PROJECTION OF ACOUSTIC FIELDS USING THE FOURIER TRANSFORM

by

ERROL YUDELMAN, B.Sc. (Elec) Engineering, Cape Town.

Submitted to the University of Cape Town
in partial fulfilment of the requirements for
the degree of Master of Science in Engineering

JUNE 1985

ABSTRACT

A method is presented for projecting acoustic fields using the Fourier transform.

It is shown that the source velocity distribution can be represented by a number of sinusoids. Each sinusoid gives rise to a plane wave whose direction is given by the wavelength of the sinusoid. The plane waves are summed at the plane of interest to obtain the resulting pressure field.

Errors are introduced when projecting pressure fields by large distances. These are circumvented by observing that the farfield pressure pattern can be found by simply taking the Fourier transform of the nearfield pressure distribution.

A rectangular source is simulated on the computer; the Fourier transform technique of projecting fields is compared to a direct integration technique. The Fourier transform technique is used to back project a measured pressure pattern to detect defects on the transducer face.

Measurements of pressure are made in the nearfield of a circular transducer. These measurements are forward and back projected to give the pressure and velocity at other planes.

ACKNOWLEDGEMENTS

My appreciation and thanks to:

Prof. P.N. Denbigh for his invaluable help and guidance,

The members of the Institute for Maritime Technology and the Central Acoustics Laboratory for their support throughout this thesis.

The work was sponsored by the Institute for Maritime Technology, Simon's Town.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. Background

The farfield properties of large underwater transducers are often difficult to determine by direct farfield measurements. It is important to measure the farfield pressure pattern since it is this region in which the transducer normally operates. Were conventional farfield measurements to be carried out, large bodies of water or very large and expensive facilities would be required. The subject of this thesis is to circumvent this problem by projecting nearfield measurements to obtain the farfield beam pattern. This is done by making use of the Fourier transform.

Additional advantages accrue from measurements made in the nearfield. Firstly, while in the sea there are natural factors such as wind, temperature and current which affect calibration measurements, a test tank provides a closely controlled system. The repeatability of measurements is therefore far better in a test tank.

Secondly, when taking measurements close to the transducer, the path difference between the direct pulses and pulses reflected from the surface is large. This eliminates the interference which would result were the measurements taken further from the transducer, where the path differences are small (1).

There are, however, drawbacks to the proposed system. The most significant of these is the conflicting near- and farfield requirement with respect to the area to be covered and the sample spacing. The farfield beam pattern is broad, whereas the nearfield pattern is narrow in comparison. The sample spacing in the nearfield needs to be closer than in the farfield due to subtle changes in phase. Thus, when projecting a nearfield pressure pattern to a farfield pressure pattern, the sample spacing in both planes is determined by the sample spacing requirement for the nearfield. The area to be covered is, on the other hand, governed by the farfield beamwidth.

The projection technique has been used in areas such as optics, holography and acoustics. M.M. Sondhi (2) used sound to reconstruct a shape (the letter H) from measurements of the diffraction pattern of the object when placed in the path of a monocromatic sound wave. Williams, Maynard and Skudrzyk (3) recorded the sound pressure amplitude and phase with a plane array of microphones, then exactly reconstructed the sound sources digitally using a computer. Williams (4) evaluated the radiation from unbaffled, finite plates using an iteration technique in the form of a fast Fourier transform (FFT). A paper which has a good description of the propagation problem is one by Stepanishen and Benjamin (5). In their analysis, the Fourier transform is used to predict the farfield pressure pattern.

The farfield pattern can be produced from the nearfield pattern in other ways. One of these is the use of the Trott nearfield calibration array (NFCA) (6, 7, 8). This is a large planar configuration of small reciprocal transducers whose responses are amplitude shaded. This configuration produces a nearly uniform wave directed normal to the NFCA throughout the volume occupied by the unknown transducer. The method requires the NFCA to be constructed, and the amplitude shading to be calculated for each frequency required, which, as Bobber (1) points out, is an "uncommon and sophisticated engineering task".

Another method was developed at the Defence Research Laboratory (the DRL method) (1). This technique is based on formulae proposed by Helmholtz. The direct use of the Helmholtz formula requires that the magnitude and phase of both pressure and pressure gradient be measured. The pressure gradient is difficult to measure accurately, and so to use this method, an approximation, which introduces errors at angles well off the main beam, has to be made. Another disadvantage is that the DRL method is not flexible. For transducers of different shapes, different integrating techniques are required.

In contrast to the above, the Fourier transform technique investigated in this thesis can be used in real time, with no major construction as is required when the NFCA is used. Neverless the initial expense of constructing an accurate measuring system must be considered. The technique is flexible, so that any shape of transducer can be measured without complication.

1.2. Experimental procedure

Forward projection of a measured nearfield pressure pattern was used to find the farfield pressure pattern. From measurements taken at a distance from the transducer, the back projection technique was used to find the velocity distribution on the face of the transducer. The transducer was used in both masked and unmasked modes. The masked mode involved sticking a strip of non-porous material across the transducer face.

The experiments were carried out in a test tank of dimensions 2m x 1m x 1m. The tank has a gantry system which was used to accurately step a probe in a plane horizontal to the water's surface. A circular transducer of radius 33.5mm was used, transmitting at a frequency of 200KHz. The diameter of the transducer was thus 9.05 wavelengths. The transducer was positioned near the bottom of the tank and radiated upwards. The probe was swept over a plane parallel to the water's surface. The real and imaginary components of the pressure field for each point measured, were transferred to magnetic disk.

1.3. Outline of presentation

The first few chapters of this thesis are concerned with the theoretical development of the equations for the back and forward projection of acoustic fields. Chapter 2 deals with a source that generates a two-dimensional pressure pattern; this theory is extended to a source producing a three-dimensional pressure distribution in chapter 3. Chapter 4 takes a closer look at the equations developed in the previous two chapters, together with the tools needed to perform the simulations. Computer simulations are given in chapter 5. The measurement setup is presented in chapter 6, together with experimentally obtained plots of pressure and velocity patterns. The Fourier technique is assessed in chapter 7, and concluding remarks are made about the applicability of this method to solving the nearfield to farfield problem.

1.4. Definitions and terms used

A sound field produced by an acoustic transducer can be divided into two separate parts. The region in the vicinity of the transducer is called the nearfield, whilst the area at a distance from the transducer is called the farfield. An example of the pressure distribution on axis of a transducer is shown in figure 1.1.

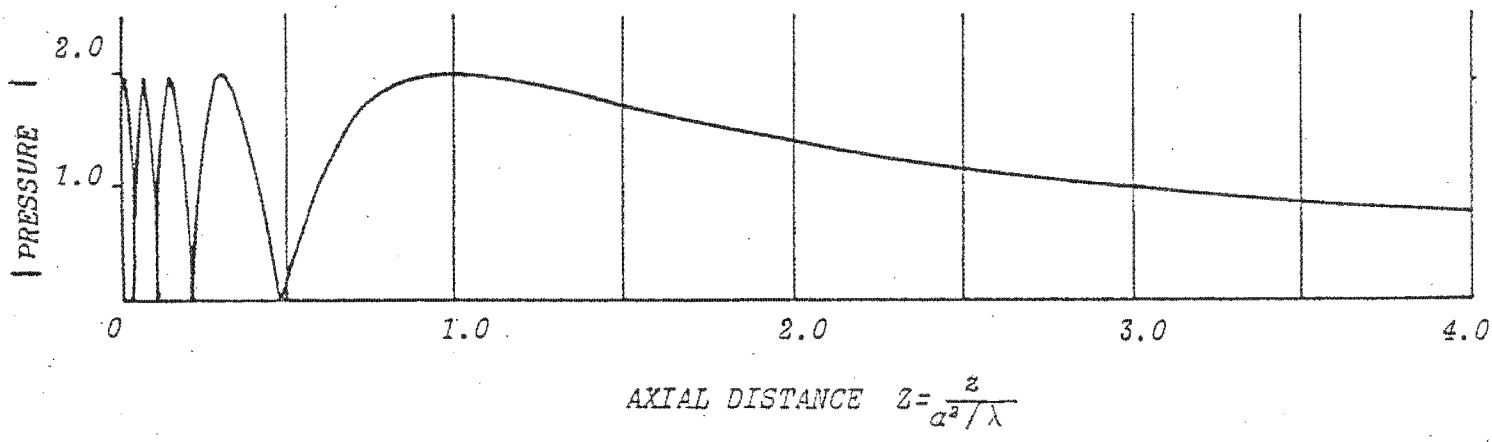


Figure 1.1.: On axis pressure of transducer (after Zemanek (11)).

Authors differ as to the definition of the beginning of the farfield. Clay and Medwin (9), for example, argue that because the last axial maximum occurs at approximately a^2/λ, one needs only to decide the range at which there exists a monotonic decrease close enough to 1/R to satisfy the farfield criteria. They quote the American National Standards Institute range for a circular piston which is

$$R_{far} = \frac{\pi a^2}{\lambda} \qquad (1.1)$$

where a is the radius of the piston
and λ is the wavelength of sound ($\frac{\text{velocity of sound in medium}}{\text{frequency}}$).

For a piston of area A of any shape, the relationship is

$$R_{far} = \frac{A}{\lambda} \qquad (1.2)$$

Zemanek (10), on the other hand, defines the farfield to start at $0.75d^2/\lambda$, where d is the maximum dimension of the transducer.

It was found more convenient to adopt the definition used by Caruthers (12). This is that the farfield begins at

$$R_{far} = \frac{d^2}{\lambda} \qquad (1.3)$$

From Eq. (1.3) it can be seen that if d is large, it becomes the dominant factor. Under these circumstances, therefore, the farfield varies as the square of the diameter. The farfield distance is also inversely proportional to the wavelength. An illustration of the near- and farfield regions is shown in figure 1.2.

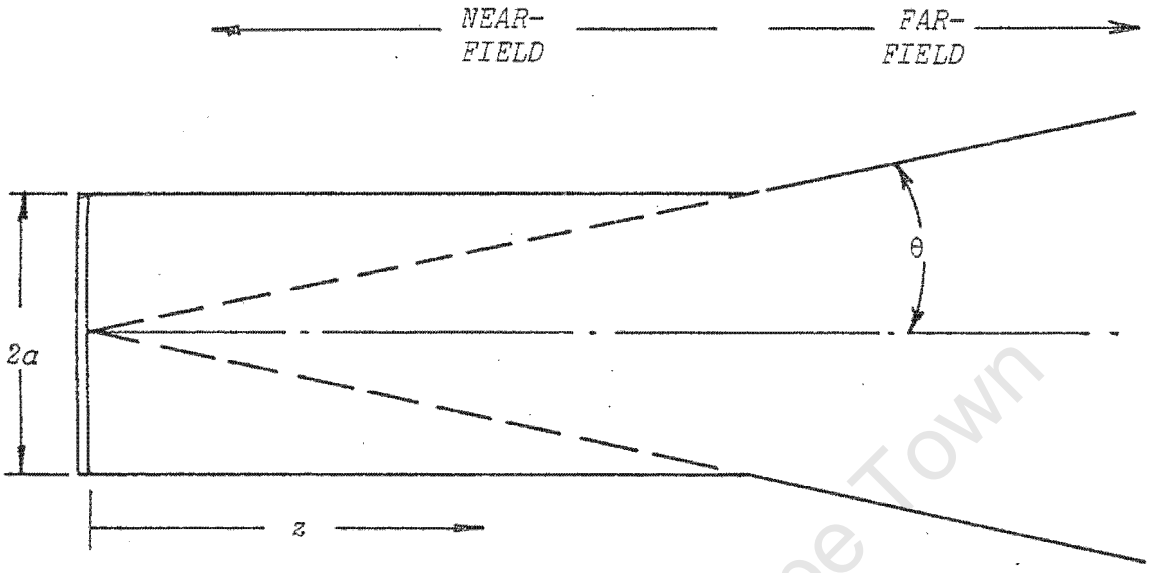


Figure 1.2.: Regions of near- and farfield (after Zemanek (13)).

