial scattering point on the target and return to the radar, is designated as the initial time \( T_i \). The time \( (T_f) \) before the final exponential decay occurrence defines the end of the target.

Let \( x \) be the length of the object along the line of sight at one aspect angle

\( \Delta t \) be the time difference between the initial time and the final time at that aspect angle

\( c \) be the speed of light
\[ \Delta t = T_f - T_i \]

then,

\[ 2x = c \Delta t \]

\[ x = \frac{c(T_f - T_i)}{2} \quad (2-5) \]

Any physical object has finite dimensions. If it is located in a rectangular coordinate system \( \bar{x} : x_1, x_2, x_3 \), then it can be said to have limited dimensions in the \( \bar{x} \) coordinates. Namely, the object is confined by:

\[ x^a_1 < x_1 < x^b_1 \]

\[ x^a_2 < x_2 < x^b_2 \]

\[ x^a_3 < x_3 < x^b_3 \]

where \( x^a_1, x^a_2, x^a_3, x^b_1, x^b_2, x^b_3 \) are some real constants.

The confinement of an object in space is equivalent in saying its spatial impulse response is time limited, as the impulse response has a settling time for every aspect angle. Using the above argument, if the impulse response measurement is obtained at all aspect angles, then an image of the object may be generated from the data.
The approach taken here is a little different from Lewis-Bojarski's work [5,6]. They use physical optics approximation to arrive at the formulation for imaging. In another words, they do not use the information on the shadowed side. The object is illuminated in every direction. Information on the $4\pi$ solid angle is required, even though measurement on the complete solid frequency sphere is difficult to be implemented. With the help of the N dimensional sampling theorem, which is described later, the implementation becomes more practical. If one requires information over a finite frequency range, then the infinite number of samples over the $4\pi$ solid angle is converted to a finite number of samples. The N dimensional sampling theorem defines a sufficient sampling criterion to characterize space limited or wave number limited signal. Thus, the response at any point may be interpolated from the sampled data using the reconstruction scheme defined by the sampling theorem.

To employ the sampling theorem, the signal must be limited in time or frequency. In this case, the object is limited in three dimensions. The transformed space will be the wave number space. Ideally, the frequency response or the k-space response can be reproduced by sampling in the k-space discretely but over the infinite space. To obtain the spatial impulse response, a three dimensional inverse Fourier transform is performed on the k-space response.
While there is only one sampling theorem, there are different forms of the sampling that satisfy the sampling theorem. In one dimension, the different forms depend on how the time limited signal or the frequency limited signal is assumed to repeat itself. In three dimensions, the forms depend on how the target is placed in the reference plane \((x_1, x_2, x_3)\), and how the target's images are repeated in the three dimensional space. Theoretically, there are infinitely different forms of the sampling, as there is an infinite number of different target shapes, sizes and orientations. One would prefer a general sampling scheme that is applicable to all, or at least most, objects with any orientation. This is where the canonical containment cell fits in. These canonical containment cells are usually of simple geometries so that a wide variety of targets can be confined by their boundaries. Examples of these simple geometries are sphere, parallelepiped, ellipsoid, finite cylinder. Thus the forms of the sampling depend on the choice of the canonical confinement units and how the unit's images are repeated in the three dimensional space.

Two simple canonical units to be treated in this report are the sphere and parallelepiped. A decision rule between these two types of units will be discussed. First, the concept of sampling efficiency will be defined. The following efficiency formula is a modified version of the original formula defined by Petersen and Middleton [7].
CONSERVATIVE ESTIMATE OF THE VOLUME OF THE OBJECT

\[ \eta_s = \frac{\text{VOLUME OF THE SMALLEST SPHERE ENCLOSING THE OBJECT}}{\text{EFFICIENCY FOR ISOtropic SAMPLING}} \]  \hspace{1cm} (2-6)

\[ \eta_p = \frac{\text{VOLUME OF THE SMALLEST PARALLELEPIPED ENCLOSING THE OBJECT}}{\text{EFFICIENCY FOR PARALLELEPIPEDIC SAMPLING}} \]  \hspace{1cm} (2-7)

Rule:

\[ \eta_s > \eta_p \quad \text{use spherical confinement} \]  \hspace{1cm} (2-8)

\[ \eta_s < \eta_p \quad \text{use parallelepiped confinement} \]

The N dimensional sampling theorem obtained by Petersen and Middleton [7] is: "A function \( F(k) \) whose inverse Fourier transform \( f(x) \) vanishes over all but a finite portion in \( x \)-space can be everywhere reproduced from its sampled values taken over a lattice of points
\{l_1 \tilde{u}_1 + l_2 \tilde{u}_2 + \ldots + l_n \tilde{u}_n \}; \ l_1, l_2, \ldots, l_n = 0, \pm 1, \pm 2, \ldots \text{ provided that the vectors } \{\tilde{u}_j\} \text{ are small enough to ensure non-overlapping of the x-space signal } f(x) \text{ with its images on a periodic lattice defined by the vectors } \{\tilde{v}_i\}, \text{ which } \tilde{v}_i \cdot \tilde{u}_j = 2\pi \delta_{ij}.$$

Let's consider the non-overlapping condition in the N dimensional sampling theorem. The requirement is 'non-overlapping' of the object cell $f(x)$ with its images on a periodic lattice. Thus, the periodic lattice is not uniquely defined. Any one of Figure 2-2, 2-3, or 2-4 has a valid two dimensional periodicity. Their respective periodicity is defined by their respective $\{\tilde{v}_1, \tilde{v}_2\}$. This non-uniqueness provides flexibility on the choice of the confinement cell.

However, an efficient sampling lattice may be defined. Petersen and Middleton [7]: "An efficient sampling lattice is one which uses a minimum number of sampling points to achieve an exact reproduction of a space limited function." In another words, the closest packing of the object cell and its images without overlapping will be efficient. $\{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3\}$ will be changed if the images are rotated around the object cell; hence, the sampling lattice is still not fully defined (Figures 2-2 and 2-4). Nevertheless, any set of $\{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3\}$ defined by the above criterion will have the same efficiency, or the same number of sampling points.
Figure 2-2. Periodicity of the canonical unit: parallelogram (shaded) with its images (dotted lined) defined by $\tilde{v}_1$ and $\tilde{v}_2$
Figure 2-3. New $\tilde{v}_1$ and $\tilde{v}_2$ defining a similar periodicity as Figure 2-2 except for the extra guard band
Figure 2-4. New $v_1$ and $v_2$ defining a similar periodicity as Figure 2-2 except the images are rotated around the unit.
The application of the sampling theorem for the reconstruction of the original signal needs only an interpolation formula, provided the non-overlapping condition is met. In this application, if one has the following:

a) A signal limited in x-space and its images are specified by \( \vec{v}_1, \vec{v}_2 \) with non-overlapping condition met.

b) The sampling lattice in k-space is defined by the vector:

\[
1\vec{u}_1 + 12\vec{u}_2, \text{ where } l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots
\]

with \( \vec{v}_i \cdot \vec{u}_j = 2\pi \delta_{ij} : \delta_{ij} = \text{Kronecker delta} \)
or

\[
U = 2\pi V^{-T}
\]

where

\[
U = [\vec{u}_1 | \vec{u}_2]
\]

\[
V = [\vec{v}_1 | \vec{v}_2]
\]

\(-T\) is the notation for the transpose of the matrix inverse of \( V \)

c) The interpolation formula

then one can reproduce the two dimensional impulse response. The procedures are:

1) Sample the k-space response at the sampling lattice for

\[
F(1\vec{u}_1 + 12\vec{u}_2).
\]

2) Interpolate other required points, if necessary.
The interpolation formula taken from Petersen and Middleton [7].

\[ F(\xi_1, \xi_2) = \sum_{1_1=-\infty}^{\infty} \sum_{1_2=-\infty}^{\infty} F(1_1 \tilde{u}_1 + 1_2 \tilde{u}_2) G(k_1 \hat{k}_1 + k_2 \hat{k}_2 - 1_1 \tilde{u}_1 - 1_2 \tilde{u}_2) \]

(2-9)

where \( \xi_1 = k_1 \hat{k}_1 \)

\( \xi_2 = k_2 \hat{k}_2 \)

3) Inverse Fourier transform the k-space response to obtain the two dimensional impulse response.

\[ f(x_1, x_2) = \frac{1}{4\pi^2} \iint_{-\infty}^{\infty} F(k_1, k_2) e^{i(k_1 x_1 + k_2 x_2)} dk_1 dk_2 \]  \hspace{1cm} (2-10)

With the ease of calculation in mind, the periodicity of the object cell and its images are defined as in Figure 2-5 for this report. Its corresponding sampling lattice is shown in Figure 2-6.

The reconstruction function for the parallelogrammatic sampling [7]:

\[ G(\omega_1, \omega_2) = \frac{\sin \pi \omega_1}{\pi \omega_1} \frac{\sin \pi \omega_2}{\pi \omega_2} \]

(2-11)

where

\( \omega_1 \) is in the direction of \( \tilde{u}_1 \)

\( \omega_2 \) is in the direction of \( \tilde{u}_2 \)
Figure 2-5. A choice of periodicity for the parallelogrammatic containment unit
Figure 2-6. Sampling lattice defined by the choice of periodicity in Figure 2-5
For the sphere or isotropic confinement, the configuration is adopted from the concept of closest packing of spheres by Coxeter [9]. The configuration chosen for this report is as shown in Figure 2-7. The corresponding sampling lattice is shown in Figure 2-8.

For the two dimensional case [7,9]:

\[
\begin{align*}
\vec{v}_1 &= R \begin{bmatrix} \frac{\sqrt{3}}{2} \\ -1 \end{bmatrix} & \vec{v}_2 &= R \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
\vec{\omega}_1 &= (2\pi/R) \begin{bmatrix} \sqrt{3} \\ 0 \end{bmatrix} & \vec{\omega}_2 &= (2\pi/R) \begin{bmatrix} -1 \\ 1 \end{bmatrix}
\end{align*}
\]  

(2-12)

where \( R \) is the radius of the isotropic cell. The reconstruction function [7]:

\[
G(\omega_1, \omega_2) = \frac{1}{R^2 \omega_1 (\omega_1^2 - 3\omega_2^2)} \times \{ 2\omega_1 \cos(R\omega_1/\sqrt{3}) \cos(R\omega_2) \\
-2\omega_1 \cos(2R\omega_1/\sqrt{3}) \\
-2\sqrt{3}\omega_2 \sin(R\omega_1/\sqrt{3}) \sin(R\omega_2) \}
\]

(2-13)*

where

- \( \omega_1 \) is in the direction of \( \vec{\omega}_1 \)
- \( \omega_2 \) is in the direction of \( \vec{\omega}_2 \)

(* There is a \( \omega_2 \) factor missing in the third term of this expression in reference [7]. See Appendix A for details of the derivation).
Figure 2-7. A choice of periodicity for the circular containment cell
Figure 2-8. Sampling lattice defined by the choice of periodicity in Figure 2-7
Since this report only deals up to two dimensional sampling, the reader is referred to Petersen and Middleton's paper on the three dimensional reconstruction function $G(\omega_1, \omega_2, \omega_3)$.

The $N$ dimensional sampling theorem gives the minimum sampling lattice or criterion which will sufficiently define the $k$-space signal. All other non sampled $k$-space values can be interpolated via an extension of Equation 2-9. The $k$-space signal can be inverse Fourier transformed into the spatial domain to give a representation of the spatial impulse response. In another words, the spatial signal is also characterized by those $k$-space lattice samples.

If one is interested only in the two dimensional impulse response, then Mensa et al. [8] presents a different view on the sampling criterion. Unfortunately, it is only applicable to the two dimensions. First, rectangular to polar coordinate transformation is applied to the two dimensional Fourier integral (Equation 2-10):

$$f(x_1, x_2) = \frac{1}{4\pi^2} \iint F(k_1, k_2) e^{j(k_1 x_1 + k_2 x_2)} dk_1 dk_2$$  \hspace{1cm} (2-14)

$$\rho^2 = x_1^2 + x_2^2 \hspace{1cm} \theta = \tan^{-1}(x_2/x_1)$$ \hspace{1cm} (2-15)

Let $r^2 = k_1^2 + k_2^2$; $\phi = \tan^{-1}(k_2/k_1)$

\[
\frac{1}{4\pi^2} \int_0^2 \int_0^{2\pi} rF(r, \phi) e^{j2\pi r \rho \cos(\phi - \theta)} \, dr \, d\phi
\]  

(2-16)

Then the double integral is reduced to a single integral by considering only a particular frequency ring (i.e. \(\delta(r-w_i)\)).

\[
f_i(\rho, \theta) = \frac{\omega_i}{4\pi^2} \int_0^{2\pi} F(\omega_i, \phi) e^{j2\pi \omega_i \rho \cos(\phi - \theta)} \, d\phi
\]  

(2-17)

Thus, the two dimensional Fourier transformation is reduced to a convolution type integral. \(F(k_1, k_2)\) is the frequency response of the target in the two dimensional \(k\)-space. The notion of \(\delta(r-w_i)\) represents information taken only with one frequency. Subsequently, the two dimensional impulse response is obtainable via one integration. If information from other frequency rings are available, then superposition of every frequency ring response in the spatial domain will give a wide band two dimensional impulse response representation.

\[
f_T(x_1, x_2) = \sum_i f_i(\rho, \theta)
\]  

(2-18)

where

\[
x_1 = \rho \cos \theta \]
\[
x_2 = \rho \sin \theta
\]

Naturally, the Nyquist spacings between the frequency rings and between angular samples must be used, before the total two dimensional time response obtained can be considered a sufficient representation of the true two dimensional time response.
The angular increment which satisfies the Nyquist criterion is given by Mensa et al. [8]:

\[
\Delta \theta = \begin{cases} 
\frac{\lambda}{2D} & \lambda < 2D \\
\sin \left( \frac{\lambda}{2D} \right) & \lambda < 2D \\
\frac{\pi}{4} & \lambda > 2D 
\end{cases} 
\] (2-19)

where \( D \) = maximum dimension of the object
\( \lambda \) = wavelength of the frequency to be used

(Assumption: The origin of the x-space coincides with the middle of the object.)

The author would like to propose the following for the frequency increment:

\[
\Delta f < \frac{c}{2(1 + K)D} 
\] (2-20)

where \( D \) = maximum dimension of the object
\( c \) = speed of light
\( K \) = some safety factor

The aspect angle which has the longest dimension of the object is assumed to have the longest settling time in its impulse response. All other aspect angles require Nyquist frequency increments larger than or equal to this aspect angle. The value of the safety factor is the best estimation achievable by other means. The reason for the safety factor is not all impulse response signals are limited to the length of the object at any aspect angle. Furthermore, in the GTU sense, some of the
multiple scattering effects may be eliminated or included by employing a smaller or larger value of the safety factor. This is equivalent to truncation of the signal in time during measurement.
CHAPTER III
PRACTICAL CONSIDERATIONS

In this chapter, the practical aspects of the previously described theory are presented. Since negative frequencies cannot be physically generated, an assumption on the negative frequency response must be made. First, the assumption of a real measured signal is discussed. The frequency limits are then considered in relation to the object's size, its impulse response and the data processing requirement. For clarity, the practical aspects are discussed in one dimension. Extension to the higher dimensions can be easily accomplished.

In general, a true impulse is hard to generate; instead, a Gaussian pulse is often used. There are also times when good narrow Gaussian pulses are not readily available. In these cases, the approach of frequency sweeping may be used. The bandwidths of most oscillators, waveguide components, transmitting and receiving antennas are limited. In essence, the frequency band can only be swept from $ω_L$ to $ω_H$ (Figure 3-1). To overcome part of the problem, let's consider the one dimensional sampling theorem [1, 2]:

28
Figure 3-1. Frequency information available
If \( f(t) = 0 \) for \(|t| > T\)

then \( F(\omega) \) can be uniquely determined from

\[
F_n = F(-\frac{n\pi}{T})
\]

and

\[
F(\omega) = \sum_{n=-\infty}^{\infty} F\left(-\frac{n\pi}{T}\right) \frac{n\pi}{\omega - n\pi} \quad (3-1)
\]

{Note: If \( F(\omega) \) is even, then the required measurement
information is \( F\left(-\frac{n\pi}{T}\right) \); for \( n = 0, 1, 2, \ldots \)}

Consider the inverse Fourier Transform

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} \, d\omega \quad (3-2)
\]

\[
= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cos(\omega t) \, d\omega + \frac{j}{2\pi} \int_{-\infty}^{\infty} F(\omega) \sin(\omega t) \, d\omega \quad (3-3)
\]
If $F(\omega)$ is even, then the integrand in the second integral is an odd function of $\omega$. It follows that the second integral is zero. Therefore,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\cos(\omega t) \, d\omega$$

(3-4)

= real function, if $F(\omega)$ is even

(Note: If $F(\omega)$ is complex, then $\text{Re}[F(\omega)] = \text{Re}[F(-\omega)]$ and $\text{Im}[F(\omega)] = -\text{Im}[F(-\omega)]$ are the conditions for $f(t)$ to be real).

In this discussion, the object is real. It follows that $f(t)$ is also real. Now, information from $-\omega_H < \omega < -\omega_L$ and $\omega_L < \omega < \omega_H$ is available. (Figure 3-2)

Next, the Rayleigh Law is used to determine the scattered field at d.c. or zero frequency. From Kennaugh and Cosgriff [3]: "As the source frequency tends to zero, all finite scatterers follow the Rayleigh Law, giving a scattered field intensity which diminishes as the square of frequency." The scattered field intensity at d.c. will be zero. If

$$\frac{\pi}{\tau} > \omega_L,$$

then

$F_n$ is known for $-N < n < N$
Figure 3-2. Frequency information available after assumption is made about the negative frequency samples.
where

\[ N = \left\lfloor \frac{\omega_L T}{\pi} \right\rfloor \]  \hspace{1cm} (3-5) \]

\( I[x] \) = truncation of \( x \)'s value after the decimal point

That is, if the first sampled location (\( T \)) is higher than or equal to \( \omega_L \), then information is available from \(-\omega_H \) to \( \omega_H \), because one can interpolate the in between data points using the sampling theorem.

Let

- \( x_s \) be the object size
- \( K \) be some safety factor

then the settling time \( T_s \) is

\[ T_s = 2(1+K)x_s \]

with \( T = T_s/2 \)

One must have the condition:

\[ \pi T > \omega_L \]

\[ \frac{2\pi c}{2(1+K)x_s} > \omega_L \]

\[ \iff \frac{\pi c}{(1+K)x_s} > \frac{2\pi c}{\lambda L} \]

\[ \iff \lambda_L > 2(1+K)x_s \]  \hspace{1cm} (3-6) \]

This puts a limit on the lowest frequency usable, or the largest object size assumed by a given \( \omega_L \).
The span \( (\omega_H - \omega_L) \) effectively determines the resolution of the impulse response signature. Let \( x_r \) be the desired resolution on the object, then the impulse response resolution is

\[
\tau_r = \frac{2x_r}{c}
\]

\[
\Leftrightarrow \omega_r = \frac{2\pi c}{2x_r}
\]

One has the condition:

\[
\omega_H > \frac{2\pi c}{2x_r} = \omega_r
\]

\[
\Leftrightarrow \frac{2\pi c}{\lambda_H} > \frac{\pi c}{x_r}
\]

\[
\Leftrightarrow \lambda_H < 2x_r
\]  \quad (3-7)

Conditions 3-6 and 3-7 will help to decide how wide a frequency band may be used. If the bandwidth is wide enough, then the impulse response can be generated to a very good approximation.

A common problem during implementation may involve the inverse Fourier transform on the reconstructed frequency spectrum. This waveform may or may not be a well defined function which can be inverse Fourier transformed into a closed form solution. The common approach would be to approximate the integration using a summation on a digital computer. In effect, this approach will be a Fourier series representation which requires the time or frequency waveform to be periodic. Consequently, there are 2 more limitations:
1) The period \( (T_c) \) used in the digital computation must be greater than two times the settling time \( (T_s) \) of the impulse response. 
\[ (i.e., T_c > 2T_s) \]

2) The period \( (\omega_c) \) used in the digital computation must be greater than two times the highest frequency \( (\omega_H) \) swept. 
\[ (i.e., \omega_c > 2\omega_H) \]

Fortunately, a standard IBM subroutine FFT (Fast Fourier Transform) package is available to do the required Fourier analyses.

Furthermore, one dimensional impulse response requires a two dimensional plot. The two axis quantities are amplitude and time. Two dimensional impulse response requires a three dimensional plot. The three axis quantities are the amplitude and the plane axes. Should anyone consider three dimensional impulse response, one requires a plot in four dimensional space. Consequently, this report will only deal with impulse responses up to two dimensions. With the knowledge in one's mind that the spatial impulse response can easily be obtained by an extension of this two dimensional approach when there is an appropriate representation.
CHAPTER IV
ONE DIMENSIONAL IMPULSE RESPONSES

In this chapter, the interpolation of one dimensional impulse responses is presented; first, the result of interpolation using one dimensional data; then, using two dimensional data. The object is a six inch diameter metallic sphere. This object choice is because the Mie solution in frequency domain is readily available. In this first section, the Mie solution frequency data are sampled at different rates and interpolated either using a straight line or a sinc reconstruction function.

\[ F(\omega) = \sum_{n=-N}^{N} F(\frac{n\pi}{T}) \frac{\sin(\omega T - n\pi)}{\omega - n\pi} \]  

(4-1)

where

\[ N = I \left( \frac{\omega_H T}{\pi} \right) \]

\[ I(x) = \text{truncation of } x's \text{ value after the decimal point} \]

\[ \omega_H = \text{highest frequency used} \]

[Note: This is Equation (3-1) except the summation is from -N to + N].
A cosine tapering weighting function [10]: (Figure 4-1) (for the convenience of the reader, all tables and figures of Chapter IV are grouped together at the end of the chapter)

\[
W(n) = \begin{cases} 
0.5[1 + \cos\left(\frac{n\pi}{N_1}\right)] & |n| < N_1 \\
0 & |n| > N_1
\end{cases}
\] (4-2)

is multiplied to the interpolated frequency data. The purpose of the weighting or filtering is to reduce the effect of the Gibb's phenomenon [2]. The resulting data are inverse Fourier transformed into the time domain discretely to give the impulse response in time.