CHAPTER 2

THE PLANE WAVE APPROACH

2.1. Use of sinusoids to represent the source function

It is well known that the continuous Fourier transform can be used to represent a time waveform in terms of its spectral components. The case of a spatial velocity distribution is similar to that of a time waveform, in that the source can be represented by its spatial frequencies.

If the spatial velocity distribution is represented in a sampled system, then this distribution can be represented as a Fourier series, as long as periodicity can be tolerated.

A source of width d and period T, and associated sinusoids found by calculating the Fourier series of the repeated sources, is shown in figure 2.1.

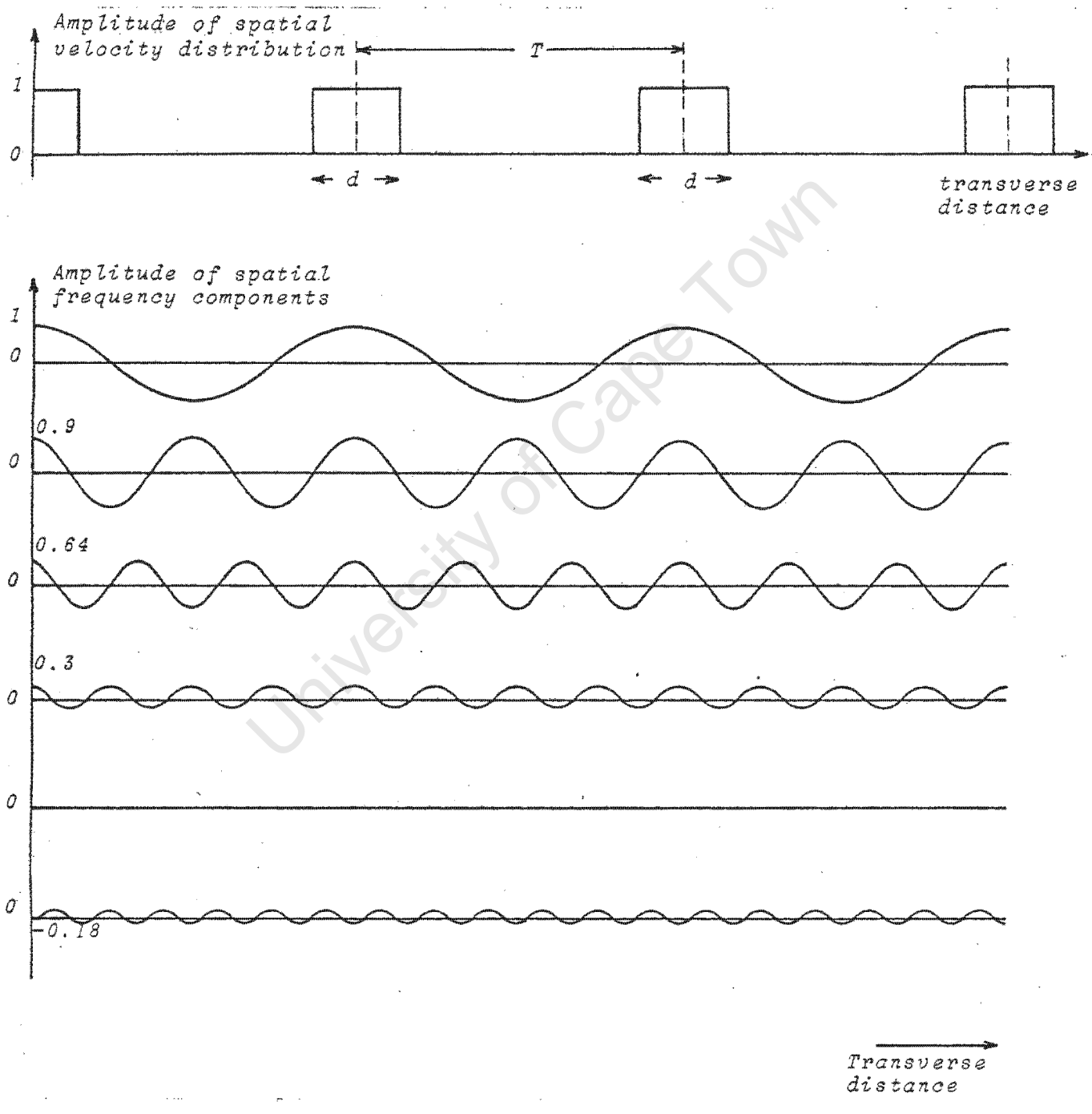


Figure 2.1.: Representation of periodic source with associated sinusoids

2.2. Use of plane waves to solve the projection problem

Kinsler and Frey (14) define a plane wave as a wave with constant amplitude and phase at a plane perpendicular to the direction of wave propagation. Consider the hypothetical example of two propagating plane waves, one at an angle θ to the horizontal, and the other at an angle $-\theta$.

This set of plane waves will interfere at the origin to give a sinusoidal distribution of velocity as shown in figure 2.2.

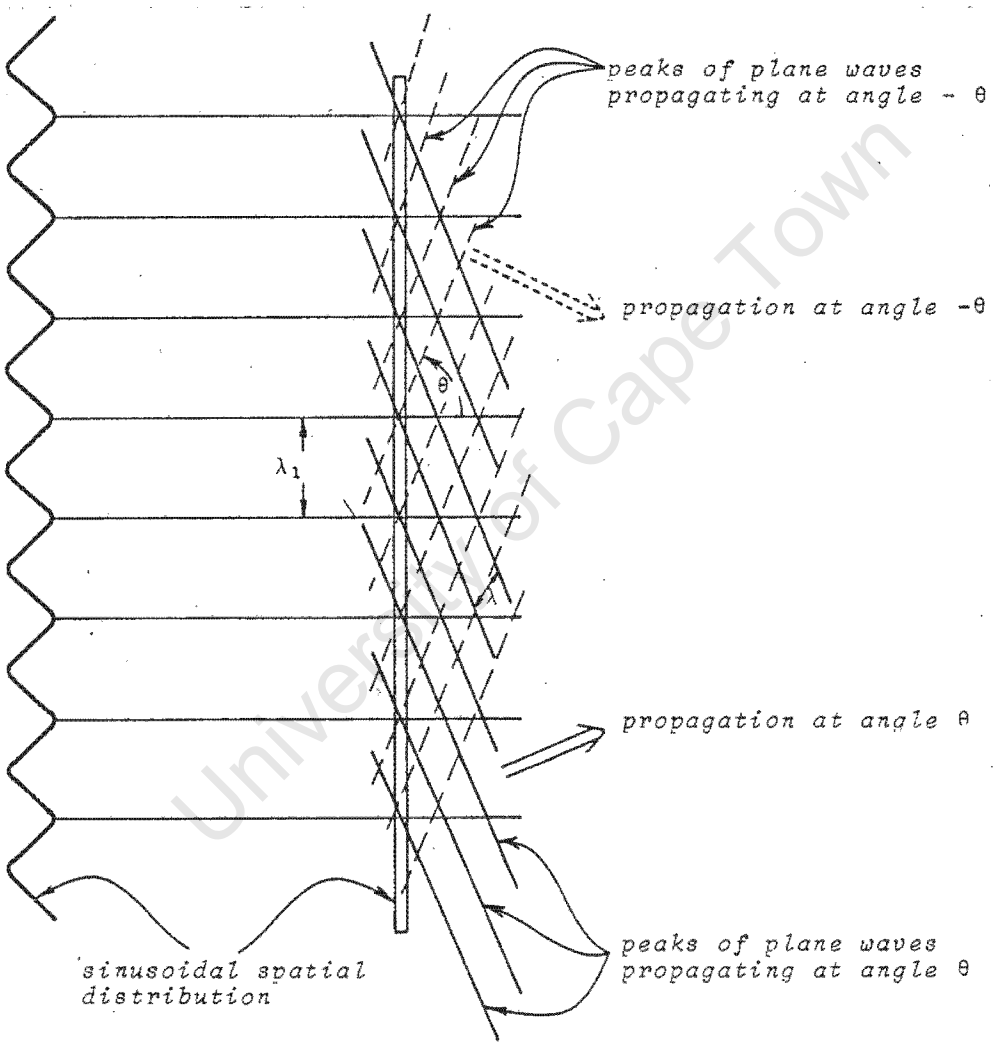


Figure 2.2.: Example of sinusoidal spatial velocity distribution with associated plane waves.

Conversely, a sinusoidal distribution of velocity at the origin will give two propagating plane waves, one at an angle of θ to the horizontal, and the other at an angle $-\theta$ (15).

In general, however, there will not be a single sinusoidal source at the origin, but a more complex function. This function will give rise to many elementary sinusoidal components in the spatial frequency domain. Each component will, in turn, give rise to a set of propagating plane waves.

It can be seen from figure 2.2 that the wavelength of sound propagation is λ, and the wavelength of the sinusoid at the origin is λ_1, and may be termed the spatial wavelength. The angle of propagation of the plane wave is related to the spatial frequency of the sinusoid which causes it. This can be seen from figure 2.3.