Figure 4-2 represents a time domain impulse response plot obtained from the Mie solution for a six inch diameter metallic sphere using the frequency spectrum from 0 to 12 Ghz with a cosine tapering filter. The imperfect specular impulse and the ripples around it at the start of the response are caused by 1) finite bandwidth, and 2) Gibb's phenomenon.

Figure 4-2 is considered to be the standard for comparison with the other responses. Its frequency samples are taken every 60 Mhz and linked together by straight lines. Figure 4-3 has frequency samples taken every 0.125 Ghz over the spectrum of 0 to 12 Ghz and interpolated the in between points using Equation (4-1). The differences of the impulse responses in this chapter from Figure 4-2 (the 'exact' solution) are shown in figures designated with their respective figure number plus an 'a' attached. For example, to obtain Figure 4-2 from Figure 4-3, one adds Figure 4-3a to Figure 4-3.
Figures 4-4, 4-5, and 4-6 are similar to Figure 4-3, except frequency samples are taken every 0.25 Ghz, 0.5 Ghz, and 0.75 Ghz respectively. Figure 4-5 is still recognizable to be Figure 4-2 but Figure 4-6 is not. Figure 4-6 does not have similar behavior because it employs a frequency sampling rate below the Nyquist rate. From Figure 4-2, one can estimate the settling time of the impulse response of the sphere to be about 1.5 nsec. Equally well, one could use a longer settling time depending on one's assumption of the noise amplitude.

\[ T = 0.75 \times 10^{-9} \text{s} \]

\[ f_s > \frac{1}{2T} = \frac{1}{1.5 \times 10^{-9}} \approx 0.66 \text{ Ghz} \]

\[ < 0.75 \text{ Ghz (Figure 4-6)} \]

Figures 4-2 to 4-5 have sampling rate better than the Nyquist rate. If the sampling rate is high enough, then one may use straight line interpolation (Figure 4-2) instead of the sinc reconstruction function to save computer time on interpolation. On the other hand, if samples are scarce but still satisfy the Nyquist criterion, then the sinc reconstruction function (Equation (4-1)) is preferred for more pleasing results. (Figures 4-3, 4-4, 4-5)

Now, the reconstruction of the one dimensional impulse responses from data obtained on two dimensional isotropic and cubic sampling lattice is presented.

\[ F(k_1, k_2) = \sum_{l_1} \sum_{l_2} F(l_1 u_1 + l_2 u_2) G(k_1 k_1 + k_2 k_2 - l_1 u_1 - l_2 u_2) \]

\[ (4-3) \]
where

\[ l_1 \text{ and } l_2 \text{ are summed over } l_1 \text{ and } l_2 \text{ which satisfy} \]

\[ k_L < |l_1 \bar{u}_1 + l_2 \bar{u}_2| < k_H \]

and

\[ k_L \text{ and } k_H \text{ define the frequency bandwidth used.} \]

(Note: This is Equation (2-9) with finite summation.)

Samples are taken out of the Mie's frequency solution on the isotropic sampling lattice defined by \( \bar{u}_1 \) and \( \bar{u}_2 \) in Equation (2-12), and the cubic sampling lattice defined by:

\[
\bar{u}_1 = 2\pi \begin{bmatrix} l_1 \\ 0 \end{bmatrix} \quad \bar{u}_2 = 2\pi \begin{bmatrix} 0 \\ l_1 \end{bmatrix} \tag{4-4}
\]

i.e.,

samples are taken over \( l_1 \bar{u}_1 + l_2 \bar{u}_2 \) for \( l_1, l_2 = 0, \pm 1, \pm 2, \ldots \)

\[
\tag{4-5}
\]

This sampling lattice is generated by the program PTGRIU (see Appendix B). The frequency response for an aspect angle is interpolated using Equation (4-3) in conjunction with either Equation (2-13) for isotropic sampling or Equation (2-11) for cubic sampling. This interpolation work is done by the program INTERPOL (see Appendix B). The resulting frequency response is again cosine tapering low pass filtered to reduce the Gibb's phenomenon. After the filtering, the frequency spectrum is inverse Fourier transformed discretely into the time domain to give an impulse response picture for the six inch metallic sphere. The
filtering and the discrete inverse Fourier transform are functions of FTRAN [11] (see Appendix B).

Sampling at every 0.5 Ghz for a six inch diameter metallic sphere, is equivalent to assuming a signal having four times the diameter of the sphere in one dimensional space. Therefore, the two dimensional confinement cell is assumed to contain the sphere and has a guard band of 1.5 times the maximum dimension of the sphere surrounding the sphere. The safety factor thus chosen is 1 (i.e., \(2(1+K)=4 \Leftrightarrow K=1\)).

The frequency range sampled is 0 to 12 Ghz.

Figures 4-7, 4-8, 4-9, and 4-10 are one dimensional impulse responses reconstructed from two dimensional isotropic sampling data, for aspect angles of 0, 0.719, 1.438, 30 degree respectively. The CPU time taken for interpolating each waveform is about 5 minutes for 200 plotting points. Figures 4-11, 4-12, 4-13, 4-14 are reconstructed from cubic sampling data, for aspect angles of 0, 1.193, 2.386, 45 degrees. The CPU time taken for these waveforms is about 2.5 minutes for 200 plotting points. Less time in interpolation for cubic sampled data is probably due to the simplicity of the reconstruction function (Equation (2-11) versus (2-13)). These angular choices are arbitrarily chosen. One should note the close resemblance of all these figures (4-7 to 4-14) with Figure 4-5. Next, let's consider the case of more samples taken. Effectively, the safety factor is changed from one to three but the frequency range remains the same. Figures 4-15, 4-16, and 4-17 are reconstructed one dimensional impulse responses using isotropic samples for aspect angles of 0, 0.352, 30 degrees respectively. Again discrepancy is not high (Figures 4-15a, 4-16a, and 4-17a). Since the
density of samples taken is finer, or more samples participated in the interpolation, the CPU time has increased to about 20 minutes for 200 plotting points.

If both the isotropic and cubic sampling can perform competitively, how does one decide on which sampling grid? The answer lies in the efficiency definition defined previously. However, the area is considered in a plane instead of the volume in a three dimensional space.

i.e., one modifies Equations (2-6) and (2-7) to

\[
\text{Efficiency} = \frac{\text{AREA OF A CROSS-SECTION ON THE OBJECT}}{\text{AREA OF THE SAME CROSS-SECTION ON THE TWO DIMENSIONAL ENCLOSURE}}
\]

but the decision rule: Equation (2-8), remains the same.

Efficiency, as mentioned before, is defined as a minimum sampling requirement. Three objects: a six inch diameter sphere, a \(3\sqrt{2}\) inch cube and a sphere cap cylinder are shown in Figure 4-18 on their major axis cross-section. Their respective cross-sectional areas; closest circular, squared, rectangular enclosure cross-sectional areas; and efficiencies are tabulated in Table 4-1.

The definition of Equation (4-5) is used to locate the sampling lattice. The number of these locations over a frequency range is summed to give the minimum number of sampling, defined by the sampling theorem, to sufficiently characterize the spatial impulse response in that frequency range. This work is done by the program PTGRID (see Appendix B). The numbers are tabulated in Table 4-2 and plotted in Figures 4-19,
4-20, and 4-21. A safety factor of one is used in the computation. The sizes of the objects are chosen such that the circular enclosure is the same for all three objects for easy comparison in the graphs.

As the frequency range becomes larger and larger, the significance of the efficient sampling grid becomes more and more important. Let's take the example of the sphere cap cylinder (Figure 4-19). The use of the rectangular enclosure will provide an efficiency of 0.96. The number of samples required over 0 to 12 GHz is 696. The use of the squared enclosure can only give an efficiency of 0.38. The number of samples required is 2.5 times that of the rectangular enclosure. The use of the circular enclosure has an efficiency of 0.45. The number of samples is about 2.3 times that of the rectangular enclosure. As the frequency range is expanded, saving in measurement time by the proper choice of the enclosure becomes very substantial.
**TABLE 4-1**

**EFFICIENCY COMPARISON ON 3 OBJECTS USING 3 TYPES OF CONTAINMENT CELLS**

<table>
<thead>
<tr>
<th>Object</th>
<th>SPHERE</th>
<th>CUBE</th>
<th>SPHERE CAP CYLINDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area on major axis</td>
<td>182.4</td>
<td>116.1</td>
<td>82.2</td>
</tr>
<tr>
<td>Area of closest circular enclosure</td>
<td>182.4</td>
<td>182.4</td>
<td>182.4</td>
</tr>
<tr>
<td>Area of closest squared enclosure</td>
<td>232.3</td>
<td>116.1</td>
<td>214.7</td>
</tr>
<tr>
<td>Area of closest rectangular enclosure</td>
<td>232.3</td>
<td>116.1</td>
<td>85.9</td>
</tr>
<tr>
<td>Efficiency of circular enclosure</td>
<td>1</td>
<td>0.64</td>
<td>0.45</td>
</tr>
<tr>
<td>Efficiency of squared enclosure</td>
<td>0.79</td>
<td>1</td>
<td>0.38</td>
</tr>
<tr>
<td>Efficiency of rectangular enclosure</td>
<td>0.79</td>
<td>1</td>
<td>0.96</td>
</tr>
</tbody>
</table>

(Note: the areas are in units of squared centimeters)
<table>
<thead>
<tr>
<th>Object</th>
<th>SPHERE</th>
<th>CUBE</th>
<th>SPHERE CAP CYLINDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of enclosure</td>
<td>circular</td>
<td>squared</td>
<td>squared</td>
</tr>
<tr>
<td>Frequency ranges from 0 up to (Ghz)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>96</td>
<td>120</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>186</td>
<td>212</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>282</td>
<td>324</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>408</td>
<td>472</td>
<td>240</td>
</tr>
<tr>
<td>7</td>
<td>598</td>
<td>640</td>
<td>324</td>
</tr>
<tr>
<td>8</td>
<td>720</td>
<td>828</td>
<td>420</td>
</tr>
<tr>
<td>9</td>
<td>912</td>
<td>1048</td>
<td>516</td>
</tr>
<tr>
<td>10</td>
<td>1134</td>
<td>1304</td>
<td>656</td>
</tr>
<tr>
<td>11</td>
<td>1368</td>
<td>1564</td>
<td>776</td>
</tr>
<tr>
<td>12</td>
<td>1626</td>
<td>1876</td>
<td>940</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plotted in Figure</th>
<th>4-19</th>
<th>-</th>
<th>-</th>
<th>4-19</th>
<th>4-19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation:</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

44
Figure 4-1. Cosine tapering weighting
(Equation (4-2) with $N_1 = 16$)
Figure 4-2. Impulse response of a 6" metallic sphere with frequency samples taken every 60 Mhz over the range of 0 to 12 GHz.
Figure 4-3. Impulse response of a 6" metallic sphere with frequency samples taken every 125 Mhz over the range of 0 to 12 Ghz and interpolated using Equation (4-1)
Figure 4-3a. Error of Figure 4-3 from Figure 4-2
(Magnification = 10x)
Figure 4-4. Similar description as Figure 4-3, except frequency samples are taken every 250 Mhz
Figure 4-4a. Error of Figure 4-4 from Figure 4-2 (Magnification = 10x)
Figure 4-5. Similar description as Figure 4-3, except frequency samples are taken every 500 Mhz
Figure 4-5a. Error of Figure 4-5 from Figure 4-2
(No magnification)
Figure 4-6. Similar description as Figure 4-3, except frequency samples are taken every 750 MHz.
Figure 4-6a. Error of Figure 4-6 from Figure 4-2
(No magnification)
Figure 4-7. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from 2-D isotropic lattice samples taken with a safety factor of 1 over the range of 0 to 12 GHz using Equation (4-3) and (2-13)
Figure 4-7a. Error of Figure 4-7 from Figure 4-2
Figure 4-8. Similar description as Figure 4-7, except the aspect angle is $72^\circ$. 
Figure 4-8a. Error of Figure 4-8 from Figure 4-2
Figure 4-9. Similar description as Figure 4-7, except the aspect angle is 1.44°
Figure 4-9a. Error of Figure 4-9 from Figure 4-2
Figure 4-10. Similar description as Figure 4-7, except the aspect angle is 30°
Figure 4-10a. Error of Figure 4-10 from Figure 4-2
Figure 4-11. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from 2-D cubic lattice samples taken with a safety factor of 1 over the range of 0 to 12 GHz using Equation (4-3) and (2-11)
Figure 4-11a. Error of Figure 4-11 from Figure 4-2
Figure 4-12. Similar description as Figure 4-11, except the aspect angle is 1.19°.
Figure 4-12a. Error of Figure 4-12 from Figure 4-2
Figure 4-13. Similar description as Figure 4-11, except the aspect angle is 2.39°.
Figure 4-13a. Error of Figure 4-13 from Figure 4-2
Figure 4-14. Similar description as Figure 4-11, except the aspect angle is 45°
Figure 4-14a. Error of Figure 4-14 from Figure 4-2
Figure 4-15. Impulse response of a 6" metallic sphere at 0° aspect angle with the 1-D frequency response interpolated from a 2-D isotropic lattice samples taken with a safety factor of 3 over the range of 0 to 12 Ghz using Equation (4-3) and (2-13)
Figure 4-15a. Error of Figure 4-15 from Figure 4-2 (Magnification = 10x)
Figure 4-16. Similar description as Figure 4-15, except the aspect angle is 0.35°
Figure 4-16a. Error of Figure 4-16 from Figure 4-2 
(Magnification = 10x)
Figure 4-17. Similar description as Figure 4-15, except the aspect angle is 30°.
Figure 4-17a. Error of Figure 4-17 from Figure 4-2
(Magnification = 10x)
Figure 4-18. Dimensional perspective among the sphere, cube and sphere cap cylinder used in the example (Diameter of sphere = 6 inches)
Figure 4-19. The number of samples specified by the different types of sampling lattices on a sphere cap cylinder described in Figure 4-18
Figure 4-2U. The number of samples specified by two types of sampling lattices on a 6" sphere.
Figure 4-21. The number of samples specified by two types of sampling lattices on the cube described in Figure 4-18.
Thus far, the discussion has focused only on the number of samples required. The smaller the number of samples, the less measurement time is required. However, if the two dimensional impulse response is of interest, then the transform method must also be considered. This chapter will discuss the potential time consumption problem of multi-dimensional Fourier transform plus some possible solutions. Image reconstruction using the spatial impulse response is also considered.

While an isotropic enclosure may be more efficient to enclose a sphere than a cube, it is not as easy to do three dimensional discrete Fourier analyses as the cubic enclosure. Most of the discrete Fourier transform techniques are developed to fit equally spaced data. In another words, programs are written to perform readily on the cubic sampling lattice. Any other sampling grid data must be interpolated to the cubic grid either before or after the discrete Fourier transform step; otherwise, the proper representation cannot be achieved. Sometimes, interpolation may also be desired for the cubic sampling grid data, as in the case of higher resolution requirement than

81
measured. This type of interpolation requires a lot of computer time as pointed out in Mensa et al.'s paper [8].

Let's consider an example to estimate the time involved.

From Chapter IV:

1) Interpolation for 200 plotting points using 1626 isotropic samples is about 5 mins.
2) Interpolation for 200 plotting points using 1876 cubic samples is about 2.5 mins.
3) The number of samples required on a sphere cap cylinder for circular, rectangular and squared enclosures are 1626, 696, 1740 respectively.

(Tables 4-2)

Compact range measurement facility at U. S. U. per Walton [12]:

The measurement system response time is about:

1) 0.2s/data point, if frequency scan is used.
2) 1 min/frequency ring, if angular scan is used.

The time estimation for each type of the sampling measurement on the sphere cap cylinder and two dimensional interpolation is presented in Table 5-1. (For the convenience of the reader, all tables and figures of Chapter V are grouped together at the end of the chapter.) The interpolation time for the rectangular enclosure data may also be considered zero. By remembering a scale factor, the data can be processed as in a squared grid. If the proper signal representation or a finer resolution is required, the interpolation step is still
unavoidable. Therefore, the numbers in the table do give a fair comparison, as all data are brought to the same level—squared grid representation. The interpolation step plays an important role on the decision of which type of sampling to use. The saving in measurement time is sometimes balanced out by the data processing time.

The Nyquist criteria (Equations (2-19), (2-20)) plus the polar transformation (Equation (2-15)) described earlier give new perspective for the efficient but non-cubic enclosure. Now one only needs to interpolate for a sufficient number of frequency rings. This work is done by program INTERPOL (see Appendix B). Each frequency ring is transformed to the spatial domain individually and summed together to get the total time response. The former is the work of program INTEGFFT (see Appendix B); the latter is the job of program SUM3D (see Appendix B). The interpolation performed this way may require as much time as the interpolation onto the squared grid. It nevertheless gives an alternative way to the solution of the problem. Now, another question may be raised: Why perform interpolation if it is so time consuming?

The interpolation may be avoided, if the two dimensional time response is the only interest. All one has to do is to measure the frequency rings and then process the data as described before. However, if one desires the impulse response of a target at a particular aspect angle or the response over a frequency ring other than those measured, one is required to develop a different interpolation scheme than the one described in the theory section. A possible way to reduce the
interpolation time is to derive a faster interpolation scheme or a
general multi-dimensional Fourier transform technique which operates on
any grid. A parallel processor which uses optics may be another
possibility in terms of hardware. A lens system has a Fourier
transform relationship between the source and the image [13].

The Mie solution for a sphere is a very good data example; except
its backscattered response is isotropic. A proper choice of test object
must have non-isotropic property. The sphere shifted off the centre of
the plane is one possibility, but it only has variation in the phase
term and not in the magnitude term. Since most of the other exact
solutions are not readily available, a first order UTD solution for a
finite circular cylinder [14] is used; with the caution that UTD gives a
valid approximation to the exact solution at high frequency.

The size of the circular cylinder is chosen to be six inches in
length and three inches in diameter. Having chosen the size of the
cylinder, one has defined the low frequency limit of this UTD model to
about 2 Ghz. Since the step and ramp response of an object reduce the
need for the high frequency information, only the impulse response of
this cylinder solution is considered here.

Figure 5-1 is the two dimensional impulse response (Equation (2-9))
for the above circular cylinder. The frequency samples are first
generated on the grid points defined by Equation (4-5) over only 180°.
In this case, the spatial impulse response is assumed to settle after
the wave has passed over four times the length of the object at every
aspect angle. The rectangle so designated has dimensions: 24" in
length and 12" in diameter. The periodic lattice is chosen as in Figure 2-3. Consequently,

$$\tilde{v}_1 = 24" \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \tilde{v}_2 = 12" \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The guard band is 9" on each side of the cap of the cylinder and 4.5" on the circular surface. Then,

$$\tilde{u}_1 = \frac{2\pi}{24"} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \tilde{u}_2 = \frac{2\pi}{12"} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Since,

$$l_1\tilde{u}_1 = l_1|\tilde{u}_1|k_1$$

$$l_2\tilde{u}_2 = l_2|\tilde{u}_2|k_2$$

where

$$l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots$$

Therefore, the sampling lattice is defined by:

$$l_1\tilde{u}_1 + l_2\tilde{u}_2 = l_1|\tilde{u}_1|k_1 + l_2|\tilde{u}_2|k_2$$

$$\iff$$ points on the k-plane:

$$k_1 = l_1|\tilde{u}_1| \text{ and } k_2 = l_2|\tilde{u}_2|$$
where

\[ l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, \ldots \]

Then the sampled data are Fourier transformed discretely into the spatial domain. The discrete two dimensional Fourier transform is performed by an IBM subroutine HARM (see Appendix B). The frequency range used is 2 to 5 GHz. The safety factor of one is used. The vertical polarization data are taken from the UTU solution on a major axis cut. Figure 5-1a is the contour plot of Figure 5-1. Figure 5-2 is also a two dimensional impulse response for a cylinder solution that has the same previous description. The differences are the technique of the Fourier transformation and the sampling locations in the frequency plane. It employs Mensa et al.'s method on 6 different frequency rings: 2.5, 3, 3.5, 4, 4.5, 5 GHz. (Equation (2-18)) The samples are taken so that Equations (2-19) and (2-20) are satisfied. Again the samples are taken over 180°. Figure 5-2a is the contour plot of Figure 5-2.

Comparing Figures 5-1 and 5-2, one can see many similarities. The differences can be deduced by recalling their respective generating methods. Figure 5-1 has information scattered all over the frequency range; while, Figure 5-2 has values only over those frequency rings mentioned before. There is also the processing error involved.

One interesting thing is to be noted in Figure 5-1, or 5-2. If one records just the highest points within its neighborhood, one can trace out a rectangle. This may be easier to see on a contour plot (Figure 5-1a, 5-2a). This looks like one cross-section of the target. If \( 2\pi \)
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS — B453 — 1
solid angle information is available, then an image of the target can indeed be produced. Of course, the resolution is still governed by the highest frequency used. In this case the bandwidth used is quite narrow; hence, the resulting resolution is not high.

Figure 5-3 is generated similarly as Figure 5-2. It is generated using Mensa et al.'s method on 16 different frequency rings: 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10 Ghz. Figure 5-3a is the contour plot of Figure 5-3. The dimensions of the rectangle defined by the high peaks do correspond to the major axis cross-section of the circular cylinder described earlier. One may wonder how a rectangle is concluded from Figure 5-3a. There are a few clues available. The T_i's are quite obvious from the high peaks on the illuminated side. The final time (T_f) of the one dimensional impulse response, as one recalls, is defined by the beginning of the exponential decay of the signal. The final peaks at the coordinates (5,3) and (5,-3) are indeed higher than any point x_1 > 5. The ridges and valleys on the shadow side (x_1 > 5) of the cylinder is fairly straight.