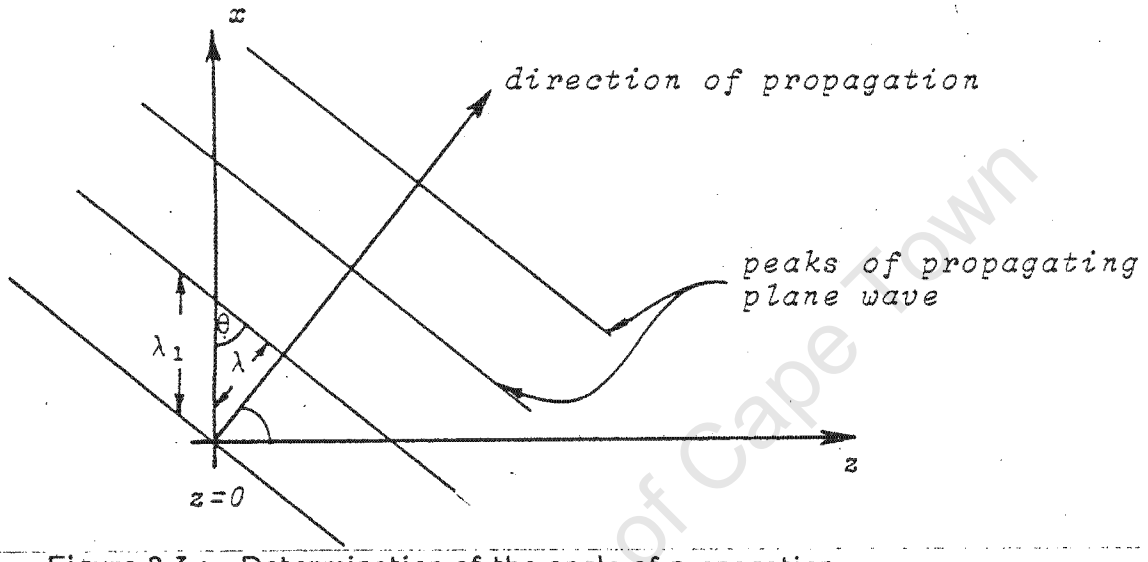


Figure 2.3.: Determination of the angle of propagation

From the figure,

$$\text{Sin } \Theta = \frac{\lambda}{\lambda_1} \tag{2.1}$$

Therefore

$$\Theta = \text{Sin}^{-1}\left(\frac{\lambda}{\lambda_1}\right) \tag{2.2}$$

The only angles at which the plane waves will propagate are those in front of the transducer. Thus Θ lies between -90° and 90°. The angles of propagation can be normalised by using directional cosines instead of angles. A directional cosine is defined as

$$\text{Cos } (90-\Theta) = \text{Sin } \Theta = \ell_x = \frac{\lambda}{\lambda_1} \tag{2.3}$$

where Θ is the angle that the plane wave makes with the horizontal. For real angles, the directional cosine will range from -1 to 1.

The resultant field at a plane other than that for z=0 may be regarded as a superposition of plane waves propagating over a wide spectrum of angles. The effect of a wave travelling from the source plane to the plane of interest, is evaluated by multiplying the plane wave by a transfer function which takes into account the change in phase undergone by each spatial frequency component upon travelling a distance ℓ. Finally, the field at the plane of interest is reconstructed by summing the phase-shifted plane waves.

The change in phase that a plane wave travelling at an angle θ undergoes, can be calculated from figure 2.4. The wave travelling horizontally is taken as the reference.

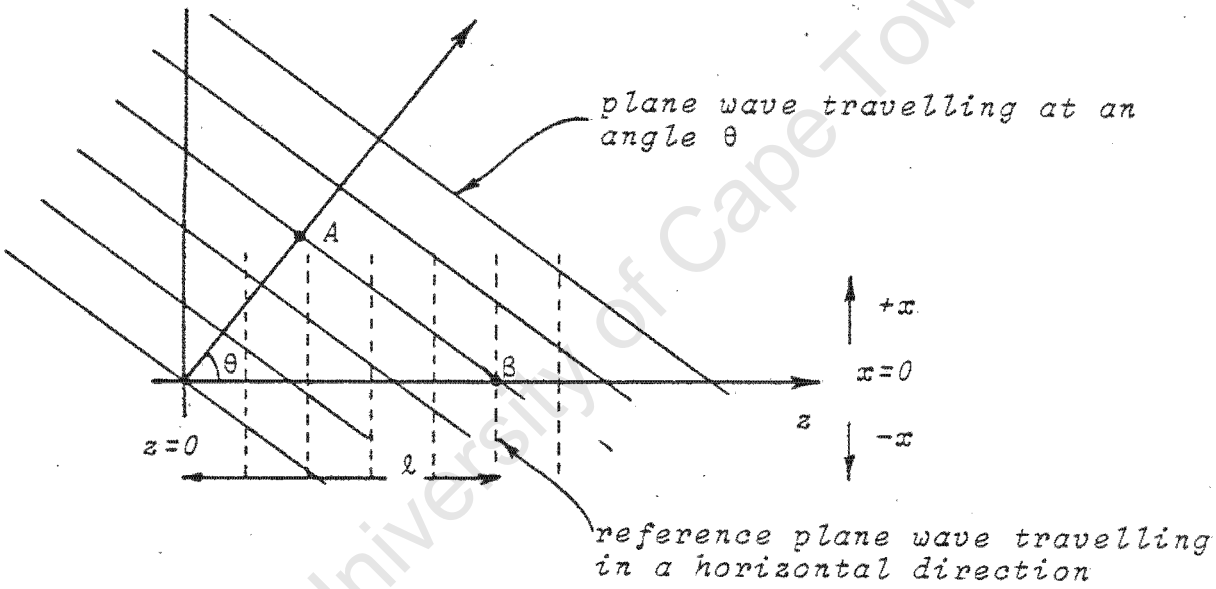


Figure 2.4.: Calculation of the phase of a plane wave travelling at an angle θ.

The horizontal wave has phase dependant only on the distance travelled (in wavelengths).

Thus

$$\text{phase } (\theta = 0) = \frac{\ell}{\lambda} 2\pi \qquad (2.4)$$

The plane wave travelling at an angle θ will have the same phase at points A and B. If the wave has travelled a distance ℓ in the horizontal direction, then the distance travelled at an angle θ is $\ell \cos \theta$. Thus the phase of the wave travelling at an angle θ is

$$\text{phase } (\theta) = \frac{\ell}{\lambda} \text{Cos } \theta \, 2\pi \qquad (2.5)$$

Hence the phase difference at point B between a plane wave travelling at an angle Θ, and a horizontal plane wave, is

$$\frac{\ell}{\lambda} 2\pi \text{ Cos } \Theta - \frac{\ell}{\lambda} 2\pi$$

Therefore

$$\text{phase} = \frac{\ell}{\lambda} 2\pi \text{ (Cos } \Theta - 1) \qquad (2.6)$$

However, since phase is a relative term, this expression can be simplified by assuming that the wave travelling in the horizontal direction has zero phase. The phase difference can thus be interpreted as simply the phase of a wave travelling at an angle Θ.

Thus Eq. (2.6) becomes

$$\text{phase } (\Theta) = \alpha = \frac{\ell}{\lambda} 2\pi \text{ Cos } \Theta \qquad (2.7)$$

The magnitude of the plane waves can be determined by taking the Fourier transform of the original pattern. The Fourier transform of the sinusoid is denoted $\hat{V}(K_1)$, while the magnitude of the velocity distribution at the origin is denoted V(x).

If the pressure distribution, and not the velocity distribution is required at the plane a distance ℓ from the source, the velocity waves must be converted to pressure waves. Kinsler and Frey (16) give the relationship between velocity and pressure plane waves as

$$P = \rho_o cU \qquad (2.8)$$

where U is the velocity in the direction of propagation

$\rho_o c$ is the specific acoustic impedence.

The velocity of excitation of the source is, however, given as a normal velocity, whilst U is the velocity in the direction of propagation. So to use equation (2.8), the component of the velocity at an angle Θ must be determined from the normal velocity. This can be done with reference to figure 2.5.

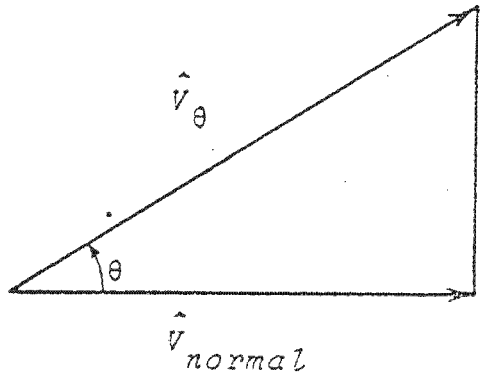


Figure 2.5.: Relationship of $\hat{V}_\theta$ to $\hat{V}_{normal}$.

The velocities in the above figure are shown as Fourier transform velocities, because they are the magnitudes of the plane waves.

Hence

$$\hat{V}_\theta = \frac{\hat{V}_{normal}}{\text{Cos } \theta} \tag{2.9}$$

A relationship between cos θ, λ and λ_1 can be obtained from figure 2.3.

From the diagram,

$$\text{Cos } \theta = \frac{(\lambda_1^2 - \lambda^2)^{1/2}}{\lambda_1} \tag{2.10}$$

Therefore

$$\text{Cos } \theta = (1-(\frac{\lambda}{\lambda_1})^2)^{1/2} \tag{2.11}$$

Thus

$$\text{Cos } \theta = \lambda((\frac{1}{\lambda})^2 - (\frac{1}{\lambda_1})^2)^{1/2}$$

$$= \frac{\lambda}{2\pi}((\frac{2\pi}{\lambda})^2 - (\frac{2\pi}{\lambda_1})^2)^{1/2}$$

and since $k = \frac{2\pi}{\lambda}$ and $k_1 = \frac{2\pi}{\lambda_1}$

$$\text{Cos } \Theta = \frac{(k^2 - k_1^2)^{1/2}}{k} \tag{2.12}$$

and similarly

$$\text{Sin } \Theta = \frac{\lambda}{\lambda_1} = \frac{k_1}{k} = \frac{f_1}{f} \tag{2.13}$$

So Eq. (2.9) becomes

$$\hat{V}_\Theta = \frac{\hat{V}_{normal}\, k}{(k^2 - k_1^2)^{1/2}}$$

Therefore Eq. (2.8) becomes

$$\hat{P} = \frac{\rho_o c k \hat{V}_{normal}}{(k^2 - k_1^2)^{1/2}} \tag{2.14}$$

Since the phase must also be taken into account, from Eq. (2.7)

$$\alpha = \frac{\ell}{\lambda} \text{Cos } \Theta \, 2\pi \tag{2.7}$$

$$= \frac{\ell}{\lambda} \frac{2\pi (k^2 - k_1^2)^{1/2}}{k}$$

$$\alpha = \ell(k^2 - k_1^2)^{1/2} \tag{2.15}$$

Combining Eqs. (2.14) and (2.15)

$$\hat{P} = \frac{\rho_o c k \hat{V}_{normal}}{(k^2 - k_1^2)^{1/2}} e^{-j\ell(k^2 - k_1^2)^{1/2}} \tag{2.16}$$

Equation (2.7) gives the phase of a wave travelling at an angle Θ at x=0 and with x=ℓ. For values of x not equal to 0, there is an additional phase factor. Since the phase delay in the x-direction is $2\pi x/\lambda_1$ radians, the phase expressed as an exponential is $e^{-jk_1 x}$, where k_1 is the wavenumber in the x-direction.

The pressure $\hat{P}(x,k_1)$ due to one plane wave will be

$$\hat{P}(x,k_1) = \rho_0 ck \frac{\hat{V}_{normal}}{(k^2-k_1^2)^{1/2}} e^{-j\ell(k^2-k_1^2)^{1/2}} e^{-jk_1 x} \tag{2.17}$$

Since there are in general many plane waves travelling in the positive z-direction, they must be summed, giving

$$P(x) = \rho_0 ck \sum_{i=1}^{N} \frac{\hat{V}_{normal}}{(k^2-k_1^2)^{1/2}} e^{-j\ell(k^2-k_1^2)^{1/2}} e^{-jk_1 x} \tag{2.18}$$

From this equation it can be seen that for every point (x) at which the pressure is to be calculated, there are N summations.

Until now, only the discrete case, as would be carried out on a digital computer, has been studied. By summing over infinitely small changes in the directional cosine dk_1, Eq. (2.18) becomes

$$P(x) = \rho_0 ck \int_{-\infty}^{\infty} \frac{\hat{V}(k_1)}{(k^2-k_1^2)^{1/2}} e^{-j\ell[(k^2-k_1^2)^{1/2}+k_1 x]} dk_1 \tag{2.19}$$

This will give the pressure, as a function of x, from the velocity distribution at the origin.

A relationship between the pressure at a reference plane and the pressure at any other plane can be obtained by substituting Eq. (2.14) into Eq. (2.19).

Thus Eq. (2.19) becomes

$$\begin{matrix} & & & \text{(i)} & & \text{(ii)} & & \text{(iii)} \end{matrix}$$

$$P(x) = \int_{-\infty}^{\infty} \hat{P}(k_1)\, e^{-j\,[\ell(k^2-k_1^2)^{1/2} + k_1 x]}\, dk_1 \tag{2.20}$$

It can be seen from Eq. (2.20) that there are three phase terms involved in the computation:

(i) The phase of the wave relative to a reference at the plane in which the measurements were made. This is given by the complex Fourier transform of the pressure.

(ii) The phase with respect to the distance of the desired plane (ℓ).

(iii) The phase introduced by computing the pressure at a point on plane B for x not equal to 0.

2.3. The Fourier Transform

Stepanishen and Benjamin (5) give the definition of the Fourier transform as

$$F(f(t)) = F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{j\omega t} dt \qquad (2.21)$$

Substituting

t by x

and ω by k_1,

Eq. (2.21) becomes

$$F(f(x)) = F(k_1) = \int_{-\infty}^{\infty} f(x)\, e^{jk_1 x} dx \qquad (2.22)$$

or inversely

$$F(x) \int_{-\infty}^{\infty} F(k_1)\, e^{jk_1 x} dk_1 \qquad (2.23)$$

Thus equation (2.20) is in the form of the Fourier transform. In particular the fast Fourier transform (FFT) will be used, which is a fast algorithm for solving the DFT. The FFT uses a factoring technique to reduce the number of multiplications and additions to be performed. For the straightforward DFT, approximately N^2 operations are required, whereas the FFT only requires $2N*\log_2 N$ operations (17). The difference in computational time is illustrated in figure 2.6. It can be seen that for large N, the FFT reduces computational time by a significant amount.

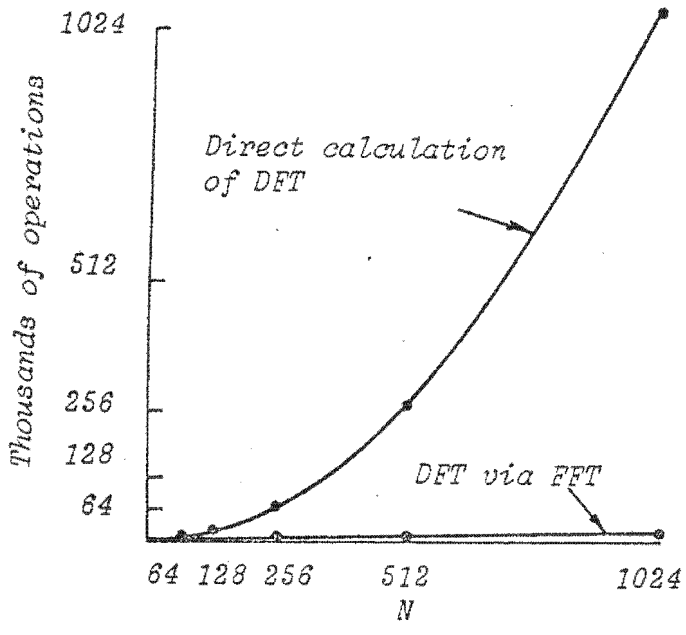


Figure 2.6.: The number of operations required for computing the discrete Fourier transform using the FFT algorithm compared with the number of operations required for direct calculation of the discrete Fourier transform (after Bergland (18)).

CHAPTER 3

A THREE-DIMENSIONAL SYSTEM

The two-dimensional model developed in the previous chapter introduced the use of plane waves to solve the near- to farfield problem. The relationship between spatial frequency and the direction of the propagating plane waves was formulated, and the pressure at a distance ℓ from the source was expressed in terms of the velocity of the source for a two-dimensional distribution.

This chapter aims to derive a relationship between the velocity at the source, and the pressure at a distance ℓ from it, for a three-dimensional model. This will be done by making use of the concepts developed in chapter two. An expression is given which uses a single Fourier transform to derive the farfield pressure directly from the velocity distribution at the source. The chapter concludes with a brief comparison between an optical and an acoustical system.

3.1. Solution of the wave equation

The three-dimensional wave equation of a mechanical radiating source vibrating at a frequency f is

$$\nabla^2 P(x) + k^2 P(x) > 0 \qquad (3.1)$$

where P(x) is the pressure measured in a plane horizontal to the transducer face and k is the wavenumber = $\frac{2\pi}{\lambda}$ measured in radians/metre.

Assume there is a plane wave P(x) satisfying the equation

$$P(x) = \exp(j(k_1 x + k_2 y + k_3 z)) \qquad (3.2)$$

where k_1, k_2 and k_3, are the wavenumbers in the x,y and z directions respectively.

Substituting Eq. (3.2) into Eq. (3.1) gives

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2} + k^2 P = 0 \qquad (3.3)$$

Therefore

$$-k_1^2 P - k_2^2 P - k_3^2 P + k^2 P = 0$$

Therefore

$$P(k^2 - k_1^2 - k_2^2 - k_3^2) = 0$$

Thus

$$k^2 = k_1^2 + k_2^2 + k_3^2 \qquad (3.4)$$

Substituting this back into Eq. (3.2) gives

$$P(x) = \exp(-j(k_1 x + k_2 y + k_3 z)) \qquad (3.2)$$

$$= \exp(-j(k_1 x + k_2 y + (k^2 - k_1^2 - k_2^2)^{1/2} z) \qquad (3.5)$$

Now let $\Delta z = z_{ref} - z_{desired} = z$, which is the distance from the measurement plane to the desired plane.

Therefore

$$P(x) = \exp(j(k_1 x + k_2 y + (k^2 - k_1^2 - k_2^2)^{1/2} \Delta z)) \qquad (3.6)$$

The general solution can be found by superposing such plane waves.

$$P(x,y,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(k_1, k_2) \exp(-j(k_1 x + k_2 y + (k^2 - k_1^2 - k_2^2)^{1/2} \Delta z))\, dk_1 dk_2 \qquad (3.7)$$

with $f(k_1, k_2)$ chosen arbitrarily.

3.2. Farfield from nearfield velocity distribution

Kinsler and Frey (19) give the relationship between the velocity distribution on the source (in rectangular coordinates) and the farfield pressure pattern (in polar coordinates). Stepanishen and Benjamin (5) give the relationship between velocity and pressure for large values of z for any transducer shape. This relationship is derived in Appendix A using the method of stationary phase. It will be shown in chapters 5 and 6 that this is the easiest method of obtaining the farfield pressure pattern.

Eq. (A.21) gives

$$P(R,\Theta,\phi) = j\,ka\rho c\,(e^{-jkaR}/_{2\pi R}) * \hat{V}\,(ka\,\mathrm{Sin}\,\Theta\,\mathrm{Cos}\,\phi,\ ka\,\mathrm{Sin}\,\Theta\,\mathrm{Sin}\,\phi,\ x_s'') \qquad (3.13)$$

In 2-dimensions ($\phi = 90°$)

$$P(R,\Theta) = jka\rho c\,(e^{-jkaR}/_{2\pi R}) \times \hat{V}\,(ka\,\mathrm{Sin}\,\Theta,\ x_s'') \qquad (3.14)$$

The Fourier transform of the velocity yields the plane waves which emanate from the source. These propagate at an angle Θ to the horizontal. The farfield pressure magnitude in that direction is given by the plane wave's magnitude divided by $2\pi R$ and modified by the constant $ka\rho c$. Thus if the velocity distribution at any plane in the nearfield is measured, the farfield distribution can readily be obtained.

3.3. Comparison between optical and acoustic reconstruction processes

The acoustic and optical reconstruction processes are similar. The property of linearity is common to both systems - the response to a complicated stimulus can be expressed in terms of certain 'elementary' stimuli.

Whereas a lens can be used to bring an object placed at infinity to the front focal plane of it, the Fourier transform technique attempts to project a field close to the object to a field a large distance away. One of the most remarkable and useful properties of a converging lens is its inherent ability to perform two-dimensional Fourier transforms. This complicated analog operation can be performed with extreme simplicity in a coherent-optical system.

In appendix B it will be shown that the equations for optical and acoustic systems are similar for the propagation of plane waves from one plane to another. A coherent-optical processor is constructed to show that the acoustic system is a special case of generalized optical diffraction theory.

CHAPTER 4

THE TRANSFORMATION EQUATION AND SAMPLING

The following relationships between pressure and velocity in two planes have been derived.

In chapter 2,

$$P(x, z) = \rho_o ck \int_{-\infty}^{\infty} \frac{\hat{V}(k_1, z)}{(k^2 - k_1^2)^{1/2}} e^{-j(\ell (k^2 - k_1^2)^{1/2} k_1 x)} dk_1 \qquad (2.19)$$

and in chapter 3,

$$P(x_1, x_2, x_3') = \rho_o ck * \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\hat{V}_{normal}(k_1, k_2)}{(k^2 - k_1^2 - k_2^2)^{1/2}} \exp(-j(k_1 x + k_2 y + (k^2 - k_1^2 - k_2^2)^{1/2}\Delta z))\, dk_1\, dk_2 \qquad (3.11)$$

It was shown in these chapters, that the above formulae are in the form of the Fourier transform. There is also a phase factor in both formulae. For two-dimensions this factor is

$$M = e^{-j\ell (k^2 - k_1^2)^{1/2}} \quad \text{in Eq. (2.19)} \qquad (4.1)$$

and for three-dimensions the phase factor is

$$N = e^{-j\ell (k^2 - k_1^2 - k_2^2)^{1/2}} \quad \text{in Eq. (3.11)} \qquad (4.2)$$

It can be seen from Eq. (4.1) that M is a phase term (magnitude = 1) for $k_1 \leq k$.

For $k_1 > k$, the quantity under the radical is negative, hence M is real. Waves for which M is real (evanescent waves) travel in the x-y plane and are exponentially damped in the z-direction. These waves carry information about the details of the source, whose size is less than one wavelength. Since they are strongly attenuated in the z-direction, they will, in practice, not be recorded.