Let's consider the case where this imaging theory converges to Lewis-Bojarski's work. Figure 5-4 is the same as Figure 5-3 except data are taken over 360°, or illuminated in every direction on the two dimensional plane. Figure 5-4a is the contour plot of Figure 5-4. As expected, the peaks (or T_i's) in the figure trace out a rectangle which is the major axis cut of the circular cylinder. One may note that literature today usually presents data plots using the absolute value of the amplitude. One must be careful when confronted by these plots.
As Figures 5-5 and 5-5a have shown, the absolute value of the amplitude does not usually tell the whole picture. Figure 5-5 plots the absolute value of the amplitude in Figure 5-4. Figure 5-5a is the contour plot of Figure 5-5. The major axis cross-section is no longer as well defined by the peaks as in Figure 5-4 or 5-4a. Nevertheless, producing an image using the spatial impulse response is very promising.

Although this thesis' imaging theory is not as 'rigorous' as Lewis and Bojarski's work, the theory is more flexible in application. The object is only required to be illuminated at the aspect angles over a $2\pi$ solid angle. The shadowed side information is also employed in the image reconstruction process. There is no need for any assumption on the object's shadowed side geometry. If the start and the end of the object's one dimensional impulse response for every aspect angle over half of the $4\pi$ solid angle are well defined by $T_i$'s (illuminated) and $T_f$'s (shadowed) as described earlier, then an image of the object may be produced using the $T_i$'s and $T_f$'s. For $T_i$'s and $T_f$'s not defined distinctly, careful interpretation on the spatial impulse response must be used. In the case of a full $4\pi$ illumination, the image may be reconstructed using only $T_i$'s information. This converges to Lewis-Bojarski's identity. This theory is better in utilizing data information available but it lacks a concrete proof.

Let's investigate this imaging possibility further using a six inch diameter metallic sphere whose centre is located three inches off the centre of the reference plane on the $x_1$-axis. The cubic sampling
lattice used has a safety factor of three. The frequency range used is 0 to 12 Ghz. Again, the frequency data are taken out of the Mie solution. The sampling lattice is defined by Equation (4-5) over half of the plane ($\pi < \theta < 2\pi$). The data are weighted by the two dimensional cosine tapering function. The two dimensional version of Equation (4-2) is replacing the variable $n$ by the radial distance from the centre of the k-space. The shape of the function is Figure 4-1 rotated around the weighting axis. The weighted data are Fourier transformed into the spatial domain and presented as Figure 5-6. Figure 5-6a is the contour plot of Figure 5-6.

Again the initial speculars trace the illuminated side of the sphere nicely. The $T_f$'s are not as well defined as the finite cylinder case. More interpretation work is required. The valley on the shadow side shows a curvature. This may be indicating the back side of the object having a curvature. The radius of the valley's curvature is about $3\pi$ inches, which is the equivalent distance a wave would creep before scattering back in the transmitting direction (Figure 5-7). The radius of the curvature created by the initial speculars is about six inches, which is the $2R$ distance travelled by the wave. The first $R$ is the distance travelled to the target. The second $R$ is the distance travelled by the wave scattering back from the target. Using this information, the circle with the three inch radius can be formed.
Of course, if one uses data over the full plane as Lewis-Bojarski's work, then there is no need for the previously described type of pattern recognition interpretation. The initial speculars usually define the perimeter of the object, if it is a smooth convex body. The perimeter indicated is only one cross-section of the target on the major axis. If more cross-sectional information of the target is available, then an image of the whole object may be produced. The potential on the image reconstruction is high, but more research work is required in the area; particularly in the pattern recognition area.
### TABLE 5-1

**AN EXAMPLE: TIME ESTIMATION**

<table>
<thead>
<tr>
<th>Type of enclosure</th>
<th>Measurement time (mins)</th>
<th>Interpolation time (mins)</th>
<th>Total time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>circular</td>
<td>5.42</td>
<td>40.09</td>
<td>45.51</td>
</tr>
<tr>
<td>rectangular</td>
<td>2.32</td>
<td>8.47</td>
<td>10.79</td>
</tr>
<tr>
<td>squared</td>
<td>5.80</td>
<td>-</td>
<td>5.80</td>
</tr>
</tbody>
</table>
Figure 5-1. 2-D impulse response of a finite metallic circular cylinder with 6" in length and 3" in diameter. Frequency samples are taken at the cubic sampling lattice over the range of 2 to 5 Ghz and Fourier transformed discretely into the spatial domain. Data are taken over the half plane: $0 < \theta < \pi$. 
Figure 5-1a. Contour plot of Figure 5-1
Figure 5-2. 2-D impulse response of a finite metallic circular cylinder with 6" in length and 3" in diameter. Frequency samples are taken over 6 rings: 2.5, 3, 3.5, 4, 4.5, 5 GHz and transformed into the spatial domain via Mensa et al.'s method. Data are taken over the half plane: $0 < \theta < \pi$. 
Figure 5-2a. Contour plot of Figure 5-2
Figure 5-3. Similar description as Figure 5-2 with 10 additional frequency rings: 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10 Ghz
Figure 5-3a. Contour plot of Figure 5-3
Figure 5-4. Similar description as Figure 5-3, except the data are taken over the full 360° plane.
Figure 5-5. The plot of the absolute value of the amplitude in Figure 5-4.
Figure 5-6. 2-D impulse response of a 6" metallic sphere with its centre situated at (3", 0). Frequency samples are taken at the cubic sampling lattice over the range of 0 to 12 Ghz. The data are cosine tapering weighted before Fourier transformed discretely into the spatial domain. Data are taken over the half plane: \( \pi < \theta < 2\pi \)
Figure 5-7. Path length of a creeping wave on a metallic sphere
CHAPTER VI

CONCLUSIONS

If a signal is limited in the spatial domain (or wave number domain), this signal is sufficiently characterized by its values over the discrete sampling lattice in the wave number domain (or the spatial domain). In the two dimensional case, the sampling locations in the wave number space are specified by the vector:

\[ [l_1 \tilde{u}_1 + l_2 \tilde{u}_2] \]

where

\[ l_1, l_2 = 0, \pm 1, \pm 2, \pm 3, ... \]

The \( \tilde{u}_1 \) and \( \tilde{u}_2 \) are related to \( \tilde{v}_1 \) and \( \tilde{v}_2 \) by the following:

\[ [\tilde{u}_1 | \tilde{u}_2] = 2\pi [\tilde{v}_1 | \tilde{v}_2]^{-T} \]

where

\( -T \) : the transpose of the inverse of the matrix formed
The $\vec{v}_1$ and $\vec{v}_2$ are specified by one's choice on how the space limited signal is assumed to repeat itself in its domain. $\vec{v}_1$ is pointed to the centre of one of the closest images in the periodic lattice. $\vec{v}_2$ is independent of $\vec{v}_1$, and it points at the centre of another close image. By defining the settling time as the end of the impulse response, one has a space limited three dimensional impulse response signal. In the finite circular cylinder data presented in Chapter V, the signal is assumed to be 4 times the size of the cylinder in the two dimensional spatial plane. The choice of 4 is equivalent to choosing the one dimensional impulse response of having a settling time four times as long as the wave would travel over the length of the object at any aspect angle. Or the safety factor is chosen to be one. (i.e., $2(1+K) = 4 \iff K = 1$) This factor provides sufficient results in both Chapter IV and V.

In Chapter V, the example of the finite circular cylinder with dimensions: 6" in length and 3" in diameter, the two dimensional containment unit is chosen to be a square. The squared repetitive lattice in the spatial domain is defined by,

$$
\vec{v}_1 = R \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{v}_2 = R \begin{bmatrix} 0 \\ 1 \end{bmatrix}
$$

where

$$R = 6"$$

and the safety factor $K$ is chosen to be 1. With these inputs to the program PTGRID (see Appendix B), the program outputs the sampling lattice defined by $\vec{u}_1$ and $\vec{u}_2$. The data output is arranged with the
aspect angles and the corresponding frequency increment. One makes the necessary data extraction at the proper aspect angles, and increments the frequency until the highest frequency is reached. One could also have defined a rectangular repetitive lattice. Then

\[ \bar{v}_1 = R \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \bar{v}_2 = R \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \]

where

\[ R = 6'' \]

Or, for a circular repetitive lattice,

\[ \bar{v}_1 = R \begin{bmatrix} \sqrt{3} \\ 2 \\ -2 \end{bmatrix}, \quad \bar{v}_2 = R \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

where

\[ R = \sqrt{(6'')^2 + (3'')^2} = 6.708'' \]

Again these \( \bar{v}_1 \) and \( \bar{v}_2 \) values can be input into the program PTGRID (see Appendix B) to obtain the sampling lattice.

Thus, one can sample discretely and interpolate in the wave number space to reproduce the frequency response of an object. Fourier transforming this wave number signal into the spatial domain gives a representation of the spatial impulse response of the object. The
spatial impulse response is defined as the image function obtained by three dimensional Fourier transforming the far field frequency response of a finite object at all aspect angles defined over the 4π solid angle. Since most objects have different shapes, sizes, and orientations, their corresponding spatial impulse responses are different. Their corresponding sampling lattices will also be different. One general sampling lattice that is applicable to a set of objects is desirable. This is accomplished by introducing different types of canonical containment units to confine the spatial impulse responses. Two common canonical cells are parallelepiped and sphere. Two dimensional examples are shown in Figures 2-5 and 2-7, with their corresponding sampling lattices shown in Figures 2-6 and 2-8 respectively.

In Chapter IV, a six inch metallic sphere is chosen as an example to compare two types of sampling lattices - cubic and isotropic. The comparison is performed on the interpolated one dimensional impulse responses at different aspect angles using the interpolation scheme defined in the sampling theorem. As one expects, the results turn out to be competitive for the two types of sampling. Efficiencies, in the sense of the least number of sampling points, are different for different canonical containment cells used on the same object.

Efficiency in using one type of canonical containment cell:

\[ \eta = \frac{\text{CONSERVATIVE ESTIMATE OF THE OBJECT'S VOLUME}}{\text{VOLUME OF THE SMALLEST CANONICAL CONTAINMENT CELL ENCLOSING THE OBJECT}} \]
Efficiencies among a sphere cap cylinder, a sphere and a cube using cubic, isotropic, and rectangular box confinement units are presented in Chapter IV. The efficiency definition proves to be a very good concept in deciding the type of sampling lattice for an object or a group of objects. This is under the assumption that the settling time of the spatial impulse response is shaped similarly to the object; e.g., the settling time of the impulse response of a sphere is the same in every aspect angle.

To obtain an approximation to the spatial impulse response from the sampled data over a finite frequency range, one can use the discrete Fourier transform. Because of today's digital computer design, the different sufficient characterization cannot be readily processed without interpolation; except, of course, the cubic lattice data sets. The interpolation step which most people like to avoid, is very time consuming. This may be referred back to the time estimation example on a sphere cap cylinder presented in Chapter V. The interpolation step is another factor that affects an engineering decision. The avoidance helps Mensa et al. to arrive at the time response faster. The price they paid is the limitation of their method's application to two-dimensional Fourier transform. Their approach is thus not recommended because of its inability to be expanded into higher dimensions. The sampling criteria accompanying their method are,
the angular increment:
\[
\Delta \theta = \begin{cases} 
\frac{\lambda}{2D} & \lambda \ll 2D \\
\sin^{-1}\left(\frac{\lambda}{2D}\right) & \lambda < 2D \\
\frac{\pi}{4} & \lambda > 2D
\end{cases}
\]

the frequency increment:
\[
\Delta f < \frac{c}{2(1+K)D}
\]

where
- \(D\) = maximum dimension of the object
- \(\lambda\) = wavelength of the frequency used
- \(c\) = speed of light
- \(K\) = some safety factor

In the finite circular cylinder example,
- \(D = 17.04\) cm
- \(K = 0.75\)
- \(\Delta f < 0.503\) Ghz

at the highest frequency of 10 Ghz,
- \(\lambda = 3\) cm
- \(\Delta \theta < 5\) degrees

Therefore,
- the frequency increment chosen = 0.5 Ghz
- the angular increment chosen = 1 degree
Nevertheless, processed results from UTD solution on a finite circular cylinder support the convergence of Mensa et al.'s method to two-dimensional discrete Fourier transform.

Lewis-Bojarski's identity requires either the object be illuminated at all angles or assumption be made on the shadowed side. Results on a finite metallic circular cylinder and a metallic sphere indicate that the spatial impulse response approach does not have the above restriction, though proper interpretation may be required. Furthermore, the use of the spatial impulse response to imaging can converge to Lewis-Bojarski's results. There is also an indication that the presentation in the form of the absolute value of the amplitude does not necessarily provide the proper picture for pattern recognition. The plain amplitude representation with positive and negative values is sometimes more appropriate. This is concluded by comparing the Figure 5-4, or 5-4a with 5-5 or 5-5a. The perimeter of a major axis cross-section of a finite circular cylinder is shown more distinctly using the plain amplitude presentation. Judging from the two-dimensional impulse responses, one can deduce the substantial potential of the spatial impulse response in image reconstruction.

The spatial impulse response has numerous applications including target identification and imaging. Although smooth convex metallic body examples are considered here, tomographic applications on other types of bodies are possible. In all, the N-dimensional sampling theorem provides new insights into the sampling criteria in the wave number space for a finite object. The potential in reduced management time on
two or three dimensional data is enormous. The two dimensional impulse response also projects a promising target imaging technique using the spatial impulse response.
CHAPTER VII

RECOMMENDATIONS

Traditionally, two and three dimensional data are presented in cubic lattices, for which present day digital computers are designed. Although the digital computer of today manages data in cubic lattices, the cubic lattices are not necessarily the most efficient in terms of the least number of data samples. The most general approach to solve this computer problem requires the cubic data management structure of the digital computer be modified into a more general data structure. In another words, data organized in any random fashion can be processed by this computer. If this general approach is not practical, the next best step is a faster interpolation scheme either in hardware or software. The lowest level on the hierarchy of improvement is the improvement for specific application. In this thesis, a general Fourier transform that can perform on any data lattice, fits into this category.

After some of the computer problems are solved, the next step in the development is to account for the experimental noise. How does one extract the true information that is embedded in noise? Without this
generalization, this report is only useful in information storage and retrieval on noise free data. Noise free is in the sense that the noise effect in measurement is eliminated before storage. The full development of this report's theory will enable significant reduction in time on data measurement, processing and storage.

The whole report is focused on the monostatic data. It would be interesting to see how this theory holds up with the bistatic data. Using the definition of the three dimensional Fourier transform, one can derive a similar method that parallels Mensa et al.'s approach.

By converting the k-space coordinates into spherical coordinates:

\[ k_1 = \rho \sin \theta \cos \phi \quad k_2 = \rho \sin \theta \sin \phi \quad k_3 = \rho \cos \theta \]  

(7-1)

equation (1-1) becomes:

\[ f(\bar{x}) = \frac{1}{8\pi^3} \int_{-\infty}^{\infty} \int_0^{2\pi} \int_0^{\pi} F(\rho, \theta, \phi) e^{j|\bar{x}| \rho \cos <(\bar{x}, \rho)>} \rho^2 \sin \theta d\phi d\theta d\rho \]

\[ \rho = 0 \quad \theta = 0 \quad \phi = 0 \]  

(7-2)

where

\[ |\bar{x}| = \sqrt{x_1^2 + x_2^2 + x_3^2} \]

(Rotate the k-space coordinate system so that the k_3-axis is in line with \( \bar{x} \): \( <|(\bar{x}, \rho)| = \theta \) ).

114
Then,

\[ f(\bar{x}) = \frac{1}{8\pi^3} \int_0^\infty \int_0^\pi \int_0^{2\pi} F(p, \theta, \phi) e^{-j|\bar{x}|p\cos\theta} \rho^2 \sin\theta d\phi d\theta dp \]

By considering one particular aspect angle: \( \phi_m, \theta_i \)

\[ f_{im}(x) = \frac{\sin\theta_i}{8\pi^3} \int_0^\infty \int F(p, \theta_i, \phi_m) e^{-j|\bar{x}|p\cos\theta_i} \rho^2 dp \]

With all aspect angular data,

\[ f(\bar{x}) = \sum_{i=1}^{\infty} \sum_{m=1}^{\infty} f_{im}(\bar{x}) \]

= the impulse response of the finite object at \( \bar{x} \)

Even though Equation (7-4) requires the integration over all frequencies, the integral can be approximated over three regions: the Rayleigh, the resonance, and the optical. Thus, the integral in Equation (7-4) is not impossible to be solved. There are problems associated with this approach. The Nyquist angular requirement is frequency dependent (Equation (2-17)). In order to satisfy the Nyquist angular requirement at high frequency, the signal is excessively sampled at the low frequency spectrum, but sampled only adequately at the high frequency spectrum for a finite frequency range of interest. Nonetheless, the approach is viable if one does not intend to extend the bandwidth of the approximated spatial impulse response.
With the help of the different sampling criteria, the infinite number of samples required to reconstruct the impulse response over a finite frequency range has changed to a finite number. Though the number is finite, the measurement time can be very substantial when an expanded frequency range in two dimensions, or three dimensions is required. Professor Leon Peters suggested another approach to further reduce the measurement time [15]. At high frequency, a target's frequency response is mostly contributed by its major scattering centres. If one can make a set of different canonical scattering centre measurement, then most targets' high frequency response can be built using the proper phase shift factors. For most scattering centres, their frequency responses are relatively simple. As a result, the Nyquist criteria for these centres in the frequency domain are more relaxed than complete structures. These canonical scattering centre data can be reused to reconstruct the high frequency response of other targets. In another words, after the canonical scattering centre data are available, one only measures the low frequency spectrum before the one dimensional, two dimensional, or three dimensional impulse response of an object can be reconstructed. This would be another interesting area for further exploration. However, more development work is required in all these described areas to extend this report into a more useful engineering tool.
APPENDIX A
THE DERIVATION OF EQUATION (2-13)

\[ G(\omega_1, \omega_2) = \frac{1}{R^2 \omega_1 (\omega_1^2 - 3 \omega_2^2)} \times \left\{ \begin{array}{l} 2 \omega_1 \cos \left( \frac{R \omega_1}{\sqrt{3}} \right) \cos (R \omega_2) \\ -2 \omega_1 \cos \left( \frac{2R \omega_1}{\sqrt{3}} \right) \\ -2 \sqrt{3} \omega_2 \sin \left( \frac{R \omega_1}{\sqrt{3}} \right) \sin (R \omega_2) \end{array} \right\} \]

Equation (65) of Petersen and Middleton [7]:

\[ G(\tilde{\omega}) = \left( \frac{1}{R^2} \right) \left( \frac{1}{2\sqrt{3}} \right) \int \int \frac{e^{j \tilde{\omega} \cdot \hat{x}}}{2 \sqrt{3}} \ dx \]

Equation description for the regular hexagon: (see Figure A-1)

1. \( x_2 = R \)
   \( \left( \frac{R}{\sqrt{3}} \right) < x_1 < \left( \frac{R}{\sqrt{3}} \right) \)

2. \( x_2 = -\sqrt{3} x_1 + 2R \)
   \( \left( \frac{R}{\sqrt{3}} \right) < x_1 < \left( \frac{2R}{\sqrt{3}} \right) \)
Figure A-1. Hexagonal integration limit
\( x_2 = \sqrt{3} x_1 - 2R \)
\[ \begin{cases} x_2 = -R \\ \left( \frac{R}{\sqrt{3}} \right) < x_1 < \left( \frac{-R}{\sqrt{3}} \right) \end{cases} \]

\( x_2 = -\sqrt{3} x_1 - 2R \)
\[ \begin{cases} x_2 = \sqrt{3} x_1 + 2R \\ \left( \frac{-2R}{\sqrt{3}} \right) < x_1 < \left( \frac{-R}{\sqrt{3}} \right) \end{cases} \]

then

\[ G(\tilde{\omega}) = \frac{1}{2\sqrt{3}R^2} (I_1 + I_2 + I_3) \]  \hspace{1cm} (A-1)

where

\[ \begin{align*}
I_1 &= \int_{x_1 = \frac{-2R}{\sqrt{3}}}^{\frac{R}{\sqrt{3}}} \int_{x_2 = 5} e^{j(\omega_1 x_1 + \omega_2 x_2)} dx_2 dx_1 \\
I_2 &= \int_{x_1 = \frac{-R}{\sqrt{3}}}^{\frac{R}{\sqrt{3}}} \int_{x_2 = -R}^{R} e^{j(\omega_1 x_1 + \omega_2 x_2)} dx_2 dx_1 \\
\end{align*} \]  \hspace{1cm} (A-2)
\[
I_3 = \int \int \frac{j(\omega_1 x_1 + \omega_2 x_2)}{e^{x_1 + x_2}} \, dx_2 \, dx_1
\]