The relationship between M and the directional cosine can be found by simplifying Eq. (4.1).

$$M = e^{-j\frac{\ell}{k}\left(1-\left(\frac{k_1}{k}\right)^2\right)^{1/2}}$$

$$= e^{-j\frac{\ell}{k}\left(1-\ell_x^2\right)^{1/2}} \qquad (4.3)$$

Since

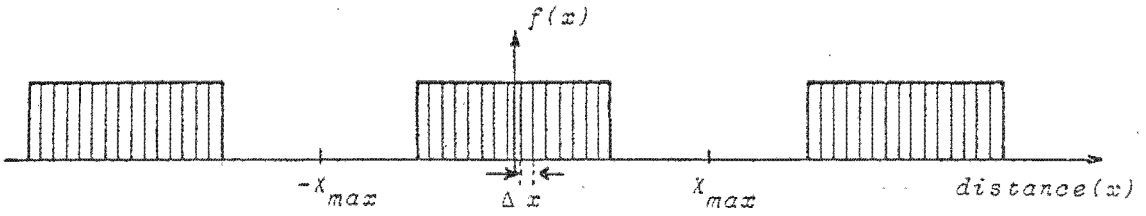
$$\ell_x = \text{Cos}(90-\theta), \quad -1 \leq \ell_x \leq 1.$$

So to realize M physically, $\frac{k_1}{k}$ must be less than or equal to 1, thus $k_1 \leq k$.

4.1. Relationship between a spatial waveform and its spatial frequency waveform

The sampling theorem states that if the Fourier transform of a function f(x) is zero for all spatial frequencies greater than a certain spatial frequency $f_{1c} = k_{1/2\pi}$, then the continuous function f(x) can be uniquely determined from a knowledge of its sampled values. The Nyquist sampling rate gives f_{1c} as $\frac{1}{2\Delta x}$, where Δx is the sampling interval in the spatial domain.

An example of a sampled rectangular waveform with its sampled spatial frequency distribution is shown in figure 4.1.



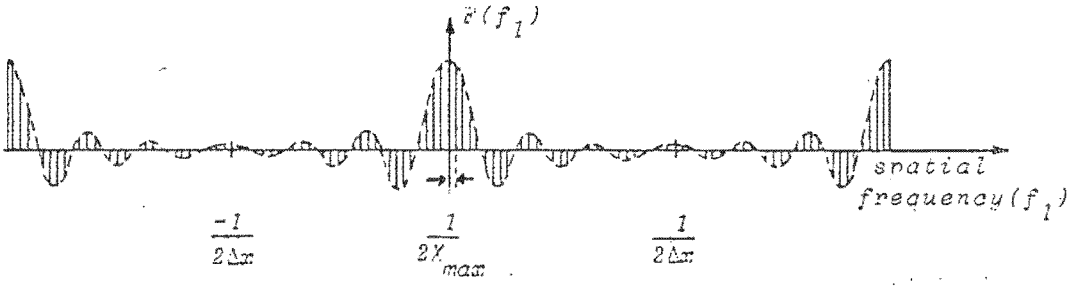


Figure 4.1.: Relationship between spatial and spatial frequency data.

A relationship between sampling rate, frequency and directional cosine can now be formulated.

From Eq. (2.3), $\ell_x = \frac{\lambda}{\lambda_1}$.

Thus $\ell_x = \frac{f_1}{f}$

For $\ell_x = 1$, $f_1 = f_{1C} = f$

Therefore $\frac{1}{2\Delta x} \leq f_{1C} = f$

Thus $\Delta x \geq \frac{\lambda}{2}$ (4.4)

which is the ideal sampling rate.

Figure 4.2 illustrates three spatial functions and their associated spatial frequency distributions.

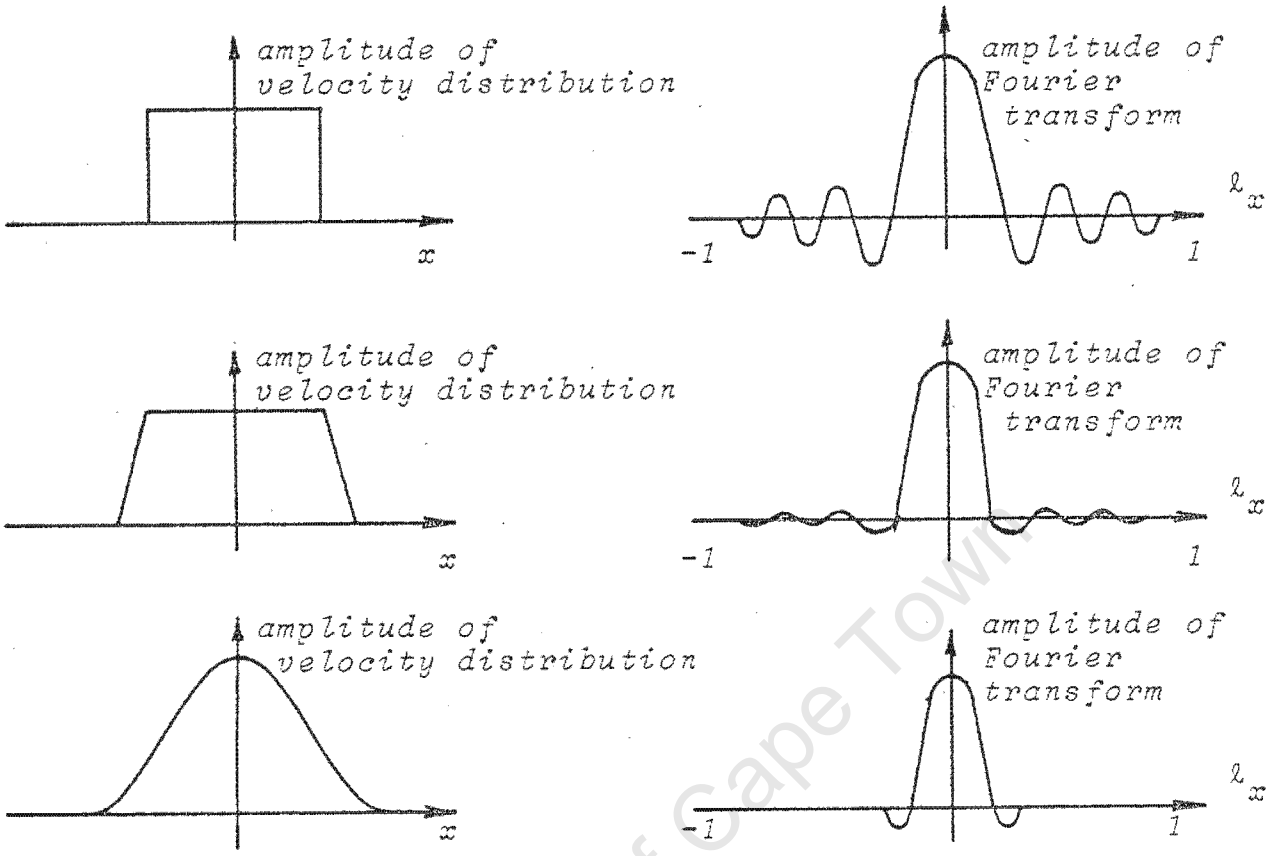


Figure 4.2.: Examples of spatial waveforms and associated Fourier transforms.

Figure 4.2(a) shows the high frequency components associated with sharp edges in the spatial domain. Thus for $\ell_x \geq 1$, the amplitude of the Fourier transform is still significant. The truncation in the spatial frequency domain causes errors in the projected spatial pattern when simulations are performed on a computer. A window will be placed on the spatial waveform to eliminate the high frequency components, as will be shown in chapter 5.

Figure 4.2(b) shows the ideal case, where the amplitude of the frequency distribution has died out sufficiently when $\ell_x = 1$.

Figure 4.2(c) illustrates that the bandwidth of the Fourier transform data could be reduced (i.e. $-0.4 < \ell_x < 0.4$). Thus fewer points could be used to describe the velocity distribution and hence computer time and storage could be saved.

4.2. Discrete Fourier transform criteria

When the discrete Fourier transform is used to solve the propagation problem, proper attention has to be paid to the questions of repeated sources and aliasing.

4.2.1. Repeated sources

The discrete Fourier transform procedure with uniform sampling actually solves the problem of an infinite array of identical sources placed side by side. Such an array with separation equal to N*S, where N is the total number of points and S is the sampling inteval, is shown in Figure 4.3.

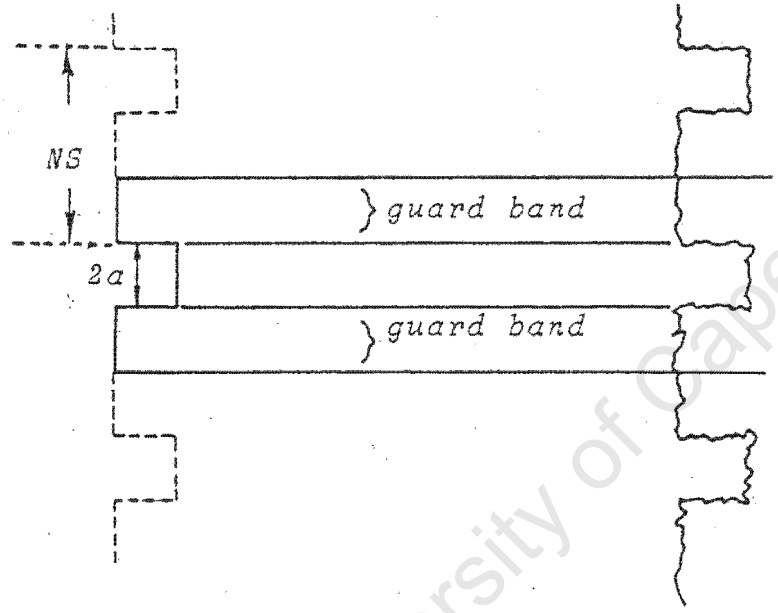


Figure 4.3.: Illustration of the periodic sources inherent in discrete Fourier transform calculations.

Figure 4.3 shows the periodic sources in two dimensions. However, the theory to be developed can easily be extended to three-dimensions if required. A guard band has been introduced in Figure 4.3 to prevent the energy from one source from spilling into the source alongside it. If the guard band is not large enough for the distance at which the projection has been calculated, errors in the calculation of the pressure distribution will occur, as illustrated in Figure 4.4.

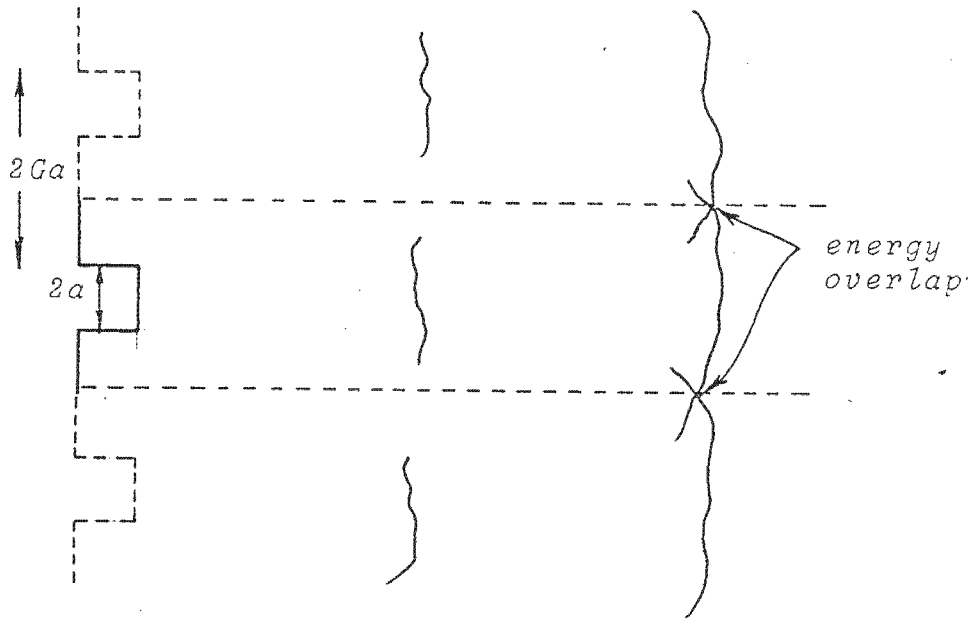


Figure 4.4.: Showing overlap of energy resulting from a small guard band

The guard band factor, G, will be used to develop equations to find the energy overlap between side by side sources.