Equation (A-2):

\[
I_1 = \int \int \frac{j\omega_1 x_1 \cdot j\omega_2 x_2}{e^{\sqrt{3}x_1 + 2R}} \, dx_2 \, dx_1
\]

\[
x_1 = \left(-\frac{2R}{\sqrt{3}}\right) \quad x_2 = -\sqrt{3} x_1 - 2R
\]

\[
= \int \left\{ \frac{e^{j\omega_2 x_2}}{j\omega_2} \right\} \left[ e^{j\omega_1 x_1} \right] \, dx_1
\]

\[
x_1 = \left(-\frac{2R}{\sqrt{3}}\right)
\]

\[
= \frac{1}{j\omega_2} \int \left\{ e^{j(\omega_1 + \sqrt{3} \omega_2) x_1} + 2jR w_2 e^{j(\omega_1 - \sqrt{3} \omega_2) x_1} - 2j\omega_2 X_1 \right\} \, dx_1
\]

\[
x_1 = \left(-\frac{2R}{\sqrt{3}}\right)
\]
\[
I_1 = \frac{2jR\omega_2}{e^{2jR\omega_2} - e^{-2jR\omega_2}} \begin{bmatrix} -j(\omega_1 + \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}}) & -j(\omega_1 + \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}}) \\ e & -e \end{bmatrix}
\]

\[
+ \frac{e^{2jR\omega_2}}{e^{2jR\omega_2} - e^{-2jR\omega_2}} \begin{bmatrix} -j(\omega_1 - \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}}) & -j(\omega_1 - \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}}) \\ e & -e \end{bmatrix}
\]
Equation (A-3):

\[ I_2 = \int_{-R}^{R} \int_{-R}^{R} e^{j(\omega_1 x_1 + \omega_2 x_2)} \, dx_2 \, dx_1 \]

\[ x_1 = \frac{-R}{\sqrt{3}} \quad x_2 = -R \]

\[ \left( \frac{R}{\sqrt{3}} \right) e^{j\omega_1 x_1} \, dx_1 \quad \left( \frac{R}{\sqrt{3}} \right) e^{j\omega_2 x_2} \, dx_2 \]

\[ x_1 = \left( \frac{-R}{\sqrt{3}} \right) \quad x_2 = -R \]

\[ = \left[ e^{j\omega_1 x_1} \frac{R}{\sqrt{3}} \right] \left[ e^{j\omega_2 x_2} \frac{R}{\sqrt{3}} \right] \]

\[ = \left[ \left( \frac{2}{\omega_1} \right) \sin \left( \frac{\omega_1 R}{\sqrt{3}} \right) \right] \left[ \left( \frac{2}{\omega_2} \right) \sin \left( \frac{\omega_2 R}{\sqrt{3}} \right) \right] \]

Therefore,

\[ I_2 = \left( \frac{4}{\omega_1 \omega_2^2} \right) \sin \left( \omega_2 R \right) \sin \left( \frac{\omega_1 R}{\sqrt{3}} \right) \]
Equation (A-4),

\[
I_3 = \int \int e^{j(\omega_1 x_1 + \omega_2 x_2)} \, dx_2 \, dx_1
\]

\[
x_1 = (R) \quad x_2 = \sqrt{3} x_1 - 2R
\]

\[
= \int e^{j\omega_1 x_1} \left[ e^{j\omega_2 \left(\sqrt{3} x_1 - 2R\right)} - e^{j\omega_2 \left(-\sqrt{3} x_1 + 2R\right)} \right] \, dx_1
\]

\[
x_1 = (R) \quad x_2 = \sqrt{3} x_1 - 2R
\]

\[
= \frac{1}{j\omega_2} \int e^{j\omega_1 x_1} \left[ e^{j\omega_2 \left(\sqrt{3} x_1 + 2R\right)} - e^{j\omega_2 \left(-\sqrt{3} x_1 + 2R\right)} \right] \, dx_1
\]

\[
x_1 = (R) \quad x_2 = \sqrt{3} x_1 - 2R
\]

\[
= \frac{1}{j\omega_2} \int \left[ e^{j\left(\omega_1 - \sqrt{3} \omega_2\right) x_1} + 2 j R \omega_2 \right] e^{-j\left(\omega_1 + \sqrt{3} \omega_2\right) x_1} - 2 j R \omega_2 \right] dx_1
\]

\[
x_1 = (R) \quad x_2 = \sqrt{3} x_1 - 2R
\]
\[ I_3 = \frac{2jR\omega_2}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \begin{bmatrix} \frac{j(\omega_1 - \sqrt{3}\omega_2)x1 + 2jR\omega_2}{\sqrt{3}} \\
\frac{e}{j(\omega_1 - \sqrt{3}\omega_2)} \end{bmatrix} \begin{bmatrix} 2R \\
R \end{bmatrix} - \frac{2jR\omega_2}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \begin{bmatrix} \frac{j(\omega_1 + \sqrt{3}\omega_2)x1 - 2jR\omega_2}{\sqrt{3}} \\
\frac{e}{j(\omega_1 + \sqrt{3}\omega_2)} \end{bmatrix} \begin{bmatrix} 2R \\
R \end{bmatrix} \]

Therefore,

\[ I_3 = \frac{2jR\omega_2}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \begin{bmatrix} \frac{j(\omega_1 - \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})}{e} - \frac{j(\omega_1 - \sqrt{3}\omega_2)(2R)}{\sqrt{3}} \\
\frac{e}{-e} \end{bmatrix} \]

\[ + \frac{-2jR\omega_2}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \begin{bmatrix} \frac{j(\omega_1 + \sqrt{3}\omega_2)(\frac{2R}{\sqrt{3}})}{e} - \frac{j(\omega_1 + \sqrt{3}\omega_2)(\frac{R}{\sqrt{3}})}{e} \end{bmatrix} \]
Now,

\[ I_1 + I_3 \]

\[ = \frac{1}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ e^{-j[(\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2]} + e^{j[(\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2]} \right\} \]

\[ - e^{-j[(\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2]} - e^{j[(\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2]} \right\} \]

\[ + \frac{1}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ e^{-j[(\omega_1 - \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} + 2R\omega_2]} + e^{j[(\omega_1 - \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} + 2R\omega_2]} \right\} \]

\[ - e^{-j[(\omega_1 - \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} + 2R\omega_2]} - e^{j[(\omega_1 - \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} + 2R\omega_2]} \right\} \]

\[ = \frac{1}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ 2\cos \left[ (\omega_1 + \sqrt{3}\omega_2)\frac{2R}{\sqrt{3}} - 2R\omega_2 \right] \right\} \]

\[ - 2\cos \left[ (\omega_1 + \sqrt{3}\omega_2)\frac{R}{\sqrt{3}} - 2R\omega_2 \right] \right\} \]

(This expression continues on the next page.)
\[ + \frac{1}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ 2\cos \left[ \frac{(\omega_1 - \sqrt{3}\omega_2) R + 2R\omega_2}{\sqrt{3}} \right] \right. \\

\left. - 2\cos \left[ \frac{(\omega_1 - \sqrt{3}\omega_2) 2R + 2R\omega_2}{\sqrt{3}} \right] \right\} \]

[Sine: \( \cos \alpha - \cos \beta = -2 \sin \frac{1}{2}(\alpha + \beta) \sin \frac{1}{2}(\alpha - \beta) \)]

\[ = -\frac{4}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ \sin \frac{1}{2} \left[ \frac{3R}{\sqrt{3}}(\omega_1 + \sqrt{3}\omega_2) - 4R\omega_2 \right] \sin \frac{1}{2} \left[ \frac{R}{\sqrt{3}}(\omega_1 + \sqrt{3}\omega_2) \right] \right\} \]

\[ + \frac{4}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ \sin \frac{1}{2} \left[ \frac{3R}{\sqrt{3}}(\omega_1 - \sqrt{3}\omega_2) + 4R\omega_2 \right] \sin \frac{1}{2} \left[ \frac{R}{\sqrt{3}}(\omega_1 - \sqrt{3}\omega_2) \right] \right\} \]

\[ = -\frac{4}{\omega_2(\omega_1 + \sqrt{3}\omega_2)} \left\{ \sin \left( \frac{\sqrt{3}R\omega_1}{2} - \frac{R\omega_2}{2} \right) \sin \left( \frac{R\omega_1}{2\sqrt{3}} + \frac{R\omega_2}{2} \right) \right\} \]

\[ + \frac{4}{\omega_2(\omega_1 - \sqrt{3}\omega_2)} \left\{ \sin \left( \frac{R\omega_1}{2\sqrt{3}} + \frac{R\omega_2}{2} \right) \sin \left( \frac{R\omega_1}{2\sqrt{3}} - \frac{R\omega_2}{2} \right) \right\} \]

126
[Since: \( \sin \alpha \sin \beta = \frac{1}{2} \cos (\alpha - \beta) - \frac{1}{2} \cos (\alpha + \beta) \)]

\[
\frac{-2}{\omega_2(\omega_1 + \sqrt{3} \omega_2)} \left\{ \cos \left[ \left( \sqrt{3} - \frac{1}{\sqrt{3}} \right) \left( \frac{R \omega_1}{2} \right) - R \omega_2 \right] - \cos \left[ \left( \sqrt{3} + \frac{1}{\sqrt{3}} \right) \frac{R \omega_1}{2} \right] \right\}
\]

\[+ \frac{2}{\omega_2(\omega_1 - \sqrt{3} \omega_2)} \left\{ \cos \left[ \left( \sqrt{3} - \frac{1}{\sqrt{3}} \right) \left( \frac{R \omega_1}{2} \right) + R \omega_2 \right] - \cos \left[ \left( \sqrt{3} + \frac{1}{\sqrt{3}} \right) \frac{R \omega_1}{2} \right] \right\}
\]

\[= \frac{-2}{\omega_2(\omega_1 + \sqrt{3} \omega_2)} \left\{ \cos \left( \frac{R \omega_1}{\sqrt{3}} - R \omega_2 \right) - \cos \left( \frac{2R \omega_1}{\sqrt{3}} \right) \right\}
\]

\[+ \frac{2}{\omega_2(\omega_1 - \sqrt{3} \omega_2)} \left\{ \cos \left( \frac{R \omega_1}{\sqrt{3}} + R \omega_2 \right) - \cos \left( \frac{2R \omega_1}{\sqrt{3}} \right) \right\}
\]
\[
\begin{align*}
&= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ (\omega_1 - \sqrt{3}\omega_2) \cos \left( \frac{R\omega_1}{\sqrt{3}} - R\omega_2 \right) \\
&\quad - (\omega_1 - \sqrt{3}\omega_2) \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \\
&\quad - (\omega_1 + \sqrt{3}\omega_2) \cos \left( \frac{R\omega_1}{\sqrt{3}} + R\omega_2 \right) \\
&\quad + (\omega_1 + \sqrt{3}\omega_2) \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \right\} \\
&= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \omega_1 \left[ \cos \left( \frac{R\omega_1}{\sqrt{3}} - R\omega_2 \right) - \cos \left( \frac{R\omega_1}{\sqrt{3}} + R\omega_2 \right) \right] \\
&\quad - \sqrt{3} \omega_2 \left[ \cos \left( \frac{R\omega_1}{\sqrt{3}} - R\omega_2 \right) + \cos \left( \frac{R\omega_1}{\sqrt{3}} + R\omega_2 \right) \right] \\
&\quad + 2 \sqrt{3} \omega_2 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \right\}
\end{align*}
\]

Since: \[
\begin{align*}
\cos \alpha + \cos \beta &= 2 \cos \frac{1}{2} (\alpha + \beta) \cos \frac{1}{2} (\alpha - \beta) \\
\cos \alpha - \cos \beta &= -2 \sin \frac{1}{2} (\alpha + \beta) \sin \frac{1}{2} (\alpha - \beta)
\end{align*}
\]
\[
\begin{align*}
\text{Therefore, } I_1 + I_3 &= \\
&= \frac{-2}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \omega_1 \left[ 2 \sin \frac{1}{2} \left( \frac{2R\omega_1}{\sqrt{3}} \right) \sin \frac{1}{2} \left( 2R\omega_2 \right) \right] \\
&\quad - \sqrt{3} \omega_2 \left[ 2 \cos \frac{1}{2} \left( \frac{2R\omega_1}{\sqrt{3}} \right) \cos \frac{1}{2} \left( -2R\omega_2 \right) \right] \\
&\quad + 2\sqrt{3} \omega_2 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \right\} \\
\text{Now, } I_1 + I_2 + I_3 &= \\
&= \frac{-4}{\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \omega_1 \sin \left( \frac{R\omega_1}{\sqrt{3}} \right) \sin \left( R\omega_2 \right) \\
&\quad - \sqrt{3} \omega_2 \cos \left( \frac{R\omega_1}{\sqrt{3}} \right) \cos \left( R\omega_2 \right) \\
&\quad + \sqrt{3} \omega_2 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \right\} \\
&= \frac{-4}{\omega_1\omega_2(\omega_1^2 - 3\omega_2^2)} \left\{ \sqrt{3}\omega_2\omega_1 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) + \omega_1^2 \sin \left( \frac{R\omega_1}{\sqrt{3}} \right) \sin \left( R\omega_2 \right) \\
&\quad - \sqrt{3} \omega_1 \omega_2 \cos \left( \frac{R\omega_1}{\sqrt{3}} \right) \cos \left( R\omega_2 \right) \\
&\quad - \omega_1^2 \sin \left( \omega_2 R \right) \sin \left( \frac{\omega_1 R}{\sqrt{3}} \right) \\
&\quad + 3 \omega_2^2 \sin \left( \omega_2 R \right) \sin \left( \frac{\omega_1 R}{\sqrt{3}} \right) \right\}
\end{align*}
\]
\[
= \frac{-4}{\omega_1 (\omega_1^2 - 3\omega_2^2)} \left\{ \sqrt{3} \omega_1 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) - \sqrt{3} \omega_1 \cos \left( \frac{R\omega_1}{\sqrt{3}} \right) \cos (R\omega_2) \right. \\
+ \left. 3 \omega_2 \sin (\omega_2 R) \sin \left( \frac{\omega R}{\sqrt{3}} \right) \right\}
\]

Equation (A-1),

\[
G(\tilde{\omega}) = \frac{1}{2\sqrt{3}R} \left( I_1 + I_2 + I_3 \right)
\]

\[
= \frac{1}{2\sqrt{3}R} \frac{1}{\omega_1 (\omega_1 - 3\omega_2)} \times \left\{ \begin{array}{c}
2 \omega_1 \cos \left( \frac{R\omega_1}{\sqrt{3}} \right) \cos (R\omega_2) \\
- 2 \omega_1 \cos \left( \frac{2R\omega_1}{\sqrt{3}} \right) \\
- 2 \sqrt{3} \omega_2 \sin \left( \frac{R\omega_1}{\sqrt{3}} \right) \sin (R\omega_2) \end{array} \right\}
\]

= Equation (2-13).
APPENDIX B

PROGRAMS DEVELOPED
Figure B-1. Flow chart showing the relationship among the written programs
**PROGRAM:** RECONS3LD

**THIS PROGRAM WILL TAKE A DATA SET: 'COEFF' IN REAL AND IMAGINARY FORMAT AND INTERPOLATE THE POINTS IN BETWEEN SAMPLES USING SAMPLING THEOREM APPROACH.** PRESENTLY, THIS PROGRAM IS ALLOWED TO TAKE 100 SAMPLES. IF MORE IS REQUIRED, PLEASE CHANGE THE ARRAY SIZE OF COEFF AND THE VALUE OF NUM.

**SIZE**: THE ARRAY SIZE = 2*(NUMBER OF SAMPLES) + 1

**NUM**: NUMBER OF SAMPLES

**XINC**: INCREMENT

THE SAMPLES ARE TAKEN AT DELTA FREQUENCY OF 'FRED'.

THE FREQUENCY RANGE FOR INTERPOLATION IS SPECIFIED BY 'INITIAL' AND 'LAST'. THE NUMBER OF KOIVYS IN 'DATA' IS 'NRF'. THE OUTPUT FILE IS SPECIFIED BY 'FNAME2'.

**THIS PROGRAM WILL LINK WITH SINC, JPLOT, SUMLD AND 'PLOTLIB**

```
REAL INITIAL, LAST
COMPLEX COEFF(201), SUM DATA(200)
CHARACTER*10 FNAME1, FNAME2
CHARACTER COM
NUM=100
MISE=2*NUM + 1
STEP=200
INITIAL=6.E7
LAST=12.E9
XINC=(LAST-INITIAL)/(STEP-1)
PI=3.14159265

THE SUMMATION IS FROM -N TO +N
COEFF - THE ARRAY OF COEFFICIENTS FOR ITH TERM IN THE SUMMATION
WHERE THE INDEX -N IS REPRESENTED BY 1
+N IS REPRESENTED BY 2*N + 1
T - ASSUMED CUTOFF TIME = 1/(2*SAMPLING FREQUENCY)

DATA STRUCTURE OF FNAME1
N : NUMBER OF POINTS IN THE FILE (*)
FMIN : SMALLEST FREQUENCY USED (*)
FREQ : SAMPLING FREQUENCY (*)
COEFF(I) : REAL - IMAGINARY (*)

TYPE *, 'TYPE FILE NAME CONTAINING COMPLEX COEFFICIENTS'
ACCEPT 2-FNAME1
```