Sziklas and Siegman (20) have solved the problem of energy overlap. They give a formula which relates guard band (G), amount of energy overlap (ϵ_1), maximum propagation distance (L) and radius of the source (a).

$$G \geq 1 + \frac{L\lambda}{2\pi^2 a^2 \epsilon_1} \qquad (4.5)$$

Rewriting Eq. (4.5)

$$L \leq \frac{(G-1)\, 2\pi^2 a^2 \epsilon_1}{\lambda} \qquad (4.6)$$

from which the minimum distance can be calculated for a given guard band and energy overlap.

Example:

Source size	=	10 λ	(2a)
Sampling interval	=	0.5 λ	(S)
N	=	128	

So $2Ga = 128$ points

and $Ga = 2a = 20$ points

Therefore $G = 6.4$

From Eq. (4.6),

$$L \leq \frac{5.4 * 2\pi^2 * (5\lambda)^2 \varepsilon_1}{\lambda}$$

Therefore $L \leq 2665\ \varepsilon_1$ (in wavelengths)

If $\varepsilon_1 = 1.0\%$, $L \leq 26.7$ wavelengths

If $\varepsilon_1 = 0.5\%$, $L \leq 13.3$ wavelengths

For the simulations presented in the following chapter, the allowable energy overlap will be 0.5%.

Since for this example,

$$d_{far} = \frac{d^2}{\lambda} = \frac{(2a)^2}{\lambda} = \frac{10^2 \lambda^2}{\lambda} = 100 \text{ wavelengths,}$$

the guard band would have to be increased for the energy overlap to be reduced to 0.5%.

Therefore

$$G \geq 1 + \frac{100\lambda^2}{2\pi^2 (5\lambda)^2 (0.005)}$$

$$\geq 41.53$$

which means that at least $2Ga = 20*41.53 = 831$ points are required.

Of course, there will have to be a trade off between energy overlap and the number of points to be used, as storage space in a computer could be limited.

Another approach was used to verify that Eq. (4.5) gives the correct results. Assume that the beam spreads out according to figure 1.2. By considering the geometry of the multiple source problem given in figure 4.5, the approximate distance from the source at which overlapping will occur can be calculated.

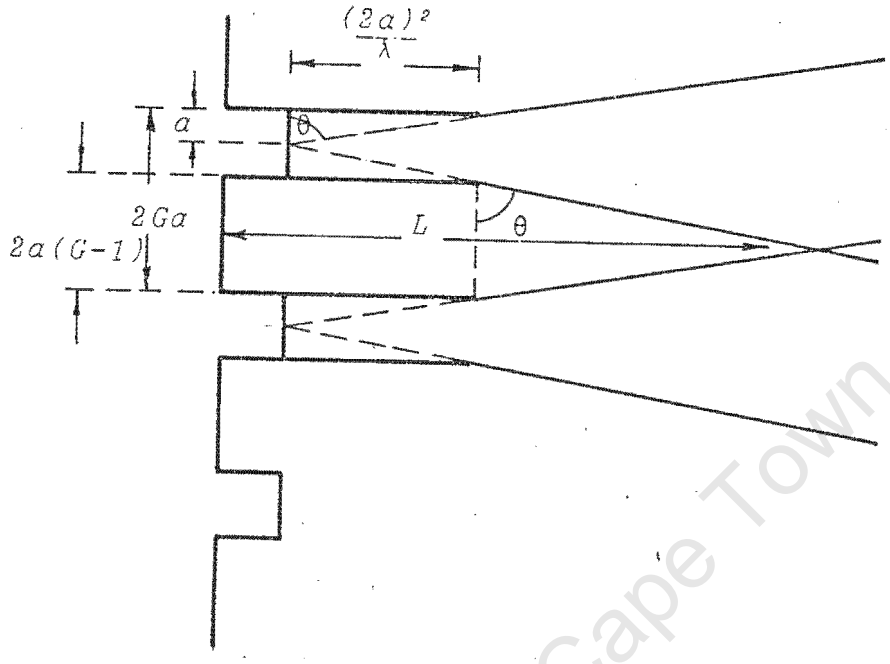


Figure 4.5: Showing overlapping beam patterns from classical near- farfield diagram

From the figure,

$$\text{Tan}\,\theta = \frac{\frac{(2a)^2}{\lambda}}{a} = \frac{4a}{\lambda}$$

and $$\text{Tan}\,\theta = \left(L - \frac{4a^2}{\lambda}\right) / (a\,(G-1))$$

Therefore for no overlap,

$$\frac{\left(L - \frac{4a^2}{\lambda}\right)}{a\,(G-1)} \leq \frac{4a}{\lambda}$$

Therefore

$$\left(L - \frac{4a^2}{\lambda}\right) \leq \frac{4a^2}{\lambda}(G-1)$$

So $$L \leq \frac{4a^2 G}{\lambda}$$

Take G = 6.4 as given in the previous example,

$$L \leq \frac{4 * (5\lambda)^2 * 6.4}{\lambda}$$

$$\leq 640\ \lambda$$

The result obtained is equivalent to having ε_1 = 24%. This shows that the assumption that the beam stays collimated until $(2a)^2/\lambda$, and then spreads out at an angle Θ, is not entirely correct. This estimate of the energy overlap between two sources is very conservative. The choice of the farfield distance as $(2a)^2/\lambda$ could also influence the results obtained using the collimated beam approach, but does not affect Sziklas and Siegman's derivation.

4.2.2. Sampling interval requirement

The sampling theorem and Nyquist rate have been discussed in section 4.1. If the sampling rate is too low, the high frequency components of a spatial function can impersonate low frequencies. This is demonstrated in Figure 4.6 by showing a relatively high frequency and a relatively low frequency that share identical sample points. Thus it is required that the sampling rate be high enough for the highest frequency to be sampled at least twice during each cycle. If this sampling rate is not satisfied, there will be aliasing of the higher frequency components back into lower frequency components.

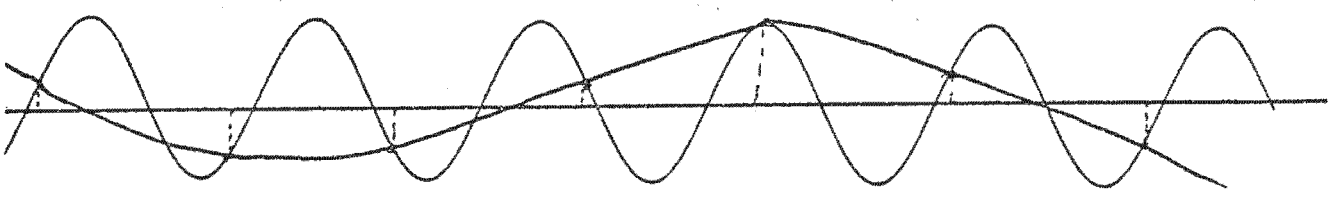


Figure 4.6: Example of a high frequency emulating a lower frequency.

There will always be some aliasing when a signal is sampled and the Fourier transform performed. Sziklas and Siegman give the relationship for the amount of energy overlap of higher spatial frequencies back into lower spatial frequencies:

$$N_p \geq \frac{2G}{\pi^2 \varepsilon_2} \tag{4.8}$$

where N_p is the number of points chosen for the DFT

G is the guard band as defined in Eq. (4.5)

ε_2 is the fractional amount of energy overlap that is aliased from higher spatial frequencies back into lower spatial frequencies.

Thus for G = 6.4,

$$N_p \geq \frac{2 * 6.4}{\pi^2 \varepsilon_2} \geq \frac{1.3}{\varepsilon_2}$$

Therefore for $N_p = 128$, $\varepsilon_2 = 0.01 = 1\%$. It can therefore be concluded that 1% of the energy present in the frequencies above the Nyquist sampling rate is aliased back into frequencies below the Nyquist sampling rate.

In the case of a transducer transmitting plane waves at various spatial frequencies (or angles), there is an inherent limitation in the bandwidth. This is because the waves cannot propagate at angles greater than 90° or less than -90°. If the aperture used is square, with sharp edges, and the frequency spectrum has not died away sufficiently before the edge of the bandwidth, problems could occur. One way of solving those problems is to use a window, such as a Tukey, Hanning or Hamming window. This technique will introduce errors due to the change of shape of the input function, but after the transformation has been carried out, the magnitude of the error could be reduced. Examples of the use of a window are given in the following chapter.

CHAPTER 5

SIMULATION OF FORWARD AND BACK PROJECTION TECHNIQUES ON A COMPUTER

5.1 Introduction

Techniques used to simulate the reconstruction process, are presented in this chapter.

The normal velocity of the source is generated on a computer. The pressure at another plane is found by applying the forward projection formulae to the source velocity. The pressure distribution calculated by the Fourier transform technique is compared with that calculated by direct integration, as described in section 5.2. The pressure data generated by applying this integration technique are used to determine the velocity distribution of the source. This is achieved by applying the back projection technique.

5.2 The integration technique

The integration technique computes the pressure at a field point by summing the pressures produced by the individual sources of the radiating object. This is achieved by considering a continuous source as comprising a number of discrete point sources, as shown in Figure 5.1. The source is in plane A, while the plane in which the pressure is to be found is plane B.