2 FORMAT (A10)
OPEN (UNIT=8, NAME=FNAME1, TYPE='OLD')
READ(8, *) N
READ(8, *) FMIN
READ(8, *) FREQ
T=1/(2.*FREQ)
IF (N.GT.NUM) GO TO 8888
```
NDIM=2*N +1

**INITIALIZATION**

DO 5 I=1,MSIZE
   COEFF(I)=(0.,0.)
5 CONTINUE

READ(8,*) COEFF(NUM)

DO 10 I=1,NS+1
   READ(8,*) COEFF(NUM+I)

**FILL IN COMPLEX CONJUGATE FOR THE NEGATIVE FREQUENCY**

   COEFF(NUM+I-1) = CONJ (COEFF(NUM+I))
10 CONTINUE

CLOSE (UNIT=8,DISP='SAVE')

**SET ANY FREQUENCY INTERPOLATION SPECIFIED BELOW FMIN TO ZERO**

NZERO: NUMBER OF ZEROS TO BE PLACED

IF (FMIN.LE.FREQ) DO TO 127
   NZERO=INT (FMIN/XINC)
   DO 120 I=1,NZERO
      DATA(I)=(0.,0.)
120 CONTINUE

**STARTS THE INTERPOLATION**

THE SUMMATION IS PERFORMED BY FUNCTION SUMID)

ILAST=INT ((N-1)*FREQ/XINC)
IF (ILAST.LT.NSTEP) INDEX=ILAST
IF (ILAST.GE.NSTEP) INDEX=NS+1

DO 20 I=NZERO+1,INDEX
      DATA(I) = SUMID (((I-1)*XINC+INITIAL)*T*2.*PI/COEFF,MSIZE,NDIM)
20 CONTINUE

**SET ANY SPECIFIED FREQUENCY INTERPOLATION HIGHER THAN FMIN+N*FREQ**

TO ZERO

DO 133 I=INDEX +1,NS+1
      DATA(I)=(0.,0.)
133 CONTINUE

**PLOT AND OPTIONAL WRITE INTO A FILE: FNAME2**

CALL JPLOT (DATA,NSTEP,INITIAL,XINC,0.-NS+1)

WRITE (6,*) 'WRITE THE INTERPOLATED DATA INTO A FILE?'
WRITE (6,*) '1) YES, IN AMPLITUDE AND PHASE (RADIANS) FORM'
WRITE (6,*) '2) YES, IN REAL AND IMAGINARY FORM'
WRITE (6,*) '3) NO, FORGET IT!'
ACvEr
25,COM1
IF (COM1.EQ.'3') GO TO 9999
IF (COM1.NE.'1').AND.(COM1.NE.'2')) GO TO 23
WRITE(6,*) 'STORAGE FILE NAME:'
ACCEPT 30,FRAME2
30 FORMAT(A10)
IF (COM1.EQ.'2') GO TO 35
C
CONVERT TO AMPLITUDE (LINEAR) AND PHASE (RADIAN)
DO 33 I=1,NSTEP
   XREAL=ABS(DATA(I))
   XIMAG=ATAN2(ALPAG(DATA(I)),REAL(DATA(I)))
   DATA(I)=CMPLX(XREAL,XIMAG)
33 CONTINUE
C
OUTPUT TO FRAME2
C STRUCTURE OF FRAME2:
C DATA(I): FREE COMPLEX FORMAT(*)
C
35 OPEN(UNIT=8,NAME=FRAME2,TYPE='NEW')
DO 40 I=1,NSTEP
   WRITE(8,*) DATA(I)
40 CONTINUE
CLOSE(UNIT=8,DISP='SAVE')
GO TO 9999
8888 WRITE(6,*) 'ERROR: NOT ENOUGH SPACE SPECIFIED IN THE PROGRAM'
9999 STOP
END
SUBROUTINE JPLT

THIS SUBROUTINE WILL DO RECTANGULAR PLOT ON A COMPLEX ARRAY: 'DATA'
FOR REAL AND IMAGINARY PLOT OR MAGNITUDE (LINEAR) AND PHASE (Radian)
PLOT.

DATA : ARRAY NAME
NENINT : THE ARRAY SIZE
FMIN : SMALLEST ELEMENT OF THE ABSISSOR
FINCR : THE INCREMENT SIZE
NSTART : THE START PLOTTING INDEX
NLAST : THE LAST PLOTTING INDEX
IF (NLAST.GT.NEINT) THE LAST POINT PLOTTED IS NPOINT

THIS REQUIRES THE SUPPORT OF 'PLOTLIB'.

SUBROUTINE JPLT (DATA, NPOINT, FMN, FINCR, NSTART, NLAST)
CHARACTER COM
COMPLEX DATA(NPOINT)
DIMENSION XAXIS1(N000), YAXIS2(N000), XAXIS(N000)
PI=3.14159265
SIZE=N000

INITIALIZATION

DO 3 I=1, SIZE
  XAXIS1(I)=0.
  YAXIS2(I)=0.
  XAXIS(I)=0.
3 CONTINUE

WRITE (6,*) 'DO YOU WANT REAL AND IMAGINARY PLOT? Y/N'
ACCEPT 6, COM
IF (NLAST.GT.NPOINT) NLAST=NPOINT
IF (COM.NE. 'Y') GO TO 15
IF (COM.NE. 'N') GO TO 5

REAL AND IMAGINARY PREPARATION

DO 10 I=1, NPOINT
  XAXIS1(I)=REAL (DATA(I+NSTART))
  YAXIS2(I)=AIMAG (DATA(I+NSTART))
  XAXIS(I)=(I-1)*FINCR + FMN
10 CONTINUE

generate amplitude and phase

DO 20 I=1, NPOINT
  XAXIS(I)=(I-1)*FINCR + FMN
20 CONTINUE
YAXIS1(I) = CABS(DATA(I+NSTART))
IF (REAL(DATA(I+NSTART)) .EQ. 0.) GO TO 18
YAXIS2(I) = ATAN2(AIMAG(DATA(I+NSTART)),REAL(DATA(I+NSTART)))
GO TO 20
18 IF (AIMAG(DATA(I+NSTART)) .EQ. 0.) YAXIS2(I) = 0.
IF (AIMAG(DATA(I+NSTART)) .LT. 0.) YAXIS2(I) = PI/2
IF (AIMAG(DATA(I+NSTART)) .GT. 0.) YAXIS2(I) = -PI/2
20 CONTINUE
25 CALL PLTRMG(XAXIS, YAXIS1, 9000, NP, 1.0, 1) CALL PLTRMG(XAXIS, YAXIS2, 9000, NP, 1.0, 1)
RETURN
END
COMPLEX FUNCTION: SUMLD

This subroutine will do summation

X - New parameter

Y - Coefficient array

MSIZE - Size of the coefficient array

N - Number of coefficient array elements that has non-zero values

Note: Both MSIZE and N are odd numbers

Complex function SUMLD(X, Y, MSIZE, N)

EXTERNAL SINC

COMPLEX Y(MSIZE)

PI = 3.14159265

MED = MSIZE/2 + 1

SUMLD = Y(MED) * SINC(X)

DO 5 I = 1, N/2

      SUMLD = SUMLD + Y(MSIZE/2-I+1) * SINC(X+I*PI)
      + Y(MSIZE/2+I+1) * SINC(X-I*PI)

  5      CONTINUE

RETURN

END
FUNCTION: SINC

THIS SUBROUTINE WILL CALCULATE SIN(X)/X
WHERE X IS ASSUMED TO BE IN RADIANS

FUNCTION SINC(X)
IF (X.EQ.0) GO TO 5
SINC=SIN(X)/X
GO TO 10
5 SINC=1.
10 RETURN
END
PROGRAM : PFGRID

THIS PROGRAM WILL GENERATE ALL THE ANGLES AND FREQUENCIES
FOR THE GRID IN THE FREQUENCY DOMAIN. THE DATA ARE
ARRANGED IN ORDER FROM SMALLEST ANGLE TO THE LARGEST ANGLE.
(0 TO 2*PI) THE SMALLEST FREQUENCY
RADIUS IS STORED. SO AT THE TIME OF MEASUREMENT, ONLY
MULTIPLICATION OF THE RADIUS IS NECESSARY.

THE GRID POINTS ARE WRITE INTO FILE: 'OUT.DAT'
THE DATA FORMAT OF 'OUT.DAT';
FMN-FMAX : FREQUENCY RANGE SPECIFIED (2E15.8)
FDELTA : SAMPLING FREQUENCY (E15.8)
NPOINT : NUMBER OF POINTS IN THE FILE (18)
ISO,RADIUS : ISO=1, NON-ISOTROPIC SAMPLING IS USED (A2)
ISO=7, ISOTROPIC SAMPLING IS USED
: RADIUS: RADIUS OF THE ISOTROPIC CELL IN MM (E15.8)
V1,V2 : VECTORS USED TO DEFINE THE SAMPLING LATTICE (E15.8)
U1,U2 : VECTORS USED TO DEFINE THE PERIODIC LATTICE (E15.8)
DATA(*,1),DATA(*,2) : DATA (E15.8)
DATA(*,1) IS THE MEASUREMENT ANGLE AND FREQUENCY ARRAY
DATA(*,2) IS THE INDEX SPECIFYING HOW MANY UNITS OF
V1,V2 ARE USED

WARNING: SIZE OF DATA IS (10000,2)

LINK PFGRID,GEN1,GEN2,SEARCH,INSERT,FUSH,NORMAL, 'SSP

COMPLEX V1,V2,V3,V4,U1,U2,DATA(10000,2)
DIMENSION WORK1(2),WORK2(2)
REAL M1,M2,M1,M2,MARGIN,MATRIX(4)
CHARACTER ISO
MSIZE=10000
PI=3.14159265

START WORKING

WRITE (6,*) ' LOWEST AND HIGHEST OPERATING FREQUENCY IN HERTZ'
ACCEPT *, FMN-FMAX
IF (FMN.GE.FMAX) GO TO 2

WRITE (6,*) ' DO YOU WANT TO USE ISOTROPIC SAMPLING? T/F '
ACCEPT 4,ISO

FORMAT (A1)
IF (ISO.EQ.'F') GO TO 5
IF (ISO.NE.'T') GO TO 3
WRITE (6,*) ' DIAMETER OF NORMALIZATION IN MM?'
ACCEPT *,RADIUS

DEFINITION OF ISOTROPIC VECTORS
CASE OF PARALLELPIPEDIC CONFINEMENT

WRITE (6,*) 'THE OBJECT IS ASSUMED TO BE CONFINED TO A PARALLELOGRAM'
WRITE (6,*) 'NORMALIZATION FACTOR IN MH'
ACCEPT *,RADIUS
WRITE (6,*) 'VECTOR1 IN X,Y:'
ACCEPT *,X1,Y1
WRITE (6,*) 'VECTOR2 IN X,Y:'
ACCEPT *,X2,Y2

WRITE (6,*) 'SAFETY MARGIN FACTOR ON THE TIME WAVEFORM'

MINV WILL DO MATRIX INVERSION. THIS SUBROUTINE IS IN SSP

CALL MINV(MATRIX,2,WORK1,WORK2)

NOTE THAT V1 AND V2 IS TAKEN FROM THE TRANSPOSE
OF THE INVERSE OF MATRIX. THIS IS TO BE CONSISTENCE WITH
DOT PRODUCT OF U(I) AND V(J) = DELTA(I,J)

U IS IN SPATIAL DOMAIN
V IS IN FREQUENCY DOMAIN
ALSO VI=(X-COMPONENT,Y-COMPONENT)

VI=CMPLX(MATRIX(1),MATRIX(3))
V2=CMPLX(MATRIX(2),MATRIX(4))
MAG1=CABS(V1)
PHASE1=ATAN2(IMAG(V1),REAL(V1))
MAG2=CABS(V2)
PHASE2=ATAN2(IMAG(V2),REAL(V2))
THETA=ABS(PHASE1-PHASE2)

PHASES ARE NORMALIZED TO THE RANGE 0:2*PI

CALL NORMAL(PHASE1)
CALL NORMAL(PHASE2)

SFACDR: FREQUENCY SAMPLING FACTOR
SFACOR=(3.EL1)/(RADIUS*MARGIN)
IF ((SFACOR.LE.FMIN) OR. (FMIN.EQ.0.)) FDELA=SFACOR
IF ((SFACOR.GT.FMIN) AND. (FMIN.NE.0.)) FDELA=FMIN
RADIUS=1/(FDELA*2.)
MAX1=(FMAX/FDELA)

STORE THE FIRST TWO VECTORS INTO THE ARRAY 'DATA'
IF (PHASE2.LT.PHASE1) GO TO 11
DATA(1,1)=CMPLX(PHASE1,MAG1)
DATA(1,2)=CMPLX(1.0)
DATA(2,1)=CMPLX(PHASE2,MAG2)
DATA(2,2)=CMPLX(0.1)
GO TO 12
11 DATA(1,1)=CMPLX(PHASE2,MAG2)
DATA(1,2)=CMPLX(0.1)
DATA(2,1)=CMPLX(PHASE1,MAG1)
DATA(2,2)=CMPLX(1.0)

NEINT=2

TO GENERATE THE FIRST QUADRANT POINTS
POINTS ARE GENERATED AND STORED IN SUBROUTINE GEN1 AND GEN2
IF (THET.LE.PI/2.)
& CALL GEN1 (V1, V2, MAX1, MAX1, DATA, MSIZE, NEINT, 1, 1)
IF (THET.GT.PI/2.)
& CALL GEN2 (V1, V2, MAX1, MAX1, DATA, MSIZE, NEINT, 1, 1)
IF (NEINT.GT.MSIZE) GO TO 888
V3=V1
PHASE3 = ATAN2 (AIMAG (V3), REAL (V3))
CALL NORMAL (PHASE3)
V3=CMPLX (PHASE3, ABS (V3))

SEARCH FOR LOCATION :LOC TO PLACE THE POINT
LOC=SEARCH (V3, DATA, MSIZE, NEINT)

INSERT THE POINT INTO THE PROPER LOCATION
CALL INSERT (LOC, NEINT, DATA, MSIZE, V3, -1., 0.)

TO GENERATE THE 2ND QUADRANT POINTS
IF (THET.LE.PI/2.)
& CALL GEN2 (V1, V2, MAX1, MAX1, DATA, MSIZE, NEINT, -1, 1)
IF (THET.GT.PI/2.)
& CALL GEN1 (V1, V2, MAX1, MAX1, DATA, MSIZE, NEINT, -1, 1)
IF (NEINT.GT.MSIZE) GO TO 888
V4=V2
PHASE4 = ATAN2 (AIMAG (V4), REAL (V4))
CALL NORMAL (PHASE4)
V4=CMPLX(PHASE4,CMPLX(V4))
LOC-SEARCH(V4,DATA,MSIZE,NQINT)
CALL INSERT (LOC,NQINT,DATA,MSIZE,V4.0.,-1.)

NOW TO GENERATE THE OTHER TWO QUADRANTS

IF (THETA.LT.PI/2.)
& CALL GEN2 (V1,V2,MAX1,MAX1,DATA,MSIZE,NQINT,1,-1)
IF (THETA.GT.PI/2.)
& CALL GEN1 (V1,V2,MAX1,MAX1,DATA,MSIZE,NQINT,1,-1)
IF (NQINT.GT.MSIZE) GO TO 888
IF (THETA.LT.PI/2.)
& CALL GEN1 (V1,V2,MAX1,MAX1,DATA,MSIZE,NQINT,-1,-1)
& CALL GEN2 (V1,V2,MAX1,MAX1,DATA,MSIZE,NQINT,-1,-1)
IF (NQINT.GT.MSIZE) GO TO 888

OUTPUT GRID POINTS INTO FILE: 'OUT.DAT'

OPEN(UNIT=8,NAME='OUT',TYPE='NEW')
WRITE (8,301) FMN,FMX
301 FORMAT (E15.8)
WRITE (8,305) FDELTA
305 FORMAT (E15.8)
WRITE (8,310) NQINT
310 FORMAT (I8)
WRITE (8,315), ISO,RADIUS
315 FORMAT (E15.8)
WRITE (8,320) VI,V2
320 FORMAT (2E15.8)
DO 20 J=1,NQINT
WRITE (8,320) DATA(J,1),DATA(J,2)
20 CONTINUE
CLOSE (UNIT=8,DISP='SAVE')
GO TO 999
888 WRITE (6,'*') 'ERROR: SPECIFIED ARRAY SIZE TOO SMALL!'
999 STOP
END
SUBROUTINE: GER1

THIS SUBROUTINE WILL GENERATE ALL THE GRID POINTS WITHIN THE TWO VECTORS V1, V2 ON THE TWO DIMENSIONAL PLANE. THAT IS IF THE ANGLE IS ACUTE.

V1 : FIRST VECTOR
V2 : SECOND VECTOR
MAX : MAXIMUM FREQUENCY IN EACH VECTOR DIRECTION
AMAX : ABSOLUTE MAXIMUM FREQUENCY
DFILE : DATA ARRAY FOR STORAGE
MSIZE : SIZE OF DFILE
M : NUMBER OF POINTS GENERATED
NLSIGN : SIGN OF N1 (N1*V1)
N2SIGN : SIGN OF N2 (N2*V2)

THIS REQUIRE THE SUPPORT OF SEARCH, NORMAL, INSERT EXTERNAL SEARCH

COMPLEX DFILE(MSIZE,2),VECTOR,V1,V2
REAL MAG, MAX

CALCULATE THE NUMBER OF UNITS IN EACH VECTOR DIRECTIONS

NL=INT(MAX/(CABS(V1))) + 1
N2=INT(MAX/(CABS(V2))) + 1
DO 5 J=1,N2
   DO 10 I=1,NL
      VECTOR=J*V1*NL SIGN + I*V2*N2 SIGN
      MAG=CABS(VECTOR)
      IF (MAG.GT.MAX) GO TO 5
      PHASE=ATAN2(I*AIMAG(VECTOR),REAL(VECTOR))
      CALL NORMAL(PHASE)
      VECTOR=CMPLX(PHASE,MAG)
   10 CONTINUE

SEARCH THE LOCATION

LOC=SEARCH(VECTOR,DFILE,MSIZE,M)
RJ=FLOAT(J)
RI=FLOAT(I)

INSERT THE DATA

IF ((NL SIGN .LT. 0) .AND. (N2 SIGN .LT. 0))
   CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
IF ((NL SIGN .LT. 0) .AND. (N2 SIGN .LT. 0))
   CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
IF ((ISIGN .GT. 0) .AND. (N2 SIGN .LT. 0))
   CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
IF ((ISIGN .LT. 0) .AND. (N2 SIGN .LT. 0))
   CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
CALL INSERT (LOC, M, DFILE, MSIZE, VECTOR, RJ, RI)

10 CONTINUE
5 CONTINUE
RETURN
END
SUBROUTINE: GEN2

This subroutine will generate points between vectors V1, V2 if the angle between them is more than 90 degrees but less than 180 degrees.

V1: First vector
V2: Second vector
MAX: Maximum frequency in each vector direction
AMAX: Absolute maximum frequency
DFILE: Data array for storage
MSIZE: Size of DFILE
M: Number of points generated
NSIGN: Sign of N1 (N1*V1)
N2SIGN: Sign of N2 (N2*V2)

This requires the support of GEN1, INSERT, SEARCH.

SUBROUTINE GEN2(V1, V2, MAX, AMAX, DFILE, MSIZE, M, NSIGN, N2SIGN)
EXTERNAL SEARCH
COMPLEX DFILE(MSIZE, 2), VECDIR, V1, V2
REAL W, MAX
LOGICAL SAME
PI = 3.14159265

THETA = ABS(ATAN2(AIMAG(V1), REAL(V1)) - ATAN2(AIMAG(V2), REAL(V2)))
MAX = MAX/SIN(THETA)

CALL GEN1(V1, V2, MAX, AMAX, DFILE, MSIZE, M, NSIGN, N2SIGN)
IF (THETA .LT. PI/2.) GO TO 555
N1CORRECT = 0
N2CORRECT = 0
IF (MOD(MAX(CABS(V1)), 2) .GT. 0.) N1CORRECT = 1
IF (MOD(MAX(CABS(V2)), 2) .GT. 0.) N2CORRECT = 1
N1 = INT(MAX(CABS(V1))) + N1CORRECT
N2 = INT(MAX(CABS(V2))) + N2CORRECT
I2 = INT(MAX(CABS(V2)))

TO check out which region are the vectors in

VECTOR = V1*NSIGN + V2*N2SIGN
IF ((REAL(VECTOR)*AIMAG(VECTOR)).GT.0.) SAME = .TRUE.
IF ((REAL(VECTOR)*AIMAG(VECTOR)).LT.0.) SAME = .FALSE.
IF ((REAL(VECTOR)*AIMAG(VECTOR)).NE.0.) GO TO 3
VECTOR = 2. * V1*NSIGN + V2*N2SIGN
IF ((REAL(VECTOR)*AIMAG(VECTOR)).GT.0.) SAME = .TRUE.
IF ((REAL(VECTOR)*AIMAG(VECTOR)).LT.0.) SAME = .FALSE.

3 CHECK = 0.
DO 5 J = 1, N2
5 CONTINUE

IF (CHECK .GE. 2.) GO TO 50
VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
MAG=CABS(VVECTOR)

MAKE SURE THAT CHECK IS INCREMENTED PROPERLY

IF (NOT SAME).AND.((REAL(VVECTOR)*AIMAG(VVECTOR)).GT.0))
  CHECK=CHECK + 1.
& IF ((SAME).AND.((REAL(VVECTOR)*AIMAG(VVECTOR)).LT.0))
  CHECK=CHECK + 1

MAKE SURE THE POINT GENERATED IS INSIDE THE CIRCLE

IF (MAG.GT.AMAX) GO TO 5

MAKE SURE THAT CHECK IS INCREMENTED PROPERLY

IF (NOT SAME).AND.((REAL(VVECTOR)*AIMAG(VVECTOR)).GT.0))
  CHECK=CHECK - 1.
& IF ((SAME).AND.((REAL(VVECTOR)*AIMAG(VVECTOR)).LT.0))
  CHECK=CHECK - 1.
& PHASE=ATAN2(AIMAG(VVECTOR),REAL(VVECTOR))
CALL NORMAL(PHASE)
VECTOR=CPLX(PHASE,MAG)
LOC=SEARCH(VVECTOR,DFILE,MSIZE,10)
RJ=FLOAT(J)
RI=FLOAT(I)
IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
  CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
& IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
  CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
& IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
  CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
& IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
  CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)

10 CONTINUE
5 CONTINUE
50 J=J-2
DO 15 I=12,M2
   DO 20 J=1,M1
      VECTOR=J*V1*N1SIGN +I*V2*N2SIGN
      MAG=CABS(VVECTOR)
      IF (MAG.GT.AMAX) GO TO 15
      PHASE=ATAN2(AIMAG(VVECTOR),REAL(VVECTOR))
      CALL NORMAL(PHASE)
      VECTOR=CPLX(PHASE,MAG)
      LOC=SEARCH(VVECTOR,DFILE,MSIZE,10)
      RJ=FLOAT(J)
      RI=FLOAT(I)
      IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,-RI)
      IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,-RJ,RI)
      IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,-RI)
      IF ((N1SIGN.LT.0).AND.((N2SIGN.LT.0)))
        CALL INSERT(LOC,M,DFILE,MSIZE,VECTOR,RJ,RI)
& CALL INSET (LOC, M, DFLE, MSIZE, VECTOR, -RI, RJ)
& IF ((N.SIGN.GE.0).AND. (NZSIGN.LT.0))
& CALL INSET (LOC, M, DFLE, MSIZE, VECTOR, RJ, -RI)
& IF ((N.SIGN.GT.0).AND. (NZSIGN.GT.0))
& CALL INSET (LOC, M, DFLE, MSIZE, VECTOR, RJ, RI)
20 CONTINUE
15 CONTINUE
555 RETURN
END
FUNCTION: SEARCH

This function will search the location where the element fits into a data array. The array assumes at least two members. The element may have 3 possibilities:

1) The real part is the same in the returned location
2) The real part is the same in the returned location +1
3) The real part is between the above two

This function employs binary search technique to locate

ELEMENT : Complex element to be placed with priority of the real part over the imaginary part
DFILE : Array file to be searched and inserted
MSIZE : Size of DFILE
LAST : Number of element in DFILE

FUNCTION SEARCH (ELEMENT, DFILE, MSIZE, LAST)

COMPLEX DFILE (MSIZE, 2), ELEMENT
LOC2 = LAST
LOC1 = 1
IF (REAL (ELEMENT) .LT. REAL (DFILE (LOC1, 1))) GO TO 147
IF (REAL (ELEMENT) .GT. REAL (DFILE (LOC2, 1))) GO TO 140
LOCM = (LOC1 + LOC2) / 2
IF (LOCM .EQ. LOC1) GO TO 150
IF (REAL (ELEMENT) .LT. REAL (DFILE (LOCM, 1))) GO TO 125
IF (REAL (ELEMENT) .GT. REAL (DFILE (LOCM, 1))) GO TO 130
GO TO 145

LOC2 = LOC1
GO TO 120

LOC1 = LOC1
GO TO 120

SEARCH = LOC2
GO TO 155

SEARCH = LOC1
GO TO 155

SEARCH = LOC1 - 1
GO TO 155

SEARCH = LOC1
RETURN
END
SUBROUTINE: INSERT

THIS SUBROUTINE WILL INSERT THE ELEMENT INTO THE DFILE
IF THE ELEMENT HAS A DIFFERENT PHASE OR A SMALLER
MAGNITUDE. THE PHASE HAS A SENSITIVITY SPECIFIED BY THE ERROR.

NSTART : LOCATION OF THE ELEMENT
LAST : NUMBER OF ELEMENTS IN DFILE
DFILE : ARRAY FILE TO BE INSERTED
MSIZE : DIMENSION OF DFILE
ELEMENT : COMPLEX ELEMENT TO BE INSERTED
VINDEX : NUMBER OF V1 USED
V2INDEX : NUMBER OF V2 USED

SUBROUTINE INSERT(NSTART, LAST, DFILE, MSIZE, ELEMENT, VINDEX, V2INDEX)
COMPLEX DFILE(MSIZE, 2), ELEMENT
PI=3.14159265
ERROR=0.01*PI/180.
IF (NSTART.NE.LAST) GO TO 200
IF (ABS(REAL(ELEMENT)-REAL(DFILE(LAST,1))).LT.ERROR) GO TO 290
LAST = LAST +1
DFILE(LAST,1) = ELEMENT
DFILE(LAST,2) = COMPLEX(VINDEX, V2INDEX)
GO TO 295
200 IF (NSTART.EQ.0) GO TO 202
IF (ABS(REAL(ELEMENT)-REAL(DFILE(NSTART,1))).LT.ERROR) GO TO 285