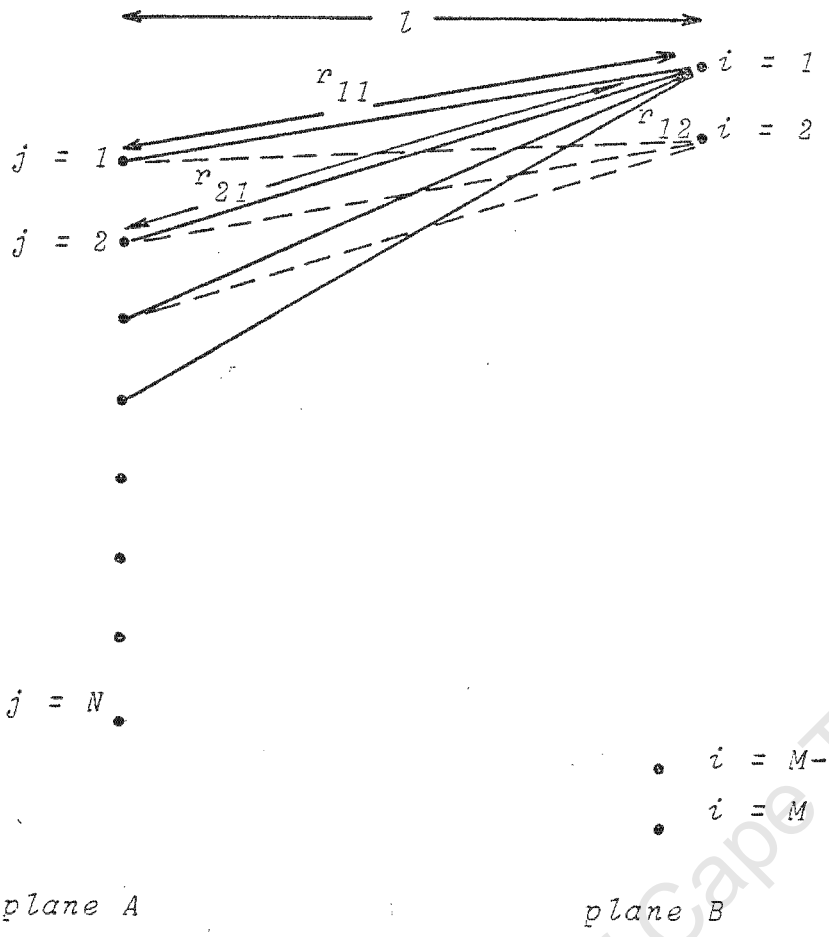


Figure 5.1: A line array of simple sources from which the pressure is found.

Assuming that the waves leaving the source are in phase, the expression for the sum is given by Eq. (5.1):

$$P_i = \rho_o c k \sum_{j=1}^{N} \frac{A_j e^{j\omega t_o} e^{jk r_{ji}}}{r_{ji}} \qquad (5.1)$$

where A_j is the normal velocity component of the jth element

r_{ji} = the distance from the jth point on the source to the ith point on plane B.

If a unit normal velocity is assumed (i.e. $A_j = 1$ for all j), and neglecting the time factor, Eq. (5.1) becomes:

$$P_i = \rho_o c k \sum_{j=1}^{N} \frac{e^{j k r_{ji}}}{r_{ji}} \qquad (5.2)$$

This gives the pressure on plane B for a line array at the source.

For the more general case of a planar array and a two-dimensional plane of interest, consider Figure 5.2.

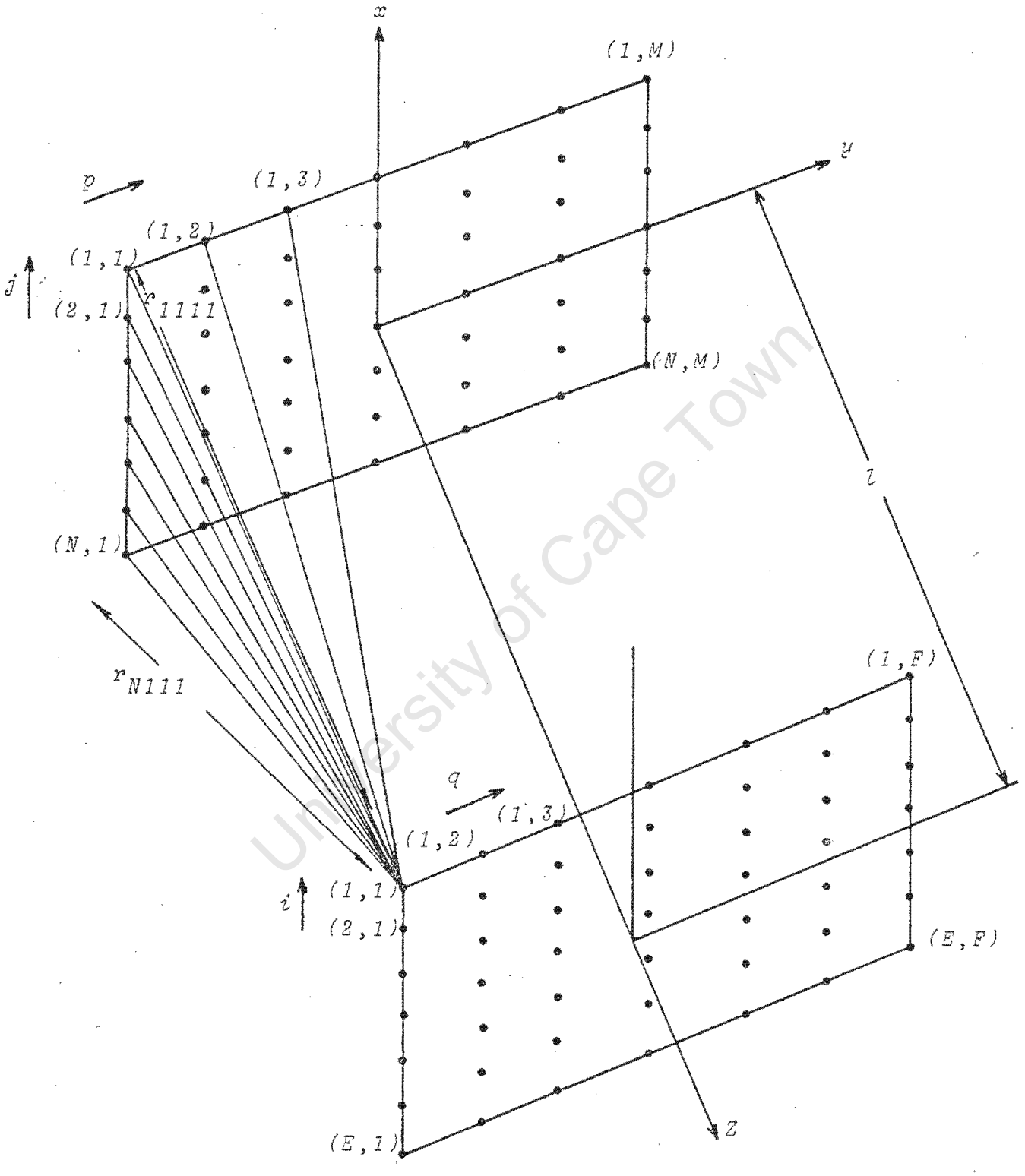


Figure 5.2: Calculation of pressure field from a planar array.

From the diagram, an expression for the pressure a distance ℓ from plane A can be developed:

$$P_{iq} = \sum_{j=1}^{N} \sum_{p=1}^{M} \frac{A_{jp}\, e^{j\omega_o t}\, e^{jkr_{jp}}}{r_{jp}} \qquad (5.3)$$

Again, assuming $A_{jp} = 1$, and by neglecting the time factor, this can be simplified:

$$P_{iq} = \sum_{j=1}^{N} \sum_{p=1}^{M} \frac{e^{jk\, r_{jp}}}{r_{jp}} \qquad (5.4)$$

A source that produces both plane waves and a two-dimensional pressure pattern was used in the simulations. This source is a planar array which is infinite in one dimension as shown in Figure 5.3. This source causes plane waves to propagate in the z-direction, and to be directed in a plane of constant y.

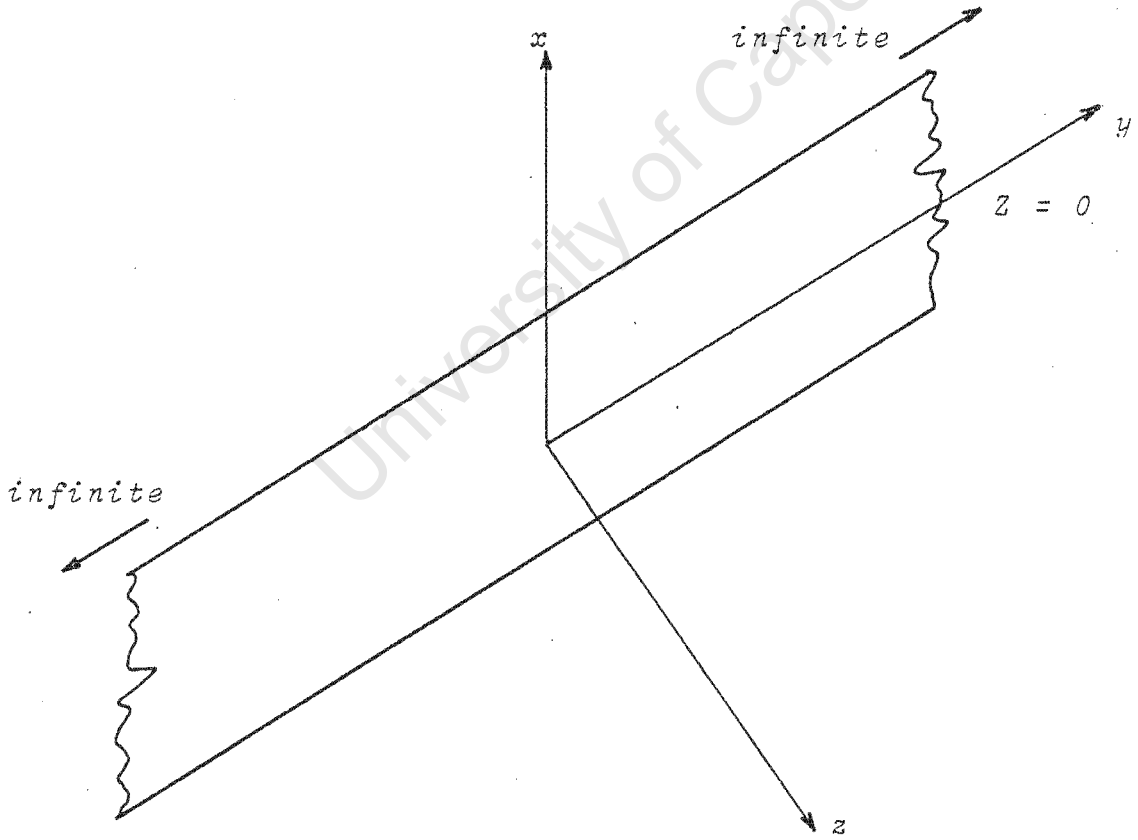


Figure 5.3: Infinite source in y-direction causes a two-dimensional pressure pattern for $z > 0$.

A word of caution on the choice of the source is relevant here. Initially, equation (5.2) was used to obtain the two-dimensional pressure at plane B from a line array located at plane A. The magnitude and phase of the pressure in the farfield was calculated by using both the integration and Fourier techniques. The magnitudes from the two techniques agreed- a Sin x/x pressure distribution was produced, which is characteristic of the farfield pressure pattern of a line array. The phase, however, was not the same. The integration technique produced a phase close to zero across the line, as expected in the farfield. The Fourier technique produced an extra 45° of phase across plane B. The phase error originates from the fundamental assumption that there have to be plane waves propagating in front of the source for the Fourier technique to yield correct results. When using a line array composed of many point sources, the majority of the plane waves propagate at angles greater than 90° and less than -90°.

It was noted from a paper by Freedman (21) that the extra 45° of phase results when the line array is extended to infinity. He presents an equation which relates the pressure of a rectangular array to velocity. This equation is such that the length of the sides of the rectangular source can be varied. The resultant phase at a field point can be calculated. His analysis shows that for a source which is infinite in the y-dimension, there will be an additional 45° of phase.