IF (ABS(REAL(DFILE(NSTART+1,1))-REAL(ELEMENT))).LT.ERROR) GO TO 280
202 CALL FUSI(NSTART+1, LAST, DFILE, MSIZE)
DFILE(NSTART+1,1) = ELEMENT
DFILE(NSTART+1,2) = COMPLEX(VINDEX, V2INDEX)
GO TO 295
280 IF (AIMAG(ELEMENT).GE.AIMAG(DFILE(NSTART+1,1))) GO TO 295
DFILE(NSTART+1,1) = ELEMENT
DFILE(NSTART+1,2) = COMPLEX(VINDEX, V2INDEX)
GO TO 295
285 IF (AIMAG(ELEMENT).GE.AIMAG(DFILE(NSTART,1))) GO TO 295
DFILE(NSTART,1) = ELEMENT
DFILE(NSTART,2) = COMPLEX(VINDEX, V2INDEX)
GO TO 295
290 IF (AIMAG(ELEMENT).GE.AIMAG(DFILE(LAST,1))) GO TO 295
DFILE(LAST,1) = ELEMENT
DFILE(LAST,2) = COMPLEX(VINDEX, V2INDEX)
295 RETURN
END
SUBROUTINE: PUSH

This subroutine will push all the data in 'DFILE' by 1 position.
The operation is specified by NSTART1 - the beginning of push
LAST1 - the last location of DFILE

SUBROUTINE PUSH (NSTART1, LAST1, DFILE, MSIZE)
COMPLEX DFILE (MSIZE, 2), TEMP1, TEMP2
N = LAST1 - NSTART1 + 1
DO 5 I = 1, N
   TEMP1 = DFILE (LAST1 + I - 1, 1)
   TEMP2 = DFILE (LAST1 + I - 1, 2)
   DFILE (LAST1 + I - 1, 1) = TEMP1
   DFILE (LAST1 + I - 1, 2) = TEMP2
5 CONTINUE
LAST1 = LAST1 + 1
RETURN
END
SUBROUTINE: NORMAL

THIS SUBROUTINE WILL NORMALIZE A PHASE RANGE OF (-PI TO PI)
TO A RANGE OF (0 TO 2*PI).

ALPHA: ANGLE IN RADIANS TO BE NORMALIZED

SUBROUTINE NORMAL (ALPHA)
PI = 3.14159265
IF (ALPHA.LT.0) ALPHA = ALPHA + 2.*PI
RETURN
END
PROGRAM: INTERPOL

THIS PROGRAM WILL INTERPOLATE FOR 3 OPTIONS.

FILE NAME STORED THE GRID POINT INFORMATION GENERATED PREVIOUSLY.

THIS PROGRAM WILL REPEAT ITSELF, EXIT ONLY BY CONTROL C

THIS PROGRAM LINKS WITH 'PLOLIB'

THIS PROGRAM REQUIRES AN INPUT OF A MEASUREMENT FILE :FNAME

THE STRUCTURE OF FNAME:

FMIN, FMAX -MAXIMUM AND MINIMUM FREQUENCY USED IN HZ(2E15.8)
FDELTA -NORMALIZATION FACTOR IN FREQUENCY(1E15.8)
NPOINT -NUMBER OF POINTS IN THIS FILE(18)
ISO.RADIUS -ISO='T' IF ISOTROPIC SAMPLING WAS DONE(A2)
-ISO='F' IF ISOTROPIC SAMPLING WAS NOT DONE
-RADIUS IS THE ISOTRPIIC CIRCLE RADIUS USED IN SECOND(1E15.8)
V1.V2 -VECTOR1 AND VECTOR2 IN FREQUENCY DOMAIN(4E15.8)
UL.U2 -VECTOR1 AND VECTOR2 IN TIME/SPACE DOMAIN(4E15.8)
*,*,*,*--REAL AND IMAG MEASUREMENT;N1.N2 FOR THE MEASUREMENT(4E15.8)
(MAKE SURE ALL THE ABOVE ARGUMENTS ARE SEPARATED BY COMMAS IN FNAME)

N1*V1.N2*V2 SPECIFIES THE SAMPLING LATTICE

LINK INTERPOL, FILE, SUM, CONSt1, CONSt2, FUNC1, FUNC2, FUNC3, SINC,-

-PILOT, ELIM, 'PLOLIB'

COMPLEX SUM, DATA(360), V1.V2, U1.U2, FREQ, COEFF (10000.2), TEMPL

CHARACTER ISO
CHARACTER*10 FNAME
MSIZE=10000

NDATA :SPECIFIES THE NUMBER OF ANGULAR POINTS
NUM :SPECIFIES THE NUMBER OF FREQUENCY POINTS

NDATA=360
NUM=201
PI=3.14159265
WRITE(6, *) ' FILE WHICH HAS GRID POINT MEASUREMENT?'
ACCEPT 50, FNAME

50 FORMAT(A10)
OPEN(UNIT=9, FILE=FNAME, TYPE='OLD')

READ IN NECESSARY DATA

READ(9, 550) FMIN, FMAX

550 FORMAT(2E15.8)
READ(9, 551) FDELTA

551 FORMAT(E15.8)
READ(9, 552) NPOINT

552 FORMAT(18)
IF (NPOINT.GT.MSIZE) GO TO 8888

153
READ(9,553) ISO, RADIUS
READ(9,554) VI,V2
READ(9,555) U1,U2

V1=2.*PI*DELTA*VI
V2=2.*PI*DELTA*V2
U1=UL/FDELTA
U2=UL/FDELTA

C OPTIONS TO GUARD AGAINST MEASUREMENT DATA IN FORMAT
C OTHER THAN REAL AND IMAGINARY

WRITE(6,*) 'DATA IN WHICH FORMAT: 1 REAL AND IMG'
WRITE(6,*) ' 2) AMP (LINEAR) AND PHASE (RADIAN) '
WRITE(6,*) ' 3) AMP (DB) AND PHASE (RADIAN) '
WRITE(6,*) ' 4) AMP (DB) AND PHASE (DEGREE) '
ACCEPT *,NCDM1
IF ((NCDM1.GT.4).OR.(CM1M.LT.1)) GO TO 570
DO 560 I=1,NCDM1
READ(9,554)XREAL,XIMAG,COEFF(I,1,2)
IF(NCDM1.EQ.1) GO TO 580
AMP=XREAL
PHASE=XIMAG
IF((NCDM1.EQ.3).OR.(NCDM1.EQ.4)) AMP=10**(AMP/20)
IF(NCDM1.EQ.4) PHASE=PHASE*PI/180.
XREAL=XAMP*COS(PHASE)
XIMAG=XAMP*SIN(PHASE)
COEFF(I,1)=CMPLX(XREAL,XIMAG)
580 CONTINUE
CLOSE(UNIT=9,DISP='SAVE')

ASK FOR THE TYPE OF INTERPOLATION

WRITE(6,*) ' 3 CHOICES: 1) INTERPOLATE FOR 1 ASPECT ANGLE AND OUTPUT'
WRITE(6,*) ' 2) INTERPOLATE FOR 1 FREQUENCY AND OUTPUT '
WRITE(6,*) ' 3) INTERPOLATE INTO 2-D SQUARE GRID'
ACCEPT *, NWDM

THE BRANCHING OF DECISION

IF (NCDM1.EQ.2) GO TO 200
IF (NCDM1.EQ.3) GO TO 300
IF (NCDM1.EQ.1) GO TO 5

TO INTERPOLATE FOR 1 ASPECT ANGLE BUT 201 FREQUENCY POINTS

WRITE(6,*) 'FREQUENCY RANGE? LOW TO HIGH IN HZ'
ACCEPT *, FMINS, FMAXS
IF (FMINS.GE.MAX) GO TO 10
WRITE(6,*,'(A5)') ' WHICH ASPECT ANGLE? (degree)'
ACCEPT *, ANGLE
ANGLE=ANGLE*PI/180.
FINCR=(MAXS-FMINS)/(NUM-1)
DO 15 I=1,NUM
   WRITE(6,*,'(A8)') 'WORKING ON: ',I,' LAST ONE: ',NUM
   AFRED=FMINS + FINCR*(I-1)
   IF (AFRED.EQ.0.) GO TO 13
   WRITE(6,*,'(A8)') 'WORKING ON: ',I,' LAST ONE: ',NUM
   AFCRN=AFRED
   IF (AFCRN.EQ.0.) GO TO 13
   CALL MCRUX (AMCDX (ANGLE) ,AMCRUX*SN (ANGLE))
   DATA(I) = SUM (FRED, CDEFF, MSIZE, V1, V2, UL, US, ISO, NPOINT, RADIUS)
   GO TO 15
15 CONTINUE
13 CONTINUE
PLOT AND WRITE FILE
CALL VLLOB (DATA, NUM, FMINS, FINCR)
CALL WFILE (DATA, NDATA, NUM)
GO TO 5
SECOND SECTION OF THE PROGRAM
TO INTERPOLATE FOR 1 FREQUENCY, BUT 360 DEGREE POINTS
WRITE(6,*,'(A5)') ' WHICH FREQUENCY? (Hz)'
ACCEPT *, AFRED
DO 205 I=1,NDATA
   WRITE(6,*,'(A8)') 'WORKING ON: ',I,' LAST ONE: ',NDATA
   ANGLE=(I-1)*PI/180.
   IF (AFRED.EQ.0.) GO TO 203
   WRITE(6,*,'(A8)') 'WORKING ON: ',I,' LAST ONE: ',NDATA
   AFCRN=AFRED
   IF (AFCRN.EQ.0.) GO TO 203
   CALL MCRUX (AMCDX (ANGLE) ,AMCRUX*SN (ANGLE))
   DATA(I) = SUM (FRED, CDEFF, MSIZE, V1, V2, UL, US, ISO, NPOINT, RADIUS)
   GO TO 205
205 CONTINUE
PLOT AND WRITE THE FILE
CALL POLARP (DATA, 35.3, 1, NDATA, 1.1)
CALL WFILE (DATA, NDATA, NDATA)
GO TO 5
C
C  THIRD PART OF THE PROGRAM
C  TO INTERPOLATE INTO A 2-D SQUARE GRID
C  (DUE TO TIME CONSUMPTION, THIS IS NOT IMPLEMENTED. BUT IT CAN
C  BE DONE SIMILARLY AS PART 1 AND 2)
C
300 WRITE(6,*) 'NOT READY YET!' GO TO 5
8888 WRITE(6,*) 'ERROR: THE SIZE OF INPUT FILE IS LARGER THAN SPECIFIED!' GO TO 5
END
COMPLEX FUNCTION: SUM

THIS FUNCTION WILL DO THE SUMMATION OF \( F(N_1*V_1+N_2*V_2)*G(W-N_1*V_1-N_2*V_2) \)
(EQUATION 2-9 IN THIS WRITE UP)
FOR ALL AVAILABLE VALUES OF N1 AND N2 IN THE FILE NAME

FREQ : COMPLEX (F1,F2); LOCATION IN THE FREQUENCY PLANE
COEFF : COMPLEX ARRAY STORING THE SAMPLED VALUES
MSIZE : SIZE OF COEFF
V1.V2 : VECTOR 1 AND 2 IN THE FREQUENCY DOMAIN
U1.U2 : VECTOR 1 AND 2 IN THE SPACE DOMAIN
ISO : I = ISOTROPIC SAMPLING, F = NON-ISOTROPIC SAMPLING (A1)
NPOINT : NUMBER OF NON-ZERO ELEMENTS IN COEFF
RADIUS : RADIUS OF THE ISOTROPIC CELL

THIS REQUIRES THE SUPPORT OF CONST1 .CONST2

COMPLEX FUNCTION SUM(FREQ,COEFF,MSIZE,V1.V2,U1.U2,ISO,NPOINT,RADIUS)
CHARACTER ISO
COMPLEX FREQ,V1.V2,U1.U2,W,COEFF (MSIZE,2)
PI=3.14159265
SUM=CMPLX(0.,0.)

CONVERSION OF FREQUENCY INTO RADIAN FREQUENCY

FREQ=2.*PI*FREQ
DO 5 I=1,NPOINT
W=FREQ-REAL(COEFF(I,2))V1-IMAG(COEFF(I,2))V2
IF (ISO.EQ. 'T') SUM=SUM + COEFF(I,1)*CONST1 (RADIUS,W)
IF (ISO.EQ. 'F') SUM=SUM + COEFF(I,1)*CONST2(U1,U2,W)
5 CONTINUE
RETURN
END
FUNCTION :CONST1

THIS FUNCTION IS FOR RECONSTRUCTION OF ISOTROPIC SAMPLING
CALCULATING:
(1/((R**2) *W1* (W1**2-3*W2**2))) *(( 2*W1*COS (W1/SORT(3)))*COS (R*W2)
-2*W1*COS(2*R*W1/SORT(3))
-2*SORT (3)*W2*SIN (R*W1/SORT(3)))*SIN (R*W2))

THIS REQUIRE THE SUPPORT OF FUNC1, FUNC2, FUNC3, ELIM

R:CONSTANT
W:COMPLEX (W1, W2)

FUNCTION (CONST 1(R, W)
COMPLEX W
DIMENSION AB(3,4)
TDLER=1.
W1=REAL (W)
W2=AIMAG (W)
C=SORT(3.)
D1=(W1-C*W2)*R
D2=(W1+C*W2)*R

THE RECONSTRUCTION FUNCTION WILL BLOW UP WHEN 1)W1=0 OR
2)W1**2-3*W2**2

IF (ABS(D1).GE.TDLER).AND. (ABS(D2).GE.TDLER)) GO TO 999
IF (D1.EQ.0.).OR. (D2.EQ.0.) GO TO 888

THERE IS ONE MORE PROBLEM DURING IMPLEMENTATION:
THE COMPUTER'S UNDERFLOW PROBLEM.
 THEREFORE FOR TDLER=1, THE FUNCTION IS INTERPOLATED
 WITH THREE POINT MATCHING POLYNOMIAL.
 THE COEFFICIENTS OF THE POLYNOMIAL IS CALCULATED BY GAUSSIAN
 ELIMINATION. THIS PROGRAM IS FURNISHED BY MR. BING KWAN
 ELIM(AB,3,4,3)

IF (ABS(D2).LT.TDLER) GO TO 555
AB(1,2)=((W1*R)+(TDLER))/C
AB(2,2)=(-W1*R/C)
AB(3,2)=((W1*R)-(TDLER))/C
GO TO 777

555 AB(1,2)=TDLER+(W1*R))/C
AB(2,2)=W1*R/C
AB(3,2)=TDLER-W1*R/C
GO TO 777

777 AB(1,4)=FUNC1(R,W1, (AB(1,2))/R)
AB(2,4)=FUNC2(R, (AB(2,2))/R)
AB(3,4)=FUNC3(R,W1, (AB(3,2))/R)
DO 800 J=1,3
AB(J,3)=1.0
$AB(3,1) = (AB(3,2))^2.$

800 CONTINUE
CALL ELIM(AB, 3, 4, 3)
CONST1 = (FUNC3(AB(1, 4), AB(2, 4), AB(3, 4), M2*W))
GO TO 1000
888 CONST1 = FUNC2(R, W2)
GO TO 1000
999 CONST1 = FUNC1(R, W1, W2)
1000 RETURN
END
FUNCTION:  
FUNCl

THIS IS THE RECONSTRUCTION FUNCTION WHEN THE DENOMINATOR IS NOT ZERO.

FUNCTION FUNCl(R, W1, W2)
C=SORT(3.)
D1=(W1-C+W2)*R
D2=(W1+C-W2)*R
Xl=(COS(R*W1/C)) *(COS(R*W2))
X2=COS (2.*R/W1/C)
X3=(W2*R)* (SINC(R*W1/C)) *(SINC(R*W2))
FUNCl=2. *(X1+X2+X3)/(D1*D2)
RETURN
END
FUNCTION:   FUNC2

THIS IS THE RECONSTRUCTION FUNCTION WHEN \( w1^2 - 3w2^2 = 0 \)

FUNCTION FUNC2(R, W2)
C1=(2./3.)*SINC(2.*R*W2)
C2=(1./3.)*(SINC(R*W2))^2.
FUNC=C1+C2
RETURN
END
FUNCTION: FUNC3

This function will produce the linear interpolation portion of the reconstruction function when the underflow problem occurs.

FUNCTION FUNC3(A, B, C, W2)
X1 = A*(W2**2)
X2 = B*W2
FUNC3 = X1 + X2 + C
RETURN
END
FUNCTION:  CONST2

THIS FUNCTION IS TO RECONSTRUCT RECTANGULAR SAMPLING

CALCULATING:
\[(\sin(0.5*(V1.*W))) \cdot (\sin(0.5*(V2.*W))) / (0.25*(V1.*W) \cdot (V2.*W))\]

THIS FUNCTION REQUIRES THE SUPPORT OF SINC

FUNCTION CONST2(V1,V2,W)
COMPLEX V1,V2,W
X1=REAL(V1)*REAL(W) + IMAG(V1) * IMAG(W)
X2=REAL(V2)*REAL(W) + IMAG(V2) * IMAG(W)
CONST2=SINC(X1/2.) * SINC(X2/2.)
RETURN
END
SUBROUTINE: RPLT

This subroutine will do rectangular plot
for real and imaginary plot or magnitude and phase plot.

DATA : Complex array to be plotted (size=360)
LAST : Last element in the array
FMIN : Smallest value on the abscissa
FINCR : Increment on the abscissa

SUBROUTINE RPLT (DATA, LAST, FMIN, FINCR)
CHARACTER COM
COMPLEX DATA(360)
DIMENSION YAXIS1(201), YAXIS2(201), XAXIS(201)

WRITE (6, *) 'DO YOU WANT REAL AND IMAGINARY PLOT? Y/N'
ACCEPT 6, COM

FORMAT (A1)
IF (COM.EQ. 'N') GO TO 15
IF (COM.NE. 'Y') GO TO 5

DO 10 I = 1, LAST
  YAXIS1(I) = REAL (DATA(I))
  YAXIS2(I) = IMAG (DATA(I))
  XAXIS(I) = (I-1)*FINCR + FMIN

10 CONTINUE
GO TO 25

DO 20 I = 1, LAST
  YAXIS1(I) = (I-1)*FINCR + FMIN
  YAXIS1(I) = ABS (DATA(I))
  IF (REAL (DATA(I)).EQ. 0.) GO TO 18
  YAXIS2(I) = ATAN (IMAG (DATA(I)), REAL (DATA(I)))

20 CONTINUE
GO TO 20

YAXIS2(I) = 0.

CALL PITAG (XAXIS, YAXIS1, 201, 201, 1.0, 1.)
CALL PITAG (XAXIS, YAXIS2, 201, 201, 1.0, 1.)
RETURN
END
SUBROUTINE :ELIM

THIS SUBROUTINE IS COURTESY OF MR. BING EWAN[16]

SUBROUTINE ELIM(A,B,NP,NDIM)
DIMENSION A(NP,NDIM)

THIS SUBROUTINE SOLVES A SET OF LINEAR EQUATIONS.
THE GAUSS ELIMINATION METHOD IS USED, WITH PARTIAL PIVOTING.
MULTIPLE RIGHT HAND SIDES ARE PERMITTED, THEY SHOULD BE SUPPLIED
AS COLUMNS THAT AUGMENT THE COEFFICIENT MATRIX.
PARAMETERS ARE:
AB COEFFICIENT MATRIX AUGMENTED WITH R.H.S. VECTORS.
N NO. OF EQUATIONS
NP TOTAL NO. OF COLUMNS IN AB
NDIM NO. OF ROWS IN AB

BEGIN THE REDUCTION
M=1-N-1
DO 35 I=1,M

FIND THE ROW NUMBER OF THE PIVOT ROW. WE WILL THEN
INTERCHANGE ROWS TO THE PIVOT ELEMENT ON THE DIAGONAL.

IPV'T=I
IP=I+1
DO 10 J=IP1,N
IF (ABS (AB (IPVT, I) ).LT.ABS (AB (J, I))) IPVT=J

10 CONTINUE

CHECK TO BE SURE THE PIVOT ELEMENT IS NOT TOO SMALL, IF SO
PRINT A MESSAGE AND RETURN.
IF(ABS(AB(IPVT,I)).LT.1.E-5) GO TO 99

NOW INTERCHANGE , EXCEPT IF THE PIVOT ELEMENT IS ALREADY ON
THE DIAGONAL, DO NOT NEED TO.
IF(IPV'T.EQ.1) GO TO 25
DO 20 J=1,NP
SAVE=AB(I,J)
AB(I,J)=AB(IPVT,J)
AB(IPVT,J)=SAVE

20 CONTINUE

NOW REDUCE ALL ELEMENTS BELOW THE DIAGONAL IN THE I-TH ROW.
CHECK FIRST TO SEE IF ZERO ALREADY PRESENT. IF SO
CAN SKIP REDUCTION FOR THAT ROW.

DO 32 J=1,NP
IF(AB(J,1).EQ.0.0) GO TO 32
RATIO=AB(J,1)/AB(1,1)
DO 30 K=1,NP
AB(J,K)=AB(J,K)-RATIO*AB(I,K)

30 CONTINUE

32 CONTINUE

35 CONTINUE

WE STILL NEED TO CHECK A(N,N) FOR SIZE.
IF(ABS(AB(N,N)).LT.1.E-5) GO TO 99
C NOW WE BACK SUBSTITUTE.
   NP1=NP1+1
   DO 50 KCOL=KCOL,NP1
       AB(N,KCOL)=AB(N,KCOL)/AB(N,N)
   DD 45 J=2,N
       NVBL=NP1-J
       L=NVEL+1
       VALUE=AB(NVBL,KCOL)
       DO 40 K=L,N
           VALUE=VALUE-AB(NVBL,K)*AB(K,KCOL)
           CONTINUE
       AB(NVBL,KCOL)=VALUE/AB(NVBL,NVEL)
   45 CONTINUE
   50 CONTINUE
RETURN
C MESSAGE FOR A NEAR SINGULAR MATRIX.
99 WRITE(66,100)
100 FORMAT(/,'SOLUTION NOT POSSIBLE. A NEAR 0 PIVOT ENCOUNTERED. ')
     RETURN
END
SUBROUTINE: WFILE

This subroutine will write a complex 'DATA' array of size 'MSIZE' into a file 'FILENAME'. The number of non-zero elements is specified by 'NUM'.

SUBROUTINE WFILE (DATA, MSIZE, NUM)
COMPLEX DATA(MSIZE)
CHARACTER*10 FNAME1
CHARACTER WRI
10 WRITE (6,*) 'DO YOU WANT TO WRITE INTO A FILE? Y/N'
   ACCEPT 9010, WRI
   IF (WRI.EQ.'N') GO TO 9999
   IF (WRI.NE.'Y') GO TO 10
   WRITE(6,*) 'STORAGE FILE NAME:'
   ACCEPT 9020, FNAME1
   OPEN (UNIT=8, NAME=FNAME1, TYPE='NEW')
   DO 30 I=1,NUM
      WRITE (8,*) DATA(I)
30 CONTINUE
   CLOSE (UNIT=8, DISP='SAVE')
9010 FORMAT (A1)
9020 FORMAT(A10)
9999 RETURN
END
PROGRAM : INTEGFFT

THIS PROGRAM WILL READ FROM A COMPLEX DATA FILE: FNAME1(MSIZE)
WHICH IS PART OF AN INTEGRAND FOR INTEGRATION.
The data argument has a range of (0, 360).
(i.e., MSIIE is usually a multiple of 360 < 680)
For 8192 < MSIIE > 512 change MNP1 and the array size of: DATA1, DATA2, S
The other part of the integrand is specified in
FNC1. The integration is like a convolution.
Therefore, the integration is using FT approach
A rectangular to polar coordinate file must be supplied
This is for the time plot which is FNAME3.
This file is generated by program: RECTPOLAR
FNAME3 format:

- NUMBER OF POINT: (*)
- RADIUS, ANGLE (Radian), X-COORDINATE, Y-COORDINATE (4E15-8)

The only part that is related to the MENSA'S actual integration is
shown at a later set of comment before TEMP1 is calculated.

Link INTEGFFT, APTOR1, EXPAND, PICK, 'SSP

Complex DATA1 (512), DATA2 (512), FACTOR, POLAR (10000), RECT (10000), Dummy
Dimension S(128)
Character*10 FNAME1, FNAME2, FNAME3
Character*9, OD
M=9
NPI=512
NPL=129
NPOLAR=10000
PI=3.14159265

X2DR is used to normalize the amp of the output

X2DR=18-9

This time factor is each unit of the time plot

TIME=1/(0.492125989*31.)
WRITE(6,*) 'FILE NAME OF MEASURED DATA:'
ACCEPT 9000, FNAME1
WRITE(6,*) 'FREQUENCY USED: (Hz)'
ACCEPT*, FREQ
WRITE(6,*) 'NUMBER OF POINTS IN THE FILE:'
ACCEPT*, MSIZE

If the number of points in the file is less than NPL,
then the data may be one quadrant data only.
IF (MSIZE.GT.NP1) GO TO 998
IF (MSIZE.GT.NP1) GO TO 110
120 WRITE(6,*) 'IS THIS ONE QUADRANT DATA? Y/N'
ACCEPT 9020,ONE
IF (ONE.EQ. 'Y') GO TO 110
IF (ONE.NE. 'Y') GO TO 120
130 WRITE(6,*) 'COMPLEX CONJUGATE IN 2ND AND 3RD QUADRANTS? Y/N'
ACCEPT 9020,ODD
IF ((ODD.NE. 'Y').OR.(ODD.NE. 'N')) GO TO 130
C TO GUARD AGAINST POSSIBILITY OF MEASUREMENT IN OTHER FORMAT
C
110 WRITE(6,*) 'FORMAT OF DATA: 
  1) REAL AND IMAGINARY' 
WRITE(6,*) ' 2) AMP(LINEAR) AND PHASE(RADIAN)'
WRITE(6,*) ' 3) AMP(LINEAR) AND PHASE(DEGREE)'
WRITE(6,*) ' 4) AMP(DB) AND PHASE(RADIAN)'
WRITE(6,*) ' 5) AMP(DB) AND PHASE(DEGREE)'
ACCEPT *,NFORM
IF ((NFORM.GT.5).OR.(NFORM.LT.1)) GO TO 110
WRITE(6,*) 'STORAGE FILE NAME:'
ACCEPT 9000,FRAME2
WRITE(6,*) 'NAME OF THE RECTANGULAR TO POLAR FILE:'
ACCEPT 9000,FRAME3
C
C INITIALIZATION
C
DO 10 I=1,NP1
DATA(I)=(0.,0.)
10 CONTINUE
C READ IN MEASUREMENT DATA
C OPEN(UNIT=8,NAME=FRAME1,TYP='OLD')
DO 60 I=1,MSIZE
READ(8,9030) DUMMY
CALL APTORI (DUMMY,NFORM)
DATA(I)=DUMMY
60 CONTINUE
CLOSE(UNIT=8,DISP='SAVE')
C EXPAND TO FILL THE HALF OF THE SPAN OF THE FFT REPETITIVE UNIT
C IF (MSIZE.LE.NP1) CALL EXPAND (DATA1,DATA2,NP1,MSIZE,NP1)
IF (MSIZE.GT.NP1) CALL EXPAND (DATA1,DATA2,NP1,MSIZE,NP1)
IF (MSIZE.GT.NP1) GO TO 170
C CASE OF FULL PLANE DATA INPUT
C IF (ONE.EQ. 'N') CALL EXPAND (DATA1,DATA2,NP1,NP1,NP1)
IF (ONE.EQ. 'N') GO TO 170
C FILL IN THE OTHER HALF OF THE SPAN BY COMPLEX CONJUGATE OR
C THE SAME
C
C MSIZE=MSIZE-2
DO 160 I=1,MSIZE-2
  IF (ODD.EQ.'N') DATA1(I+MSIZE)=DATA1(MSIZE-I)
  IF (ODD.EQ.'Y') DATA1(I+MSIZE)=CONJG(DATA1(MSIZE-I))
160 CONTINUE
C MSIZE=2*MSIZE-2
DO 165 I=1,MSIZE
  IF (ODD.EQ.'N') DATA1(I+MSIZE)=DATA1(I)
  IF (ODD.EQ.'Y') DATA1(I+MSIZE)=CONJG(DATA1(I))
165 CONTINUE
C MSIZE=MSIZE*2
IF (MSIZE.GT.NP1) GO TO 998
C MSIZE=NP1
C DELTA: SIZE OF EACH ANGLE INCREMENT
C
C DELTA=-2.*PI./MSIZE
OPEN (UNIT=8,NAME=FNAME3,TYPE='OLD')
READ(8,*) NPOINT
IF (NPOINT.NR.NP1) GO TO 998
C READ THE RECTANGULAR TO POLAR COORDINATE CONVERSION FILE
C
C DO 100 I=1,NPOINT
  READ(8,9005) POLAR(I),RECT(I)
100 CONTINUE
C THE VALUE OF AMP1 = 1 IS ONLY A DUMMY TO START THE ROUTINE
C AMP1=1.
DO 200 J=1,NPOINT
  WRITE(6,*) 'WORKING ON: ',J,' LAST ONE: ',NPOINT
  IF (REAL(POLAR(J)).LT.AMP1) GO TO 50
  AMP1=POLAR(J)
  DO 17 I=1,NP1
    DATA2(I)=(0.,0.)
17 CONTINUE
DO 20 I=1,MSIZE
  THEIA=I-1)*DELTA
  PHI=2.*PI*(REAL(POLAR(J)))*TIME*FREQ*COS(THEIA)
C
C FUNC1: EXP(JMT)
C THEREFORE, XREAL=COS(PHI)
C YIMG=SIN(PHI)
C
XREAL=COS(PHI)
DATA2(I) = CMPLX(REAL(X), IMAG(X))

CONTINUE

GO TO FREQUENCY DOMAIN

CALL FORT(DATA2,M,S,2,IPERR)

CONVOLUTION IN TIME <-> MULTIPLICATION IN FREQUENCY

DO 25 I=1,NP1
   DATA2(I) = DATA1(I) * DATA2(I)
25 CONTINUE

GO BACK TO TIME DOMAIN

CALL FORT(DATA2,M,S,-2,IPERR)

NOW POLAR(J) IS THE INTEGRAL VALUE OF 2D FOURIER TRANSFORM
WITH IMPULSIVE RADIUS VALUE (Mensa's integral)

ANGLE = IMAG(POLAR(J))
POLAR(J) = CMPLX(DATA2,NP1,0.,DELTA,ANGLE)
POLAR(J) = (POLAR(J)/FLOAT(MSIZ))/FREQ*QDR

CONTINUE

WRITE OUT THE FILE

FILE FORMAT:
NPOINT: NUMBER OF POINTS (110)
REAL, IMAGINARY, X-COORDINATE, Y-COORDINATE (4E15.8)

OPEN (UNIT=9, NAME=NAME2, TYPE='NEW')
WRITE (9,9010) NPOINT
DO 300 I=1,NPOINT
   WRITE (9,9005) POLAR(I),RECT(I)
300 CONTINUE
CLOSE (UNIT=9, DISP='SAVE')
GO TO 999

WRITE (6,*), 'ERROR: SIZE OF FILE TOO LARGE'

FORMAT STATEMENTS

9000 FORMAT (A10)
9005 FORMAT (4E15.8)
9010 FORMAT (I10)
9020 FORMAT (A1)
9030 FORMAT (2E15.8)

END

STOP
SUBROUTINE: EXPAND

THIS SUBROUTINE WILL EXPAND AN ARRAY DATA(MSIZE) CONTAINING 'NFILL'
data elements into 'NEXP' data element using linear interpolation
2<NFILL<NEXP

WORK: A DUMMY ARRAY FOR TEMPORARY STORAGE

SUBROUTINE EXPAND(DATA, WORK, MSIZE, NFILL, NEXP)
COMPLEX DATA(MSIZE), WORK(MSIZE), DIFF
ERROR=10^-6
DELTA1=1./FLOAT(NFILL-1)
DELTA2=1./FLOAT(NEXP-1)
DO 10 I=1,NFILL
   WORK(I)=DATA(I)
10 CONTINUE
   DATA(1)=WORK(1)
   DATA(NEXP)=WORK(NFILL)
   DO 20 I=2,NEXP-2
      X=(I)*DELTA2
      N=INT(X/DELTA1)+1
      REMAIN=ABS(X,DELTA1)
      RATIO=(X-DELTA1*FLOAT(N-1))/DELTA1
      DIFF=(WORK(N)-WORK(N-1))/RATIO
      DATA(I+1)=WORK(N)*RATIO
   20 CONTINUE
RETURN
END
COMPLEX FUNCTION: PICK

THIS FUNCTION WILL RETURN A LINEAR INTERPOLATED COMPLEX VALUE BACK.

DATA: ARRAY
MSIZE: SIZE OF THE ARRAY
FIRST: FIRST/Delta IS THE FIRST INDEX
Delta: INCREMENT OF EACH STEP IN THE ARRAY
VALUE: VALUE/Delta IS THE LOCATION IN THE ARRAY
N: EXACT LOCATION OR LOCATION -1
VARY: AMOUNT OF ANGLE DIFFERENCE
FACTOR: ADJUSTMENT TO DATA2(N)

COMPLEX FUNCTION PICK(DATA, MSIZE, FIRST, DELTA, VALUE)
COMPLEX DATA(MSIZE), FACTOR
IF ((VALUE.LT.FIRST) .OR. (VALUE.GT. (MSIZE*DELTA + FIRST))) GO TO 998
N=INT((VALUE-FIRST)/DELTA) +1
VARY= (VALUE-(N-1)*DELTA+FIRST)
FACTOR= ((DATA(N) - DATA(N) )/DELTA) * VARY
PICK= DATA(N) + FACTOR
GO TO 999
998 WRITE(6,*), 'ERROR: INTERPOLATED POINT IS OUTSIDE THE RANGE!'
999 RETURN
END
SUBROUTINE: APIORI

THIS SUBROUTINE WILL CONVERT AMPLITUDE AND PHASE (VALUE) INTO
REAL AND IMAGINARY (VALUE).
BY SPECIFYING N
N=1:VALUE IS ALREADY IN REAL AND IMAGINARY FORM
N=2:AMPLITUDE (LINEAR), PHASE (RADIAN)
N=3:AMPLITUDE (LINEAR), PHASE (DEGREE)
N=4:AMPLITUDE (DB) , PHASE (RADIAN)
N=5:AMPLITUDE (DB), PHASE (DEGREE)

SUBROUTINE APIORI (VALUE, N)
COMPLEX VALUE
IF (N.EQ.1) GO TO 999
PI=3.14159265
AMP=REAL(VALUE)
PHASE=AIMAG(VALUE)
IF ((N.EQ.4).OR.(N.EQ.5)) AMP=10.0**((AMP/10.)
IF ((N.EQ.3).OR.(N.EQ.5)) PHASE=PHASE*PI/180.
VALUE=CMPLX (AMP*COS(PHASE), AMP*SIN(PHASE))
999 RETURN
END
PROGRAM : RECTILAR

THIS PROGRAM WILL CHANGE RECTANGULAR COORDINATES TO POLAR COORDINATES. THE FILE 'FNAME' OUTPUTS IN THE FOLLOWING FORMAT:
RADIUS, PHI (RADIANS), X,Y (4E15.8)
The file will also be organized from smallest radius to the largest radius.

NOTE THE MAXIMUM X COORDINATE VALUE: 'MAX' WILL GIVE
(MAX**2)**2 +2*2*MAX +1 DATA POINTS
Therefore a grid of 256X256 can accommodate MAX=127

LINK RECTILAR, GEN, SEARCH, INSERT, PUSH, NORMAL

COMPLEX DATA(65536,2)
CHARACTER*10 FNAME
WRITE(6,*) 'FILE NAME FOR STORAGE OF RESULTS:
ACCEPT 5,FNAME
5 FORMAT(A10)
WRITE(6,*) 'MAXIMUM X COORDINATE VALUES:
ACCEPT *,MAX
MSIZE=65536
FX=3.14159265
M=0
M=M+1
DATA(M,1)=CMPLX(0.,0.)
DATA(M,2)=CMPLX(0.,0.)

THIS DO LOOP GENERATES ALL THE AXIS POINTS

DO 10 J=1,MAX
X1=FLOAT(J)
M=M+1
DATA(M,1)=CMPLX(X1,0.)
DATA(M,2)=CMPLX(X1,0.)
M=M+1
DATA(M,1)=CMPLX(X1,PI/2)
DATA(M,2)=CMPLX(0.,X1)
M=M+1
DATA(M,1)=CMPLX(X1,PI)
DATA(M,2)=CMPLX(-X1,0.)
M=M+1
DATA(M,1)=CMPLX(X1,3.*PI/2.)
DATA(M,2)=CMPLX(0.,-X1)
10 CONTINUE
WRITE(6,*) 'CALCULATING FIRST QUADRANT POINTS!'
CALL GEN(MAX,DATA,MSIZE,M,1,1)
WRITE(6,*) 'CALCULATING SECOND QUADRANT POINTS!'
CALL GEN(MAX,DATA,MSIZE,M,-1,1)
WRITE(6,*) 'CALCULATING THIRD QUADRANT POINTS!'
CALL GEN(MAX, DATA, MSIZE, M, -1, -1)
WRITE (6, *) 'CALCULATING FOURTH QUADRANT POINTS'
CALL GEN(MAX, DATA, MSIZE, M, -1, -1)
OPEN (UNIT=8, NAME='<NAME>', TYPE='NEW')
WRITE (8, *) M
DO 20 I=1, M
WRITE (8,1000) DATA(I,1), DATA(I,2)
20 CONTINUE
1000 FORMAT (4E15.8)
STOP
END
SUBROUTINE: GEN

THIS SUBROUTINE WILL GENERATE OFF AXIS POINTS
FOR V1=(1,0), V2=(0,1)
THIS IS A MODIFIED VERSION OF GEN1 IN PIGRID

MAX : MAXIMUM INDEX
DFILE : COMPLEX ARRAY FOR STORAGE
MSIZE : DIMENSION OF DFILE
M : NUMBER OF ELEMENT
NSIGN : SIGN OF N1 (N1*V1)
N2SIGN : SIGN OF N2 (N2*V2)

THIS REQUIRES THE SUPPORT OF SEARCH1, INSERT1, NORMAL

SUBROUTINE GEN(MAX, DFILE, MSIZE, M, NSIGN, N2SIGN)
EXTERNAL SEARCH1
COMPLEX DFILE(MSIZE, 2), VECTOR, V1, V2
REAL MAG
V1=COMPLEX(1.,0.)
V2=COMPLEX(0.,1.)
DO 5 I=1, MAX
     WRITE(*,6) I
     DO 10 J=1, MAX
         VECTOR=J*V1*NSIGN +I*V2*N2SIGN
         MAG=CABS(VECTOR)
         PHASE=ATAN2(AIMAG(VECTOR), REAL(VECTOR))
         CALL NORMAL(PHASE)
         VECTOR=COMPLEX(MAG, PHASE)
         LOCS=SEARCH1(VECTOR, DFILE, MSIZE, M)
         RJ=FLOAT(J)
         RI=FLOAT(I)
         IF ( (NSIGN.LT.0) .AND. (N2SIGN.LT.0) ) THEN
             1 CALL INSERT1(LOC, M, DFILE, MSIZE, VECTOR, -RJ, -RI)
         ELSE IF ( (NSIGN.GT.0) .AND. (N2SIGN.GT.0) ) THEN
             3 CALL INSERT1(LOC, M, DFILE, MSIZE, VECTOR, RJ, RI)
         ELSE IF ( (NSIGN.LT.0) .AND. (N2SIGN.GT.0) ) THEN
             2 CALL INSERT1(LOC, M, DFILE, MSIZE, VECTOR, -RJ, RI)
         ELSE IF ( (NSIGN.GT.0) .AND. (N2SIGN.LT.0) ) THEN
             4 CALL INSERT1(LOC, M, DFILE, MSIZE, VECTOR, RJ, -RI)
         END IF
     CONTINUE
   10 CONTINUE
     RETURN
END
FUNCTION: SEARCH1

THIS FUNCTION WILL SEARCH THE LOCATION WHERE THE ELEMENT
FITS INTO A DATA ARRAY. THE ARRAY ASSUMES AT LEAST TWO
MEMBERS. THE ELEMENT MAY HAVE 3 POSSIBILITIES:-
1) THE REAL PART IS GREATER THAN THE RETURNED LOCATION
2) THE REAL PART IS THE SAME
   BUT THE IMAGINARY PART > RETURNED LOCATION VALUES
3) THE REAL AND IMAGINARY PART ARE THE SAME
   AS THE RETURNED LOCATION VALUES

ELEMENT : COMPLEX ELEMENT TO BE INSERTED
DFILE : COMPLEX ARRAY TO BE SEARCHED
MSIZE : SIZE OF DFILE
LAST : NUMBER OF NON-ZERO ELEMENT IN DFILE

FUNCTION SEARCH1 (ELEMENT, DFILE, MSIZE, LAST)

COMPLEX DFILE (MSIZE, 2), ELEMENT
LOC2=LAST
LOC1=1
LOC(1) - INTEGER

IF (REAL(ELEMENT).LT.REAL(DFILE(LOC1,1))) GO TO 147
IF (REAL(ELEMENT).GT.REAL(DFILE(LOC2,1))) GO TO 140
IF ((REAL(ELEMENT).EQ.REAL(DFILE(LOC1,1))).AND.
   (AIMAG(ELEMENT).LT.AIMAG(DFILE(LOC1,1)))) GO TO 147
IF ((REAL(ELEMENT).EQ.REAL(DFILE(LOC2,1))).AND.
   (AIMAG(ELEMENT).GT.AIMAG(DFILE(LOC2,1)))) GO TO 140
   LOC = (LOC1 + LOC2)/2
IF (LOC .EQ. LOC1) GO TO 150
IF (REAL(ELEMENT).LT.REAL(DFILE(LOC,1))) GO TO 125
IF (REAL(ELEMENT).GT.REAL(DFILE(LOC,1))) GO TO 130
IF (AIMAG(ELEMENT).LT.AIMAG(DFILE(LOC,1))) GO TO 125
IF (AIMAG(ELEMENT).GT.AIMAG(DFILE(LOC,1))) GO TO 130
GO TO 145
LOC2 = LOC
GO TO 120
LOC1 = LOC
GO TO 120
SEARCH = LOC2
GO TO 155
SEARCH = LOC1
GO TO 155
SEARCH = LOC1-1
GO TO 155
SEARCH = LOC1
RETURN
END
SUBROUTINE: INSERT1

This subroutine will insert the element into the DFILE
if the element has a different real and/or imaginary part.
The real part has a sensitivity specified by the ERROR1.
The imaginary part has a sensitivity specified by ERROR2.
The priority is real part first then imaginary part.

NSTART: LOCATION OF THE ELEMENT
LAST: NUMBER OF NON-ZERO ELEMENTS IN DFILE
DFILE: ARRAY FILE TO BE INSERTED
MSIZE: SIZE OF DFILE
ELEMENT: COMPLEX ELEMENT TO BE INSERTED
VIINDEX: NUMBER OF VI USED
V2INDEX: NUMBER OF V2 USED

SUBROUTINE INSERT1 (NSTART, LAST, DFILE, MSIZE, ELEMENT, VIINDEX, V2INDEX)
COMPLEX DFILE (MSIZE, 2), ELEMENT
PI=3.14159265
ERROR1=1.E-6
ERROR2=0.01*PI/180.
IF (NSTART .NE. LAST) GO TO 200

CASE OF NSTART=LAST

IF (ABS(REAL(ELEMENT)-REAL(DFILE(LAST,1))) .GT. ERROR1) GO TO 190
IF (REAL(ELEMENT) .GT. REAL(DFILE(LAST,1))) GO TO 190
IF (ABS(AIMAG(ELEMENT)-AIMAG(DFILE(LAST,1))) .LT. ERROR2) GO TO 295

LAST = LAST +1
DFILE (LAST,1) = ELEMENT
DFILE (LAST,2) = COMPLEX (VIINDEX, V2INDEX)
GO TO 295

IF (NSTART .EQ. 0) GO TO 202

IF (ABS(REAL(ELEMENT)-REAL(DFILE(NSTART,1))) .GT. ERROR1) GO TO 202
IF (REAL(ELEMENT) .GT. REAL(DFILE(NSTART,1))) GO TO 202
IF (ABS(AIMAG(ELEMENT)-AIMAG(DFILE(NSTART,1))) .LT. ERROR2) GO TO 295

CALL PUSH (NSTART+1, LAST, DFILE, MSIZE)
DFILE (NSTART+1,1) = ELEMENT
DFILE (NSTART+1,2) = COMPLEX (VIINDEX, V2INDEX)
GO TO 295

RETURN

END
PROGRAM: SUM3D

THIS PROGRAM CAN DO THE FOLLOWING:
1) READ A DATA FILE
2) SUMMING OF DATA FILES
3) PLOT A DATA FILE IN A) 3-D PICTURE PLOT
   B) CONTOUR PLOT
4) 2-D FOURIER TRANSFORM
5) MODIFIES FREQUENCY DATA BY A) 1/JW
   B) -1/JW**2
6) COSINE TAPERING LOW PASS THE FREQUENCY DATA
7) HIGH PASS THE FREQUENCY DATA
8) WRITE OUT THE FILE

THE INPUT (OR OUTPUT) FILE FORMAT:
- NOINT: NUMBER OF POINTS IN THE FILE (110)
  DATA: X-COORDINATE, Y-COORDINATE (4E15.8)
- THE CONTOUR PLOT LINKS WITH NCAR PLOTTING PACKAGES
  THE LINKING REQUIRED IS:
  $ASSIGN DRA2: [NCAR2.NCARPILOT.PLOTLIB] NCAR_PLOT_LIB
  $ASSIGN DRA2: [NCAR2.NCARLIB] NCAR_LIB
  $ASSIGN DRA2: [NCAR2.NCARPLT] NCAR_PLOT
  $ASSIGN DRA2: [NCAR2.NCARPLT.CHROME] NCAR_PLOT_CHROME
  $ASSIGN DRA2: [NCAR2.NCARPLT.TEST] NCAR_PLOT_TEST
  $ASSIGN DRA2: [NCAR2.NCARPLT.2D] NCAR_PLOT_2D
  $ASSIGN DRA2: [NCAR2.NCARPLT.3D] NCAR_PLOT_3D
  $LINK SUM3D, DEPOL, TRANZ3D, WEIGHT, PLOT3D, APIDIR, 'SBS', 'PLOTLIB',-
  DRA2: [NCAR2.NCARPLT.PLOTLIB] NCARDOMRE/LIB, NCARDASHC/LIB,-
  NCARGRAF/LIB, NCARGRAPI/LIB, [NCAR2.NCARLIB] UTILITY/LIB

COMPLEX DATA (31,31), DUMY, APIDIR, CI
DIMENSION ARRAY(31,31), ARRAY(31,31), ARRAY(31,31)
CHARACTER*10 FNAME1, FNAME2
CHARACTER ADD, OMEGA, SINCE, LP, HP
N=5
MSIZE=2**N-1
PI=3.14159265
CI=CMPLX(0.1)
K=MSIZE/2+1

INITIALIZATION

DO 5 I=1,MSIZE
   DO 6 J=1,MSIZE
      DATA(I,J)=(0.,0.)
   CONTINUE
6 CONTINUE
5 WRITE(6,*)'DATA FILE NAME:'
ACCEPT 9000,FNAME1
WHAT IS THE DATA FORMAT:  
1) REAL AND IMAG'  
2) AMP (LINEAR) AND PHASE (RADIANS)  
3) AMP (LINEAR) AND PHASE (DEGREES)  
4) AMP (DB) AND PHASE (RADIANS)  
5) AMP (DB) AND PHASE (DEGREES)  

IF ((NOM GT 5) .OR. (NOM LT 1)) GO TO 110

READ IN THE DATA
OPEN (UNIT=10, NAME=FILE1, TYPE='OLD')
READ (10,9010) NPOINT
DO 10 I=1,NPOINT
   READ (10,9020) DUMMY, X1, Y1
   CALL APIDRI (DUMMY, NOD)
   M=INT (XI) +K
   N=INT (YI) +K
   DATA(M,N)= DATA(M,N) +DUMMY
10 CONTINUE
CLOSE (UNIT=10, DISP='SAVE')

PLOT
CALL DPLT (DATA, MSIZE, ARRAY1, ARRAY2, ARRAYP)

ADD ANOTHER FILE
20 WRITE (6,*) 'ADD ANOTHER FILE? Y/N'
   ACCEPT 9030, ADD
   IF (ADD.EQ.'Y') GO TO 8
   IF (ADD.EQ.'N') GO TO 20

WRITE IN A FILE
25 WRITE (6,*) 'WRITE YOUR DATA IN A FILE? Y/N'
   ACCEPT 9030, KEEP
   IF (KEEP.EQ.'N') GO TO 30
   IF (KEEP.EQ.'Y') GO TO 25
   WRITE (6,*) 'FILE NAME:'
   ACCEPT 9000, FILE2
   OPEN (UNIT=8, NAME=FILE2, TYPE='NEW')
   WRITE (8,9010) MSIZE**2
   DO 27 I=1,MSIZE
      WRITE (8,9020), (DATA(I,J), FLOAT(I-K), FLOAT(J-K), J=1,MSIZE)
27 CONTINUE
CLOSE (UNIT=8, DISP='SAVE')

FOURIER TRANSFORM
30 WRITE (6,*) 'GO TO FREQUENCY DOMAIN? Y/N'
   ACCEPT 9030, OMEGA
IF (OKOA. E"N') GO TO 50
IF (OKOA. NE. 'Y') GO TO 30
CALL TRANGD (DATA, MSIZE, -1)
CALL DFLQOT (DATA, MSIZE, ARRAY1, ARRAY2, ARRAYP)
GO TO 25

C DIVERSE FOURIER TRANSFORM
C
50 WRITE(6, *) ' GO TD TIME DOMAI N? Y/N'
ACCEPT 9030. TIME
IF (TIME. EQ. 'N') GO TO 60
IF (TIME. NE. 'Y') GO TO 50

C MODIFY TO PREPARE FOR STEP OR HAMP RESPONSE
C
120 WRITE (6, *) ' WANT TO MODIFY DATA? BY
WRITE(6,*)
WRITE (6, *)
ACCEPT * , MOD
IF ((MOD. GT. 3) OR (MOD. LT. 1)) GO TO 120
IF (MOD. EQ. 1) GO TO 95
IF (MOD. EQ. 3) GO TO 130

C MODIFY BY 1/JW
C
DO 125 I=1, MSIZE
DO 127 J=1, MSIZE
W=2.*PI *SQR ((FLOAT ( (I-MSIZE/2-1)**2+(J-MSIZE/2-1)**2)))
IF (W. NE. 0.) DATA(I, J) = DATA(I, J)/(GJ%)
IF (W. EQ. 0.) DATA(I, J) = (0., 0.)
127 CONTINUE
125 CONTINUE
GO TO 145

C MODIFY BY -1/W**2
C
130 DO 135 I=1, MSIZE
DO 140 J=1, MSIZE
W=2.*PI *SQR ((FLOAT ( (I-MSIZE/2-1)**2+(J-MSIZE/2-1)**2)))
IF (W. NE. 0.) DATA(I, J) = DATA(I, J)/W**2
IF (W. EQ. 0.) DATA(I, J) = (0., 0.)
140 CONTINUE
135 CONTINUE
145 CALL DFLQOT (DATA, MSIZE, ARRAY1, ARRAY2, ARRAYP)
C
C COSINE TAPERING LOW PASS
C THIS IS BY SPECIFYING THE NUMBER OF HARMONICS
C
95 WRITE (6, *) ' WANT TO DO COSINE LOW-PASS? Y/N'
ACCEPT 9030, LP
IF (LP. EQ. 'N') GO TO 100
IF (LP.NE.'Y') GO TO 95
WRITE(6,*) 'CUT-OFF ELEMENT NUMBER:'
ACCEPT *,NCUT
DO 200 I=1,MSIZE
   DO 300 J=1,MSIZE
      DATA(I,J)=DATA(I,J)*WEIGHT(NCUT,I,J,MSIZE)
300    CONTINUE
200   CONTINUE
      CALL DPLT (DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
C
HIGH PASS FILTERING
C AGAIN BY SPECIFYING THE NUMBER OF HARMONICS
C
100 WRITE(6,*) 'HIGH PASS FILTERING?/N'
    ACCEPT 9030,HP
    IF(HP.EQ.'N') GO TO 400
    IF(HP.NE.'Y') GO TO 100
    WRITE(6,*) 'CUT-OFF ELEMENT NUMBER:'
    ACCEPT *,NCUT
    DO 500 I=1,MSIZE
       DO 600 J=1,MSIZE
          DATA(I,J)=DATA(I,J)**2 + (DATA(I,J-MSIZE/2-1))**2
600    CONTINUE
      CONTINUE
      CALL DPLT (DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
C
DO FOURIER TRANSFORM NOW
C
400 WRITE(6,*) 'THE DATA ARE IN TIME DOMAIN NOW!'
    CALL DPLT (DATA,MSIZE,ARRAY1,ARRAY2,ARRAYP)
GO TO 25
9000 FORMAT(A10)
9010 FORMAT(I10)
9020 FORMAT(4E15.8)
9030 FORMAT(A1)
60   STOP
END
SUBROUTINE: DPLT

THIS SUBROUTINE WILL PLOT 3-D FOR A COMPLEX ARRAY DATA
EITHER IN A 3-D PICTURE OR A CONTOUR PLOT
THE CONTOUR PLOT LINKS WITH NCAR PLOTTING PACKAGE

DATA :2-D COMPLEX ARRAY TO BE PLOTTED
MSIZE :SIZE OF DATA
ARRAY1, ARRAY2, ARRAY3: DUMMY ARRAYS WITH SAME DIMENSIONS AS DATA

SUBROUTINE DPLT(DAT, MSIZE, ARRAY1, ARRAY2, ARRAY3)
COMPLEX DATAM, MSIZE)
DIMENSION ARRAY1(MSIZE, MSIZE), ARRAY2(MSIZE, MSIZE), ARRAY3(MSIZE, MSIZE)
CHARACTER ACT, FLO, COM, COM2, COM3, COM4, COM5, LINE, PICK

10 WRITE(6,*) 'DO YOU WANT TO PLOT Y/N'
ACCEPT 9030, COM
IF (COM.EQ. 'N') GO TO 9999
IF (COM.EQ. 'Y') GO TO 10

C

NORMALIZE THE WHOLE DATA FILE BY SOME FACTOR

500 WRITE(6,*) 'DO YOU WANT TO NORMALIZE WITH A FACTOR Y/N'
ACCEPT 9030, COM
IF (COM.EQ. 'N') GO TO 510
IF (COM.EQ. 'Y') GO TO 500
WRITE(6,*) 'FACTOR:
ACCEPT *, FACTOR
GO TO 20

510 FACTOR=1.

20 WRITE(6,*) 'PLOT REAL AND IMAGINARY? Y/N'
ACCEPT 9030, ACT
IF (ACT.EQ. 'N') GO TO 50
IF (ACT.EQ. 'Y') GO TO 20
DO 40 I=1, MSIZE
   DO 60 J=1, MSIZE
      ARRAY1(I, J) = FACTOR*REAL(DAT(I, J))
      ARRAY2(I, J) = FACTOR*AIMAG(DAT(I, J))

60 CONTINUE
40 CONTINUE
GO TO 100

C

CONVERSION INTO LINEAR AMPLITUDE AND PHASE (RADIANS)

50 DO 70 I=1, MSIZE
   DO 90 J=1, MSIZE
      ARRAY1(I, J) = FACTOR*CMBS(DAT(I, J))
      IF ((REAL(DAT(I, J)).EQ.0.) .AND. (AIMAG(DAT(I, J)).EQ.0.)) GO TO 55
      ARRAY2(I, J) = CMAN2(REAL(DAT(I, J)), AIMAG(DAT(I, J)))
   GO TO 90
90 CONTINUE
70 CONTINUE
GO TO 100
```
55  ARRAY2(I,J)=0.
60  CONTINUE
70  CONTINUE
C
C*************
C 100  WRITE(6,*) '1) 3D PICTURE, 2) CONTOUR PLOT'
     ACCEPT 9030, PICK
     IF ((PICK NE '2') .AND. (PICK NE '1')) GO TO 100
101  WRITE(6,*) 'PLOT AMPLITUDE/REAL/Y/N'
     ACCEPT 9030, COM
     IF (COM .EQ. 'N') GO TO 150
     IF (COM .EQ. 'Y') GO TO 101
     IF (PICK .EQ. '1') GO TO 102

C    CONTOUR PLOT
C
C  IF (PICK .EQ. '2') CALL ELowntown (ARRAY1, MSIZE, MSIZE)
   GO TO 150
C
C    PLOT 3 D PICTURE
C
C  120  WRITE (6, *) 'PLOT X-LINE, Y-LINE, OR BOTH X, Y, B'
     ACCEPT 9030, LINE
     IF ((LINE .NE. 'X') .AND. (LINE .NE. 'Y') .AND. (LINE .NE. 'B'))
         GO TO 120
     CALL SET_LINES 3 D (LINE)
     CALL SET_ROTATION 3 D (0)
     XMAX=ARRAY1 (1,1)
     XMIN=ARRAY1 (1,1)
     DO 300 I=1, MSIZE
         DO 310 J=1, MSIZE
             XMAX=MAX1 (ARRAY1 (I, J), XMAX)
             XMIN=MIN1 (ARRAY1 (I, J), XMIN)
             ARRAYP (I, J)=ARRAY (I, J)

310  CONTINUE
300  CONTINUE
301  WRITE (6,*) 'SCALE OF AMPLITUDE/REAL PLOT:'
     WRITE (6,*) '(MAX=', XMAX, ' MIN=', XMIN)
     WRITE (6,*) ' RECOMMEND SCALE: .05/(XMAX-XMIN),')
     ACCEPT *, SCALE
     WRITE (6,*) 'WHAT IS THE VIEW ANGLE W.R.T. X-Y PLANE?'
     WRITE (6,*) '(RECOMMENDING: 45)'
     ACCEPT *, VIEW
     CALL PLOT 3 D SURFACE (ARRAYP, MSIZE, MSIZE, VIEW, 0.0, SCALE)
4001 WRITE (6,*) 'IS THIS THE LAST PLOT? Y/N'
     ACCEPT 9030, PLO
     IF ((PLO .NE. 'Y') .AND. (PLO .NE. 'N')) GO TO 4001
     IF (PLO .EQ. 'N') NSIGN=1
     IF (PLO .EQ. 'Y') NSIGN=999
     CALL PLOT (0.0, 0.0, NSIGN=999)
```
C**********
C PLOT IMAGINARY OR PHASE
C
150 WRITE(6,*) 'PLOT THE PHASE/IMAGINARY? Y/N'
   ACCEPT 9030, COM3
   IF (COM.GE. 'N') GO TO 5000
   IF (COM.NE. 'Y') GO TO 150
   IF (PICT.ED. 'Y') GO TO 157
C CONTOUR PLOT
   CALL EZCRTR(ARRAY2,MSIZE,MSIZE)
   GO TO 5000
C 3 D PICTURE PLOT
157 CALL VPLTOE(0.0,0.0)
220 WRITE(6,*) 'PLOT X-LINE,Y-LINE,OR BOTH? X,Y,B'
   ACCEPT 9030, LINE
   IF ((LINE.NE. 'X') .AND. (LINE.NE. 'Y') .AND. (LINE.NE. 'B')) GO TO 220
   CALL SET_LINES 3D(LINE)
   CALL SET_ROTATION 3D(0)
C FIND THE MINIMUM AND MAXIMUM IN THE PLOTTING SET
C
   XM=
   XM=
   DO 400 I=1,MSIZE
      XMIN=ARRAY2(I,1)
      J=1,MSIZE
      XMAX=MAX1(ARRAY2(I,J),XMAX)
      XMIN=MIN1(ARRAY2(I,J),XMIN)
      ARRAY(I,J)=ARRAY2(I,J)
   CONTINUE
400 WRITE(6,*) 'SCALE OF PHASE/IMAGINARY PLOT: '
   WRITE(6,*) '(MAX= ',XMAX,' MIN= ',XMIN
   WRITE(6,*) 'RECOMMENDED SCALE: ',0.5/(XMAX-XMIN),')'
   ACCEPT 'SCALE'
   WRITE(6,*) 'WHAT IS THE VIEW ANGLE W.R.T. X-Y PLANE? '
   WRITE(6,*) 'RECOMMENDED: 45'
   ACCEPT 'VIEW'
   CALL PLOT_3D_SURFACE(ARRAYP,MSIZE,MSIZE,VIEW,0.,0.,SCALE)
4000 WRITE(6,*) 'IS THIS THE LAST PLOT? Y/N'
   ACCEPT 9030, PLO
   IF ((PLO.NE. 'Y') .AND. (PLO.NE. 'N')) GO TO 4000
   IF (PLO.EQ. 'N') NSIGN=1
   IF (PLO.EQ. 'Y') NSIGN=1
   CALL PLOT(0.,0.,NSIGN,999)
5000 WRITE(6,*) 'PLOT AGAIN? Y/N'
   ACCEPT 9030, COM2
IF (CON2.EQ.'Y') GO TO 100
IF (CON2.NE.'N') GO TO 5000
9030 FORMAT(A1)
9999 RETURN
END
SUBROUTINE: TRANZD

THIS SUBROUTINE WILL FOURIER TRANSFORM ON A COMPLEX ARRAY IN 2D PLANE.
THE PLANE IS ASSUMED TO HAVE THE ORIGIN AT THE CENTRE OF THE PLOT.
THE PLANE HAS EQUAL NUMBER OF UNITS ON EACH SIDE OF THE THE AXES.
I.E. THE SIZE OF THE ARRAY (MSIZE) IS AN ODD NUMBER.
WORKI IS THE WORKING ARRAY WITH SIZE 2**N TO JUST FIT THE DATA ARRAY.

DATA : COMPLEX 2-D ARRAY TO BE TRANSFORMED, AND RESULT STORED
MSIZE : SIZE OF DATA
IFSET : +1. GO TO TIME
         -1. GO TO FREQUENCY

SUBROUTINE TRANZD(DATA,MSIZE,IFSET)
COMPLEX WORK1(32,32),DATA(4MSIZE,MSIZE)
INTEGER INV(8),MM(3)
DIMENSION S(8)
N=(MSIZE+1)/2
MSIZE=32
MM(1)=5
MM(2)=5
MM(3)=0

PLACE THE CENTRED PICTURE INTO THE FOUR CORNERS

DO 10 I=1,MSIZE
      DO 5 J=1,MSIZE
           WORK1(I,J)=(0,0.)
           CONTINUE
      5 CONTINUE
      10 CONTINUE

DO 20 I=1,N
      DO 15 J=1,N
           WORK1(I,J)=DATA(N+I-1,N+J-1)
           CONTINUE
      15 CONTINUE
      20 CONTINUE

DO 30 I=1,N-1
      DO 25 J=1,N-1
           WORK1(MSIZE+I+1,MSIZE+J+1)=DATA(I,J)
           CONTINUE
      25 CONTINUE
      30 CONTINUE

DO 40 I=1,N
      DO 35 J=1,N-1
           WORK1(MSIZE+I+1,J+1)=DATA(I+1,J)
           WORK1(I+1,MSIZE+J+1)=DATA(I+1,J)
           CONTINUE
      35 CONTINUE
      40 CONTINUE

FOURIER TRANSFORM

CALL HARM(WORK1,MM,INV,S,IFSET,IFERR)

188
C C PUT THE FOUR CORNERS BACK TO THE CENTRE
C
DO 120 I=1,N
    DO 115 J=1,N
        DATA (N+I-1,N+J-1) = WORK1 (I,J)
    CONTINUE
115 CONTINUE
120 CONTINUE
DO 130 I=1,N-1
    DO 125 J=1,N-1
        DATA (I,J) = WORK1 (N+I-1,N+J-1)
    CONTINUE
125 CONTINUE
130 CONTINUE
DO 140 I=1,N
    DO 135 J=1,N-1
        DATA (J, I+N-1) = WORK1 (N+I-1,N+J-1)
        DATA (I+N-1, J) = WORK1 (I, J+N+1)
    CONTINUE
135 CONTINUE
140 CONTINUE
RETURN
END
FUNCTION: WEIGHT

This function will calculated the cosine low pass weighting with the assumption that the frequency response data is located in the middle of the plot.

NCUT- cut-off number for the low pass from the center
NX - x-coordinate in the MSIZE X MSIZE
NY - y-coordinate in the MSIZE X MSIZE
MSIZE- size of the array

FUNCTION WEIGHT(NCUT, NX, NY, MSIZE)
CHARACTER LP
PI=3.14159265
X=ABS(FLOAT(MSIZE/2-1))
Y=ABS(FLOAT(MSIZE/2-1))
RADIUS=SQRT(X**2+Y**2)
IF (RADIUS.LE.FLOAT(NCUT)) WEIGHT=0.5*(1.0+CD(FRADIUS/FLOAT(NCUT)))
IF (RADIUS.GT.FLOAT(NCUT)) WEIGHT=0.
RETURN
END
REFERENCES