Thus for the integration and Fourier techniques to yield comparatable results, a planar array must be used in the simulated integration technique to give the extra 45° of phase.

5.3 Estimation of energy overlap from aliasing and repeated sources

Errors introduced by the Fourier transform technique, as well as methods of minimising these errors, are investigated in this section. Equations (4.5) and (4.10) are used to find quantitive estimates for the errors. These equations are manipulated into a form suitable for computation of the energy overlap ϵ_1 and energy overlap ϵ_2.

Firstly, the energy overlap ϵ_1, due to repeated sources is determined.

Substituting Eq. (4.5) into Eq. (4.9) to eliminate G gives:

$$N_p = \frac{2a}{S} \left(1 + \frac{L\lambda}{2\pi^2 a^2 \varepsilon_1} \right) \tag{5.5}$$

Therefore

$$\left(\frac{SN_p}{2a} - 1 \right) = \frac{L\lambda}{2\pi^2 a^2 \varepsilon_1}$$

Therefore

$$\varepsilon_1 = \frac{L\lambda}{2\pi^2 a^2} \left(\frac{2a}{SN_p - 2a} \right) \tag{5.6}$$

If L and a are in wavelengths, Eq. (5.6) reduces to

$$\varepsilon_1 = \frac{L}{2\pi^2 a^2} \left(\frac{2a}{SN_p - 2a} \right) \tag{5.7}$$

Equation (4.10) gives the energy overlap due to aliasing

$$\varepsilon_2 = \frac{S}{\pi^2 a} \tag{5.8}$$

Thus with knowledge of the number of points, Eq. (5.7) can be used to calculate the energy overlap due to the repeated sources. Equation (5.8) can then be used to calculate the energy overlap due to aliasing.

For the simulations to be performed in the following sections,

$a = 5\lambda$

$N_p = 128$

$S = 0.5\lambda$

Thus using Eq. (5.7)

$$\varepsilon_1 = \frac{L}{10\pi^2} * \frac{2}{(64 - 10)}$$

$$= \frac{2L}{10\pi^2 54}$$

Therefore

$$\varepsilon_1 = 0.038L\%$$

If ε_2 is calculated in a similar way from Eq. (5.8):

$$\varepsilon_2 = \frac{S}{\pi^2 a}$$

$$= 0.01 = 1\%$$

ε_2 is independent of distance L. This is because the energy overlap due to aliasing is not dependant upon the distance from the source, but only on the sampling rate.

5.4 Rearrangement of input data for FFT algorithm

When the FFT algorithm is implemented on a computer, it transforms a wave-field which lies in the first quadrant into a spectrum which is also in the first quadrant. This is shown in Figure 5.4(a). The wave fields used in the Fourier transform projection technique are centered on the propagation axis to take full advantage of symmetry, as seen in Figure 5.4(b).

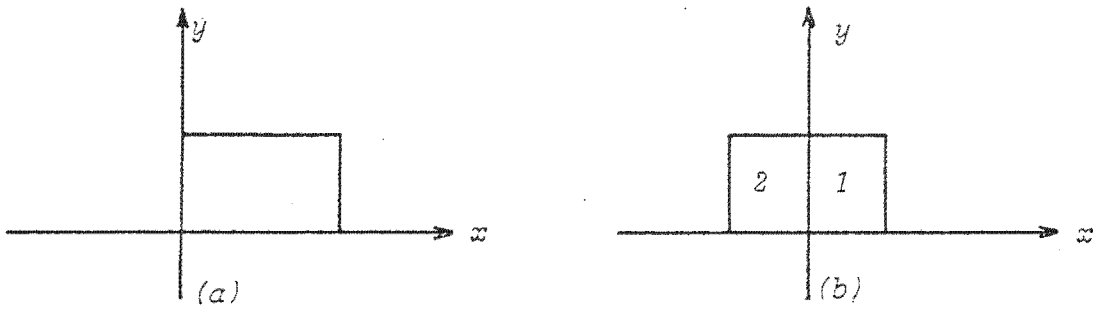


Figure 5.4: (a) Geometric arrangement of input data required by FFT algorithm

(b) Geometric arrangement of input data desired for diffraction problems.

In order to apply the usual FFT to the centered wave without getting a linear phase shift that accompanies a translation in the transform domain, a data swop (22) is used. This swop applied both before and after taking any transform or inverse transform, allows one to work with waves centered on the axis and also with FFT routines that work only in the positive half of the graph. Data in section 1 is swopped with data in section 2 in Figure 5.4(b). This gives the input data the form required by the FFT. In the program given for a two-dimensional simulation in listing C.4, however, only a swop before the first FFT and one after the second FFT is required. This is because the intermediate swops cancel.

5.5. Simulation of a two-dimensional rectangular aperture giving a two-dimensional pressure distribution

The magnitude in the graphs that follow are normalized with respect to the highest value for the graph. The source used in the simulations was a rectangular aperture of width 10λ, and infinite in the other dimension. Figures 5.5(a) and 5.5(b) compare the Fourier technique with the integration technique for distances between source and output planes of 10λ and 50λ respectively.

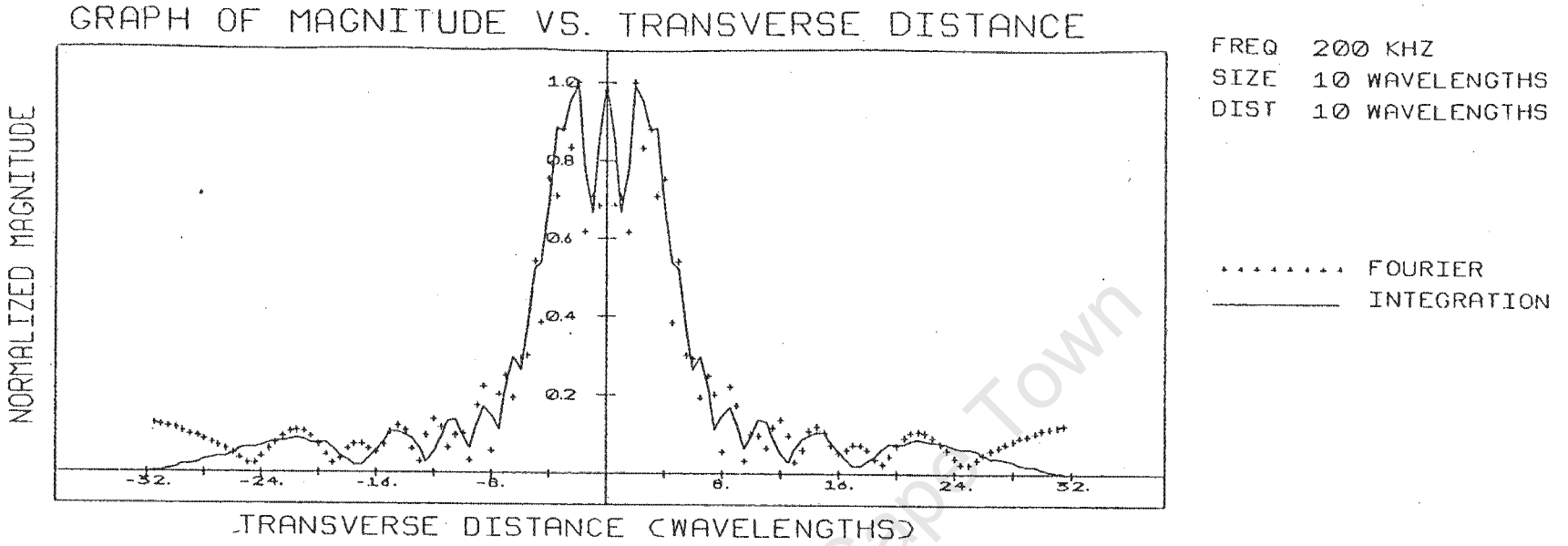


Figure 5.5(a): Prediction of pressure field at 10 wavelengths from the source with no shading

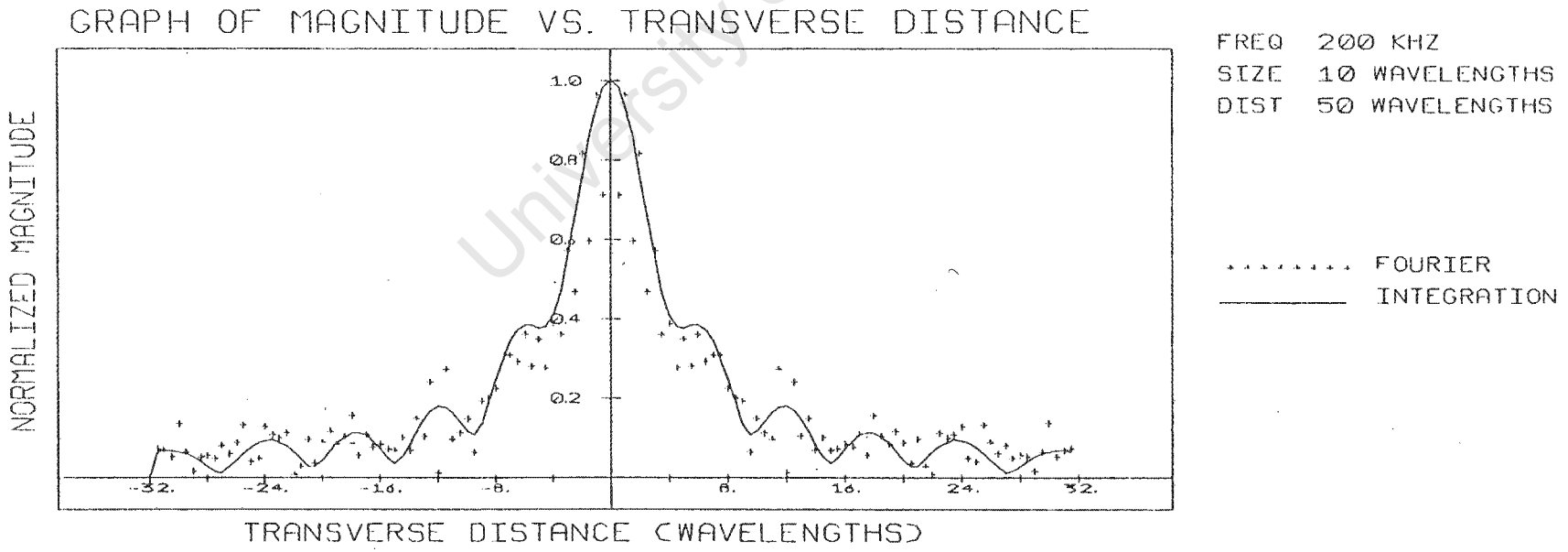


Figure 5.5(b): Prediction of pressure field at 50 wavelengths from the source with no shading.

Even close to the source, a discrepency between the two techniques was found. This was attributed to the truncation of the spatial frequency spectrum, which was performed before the magnitude had decreased to an acceptable level. The spectrum for the rectangular source is shown in Figure 5.6, plotted as magnitude vs. directional cosine.

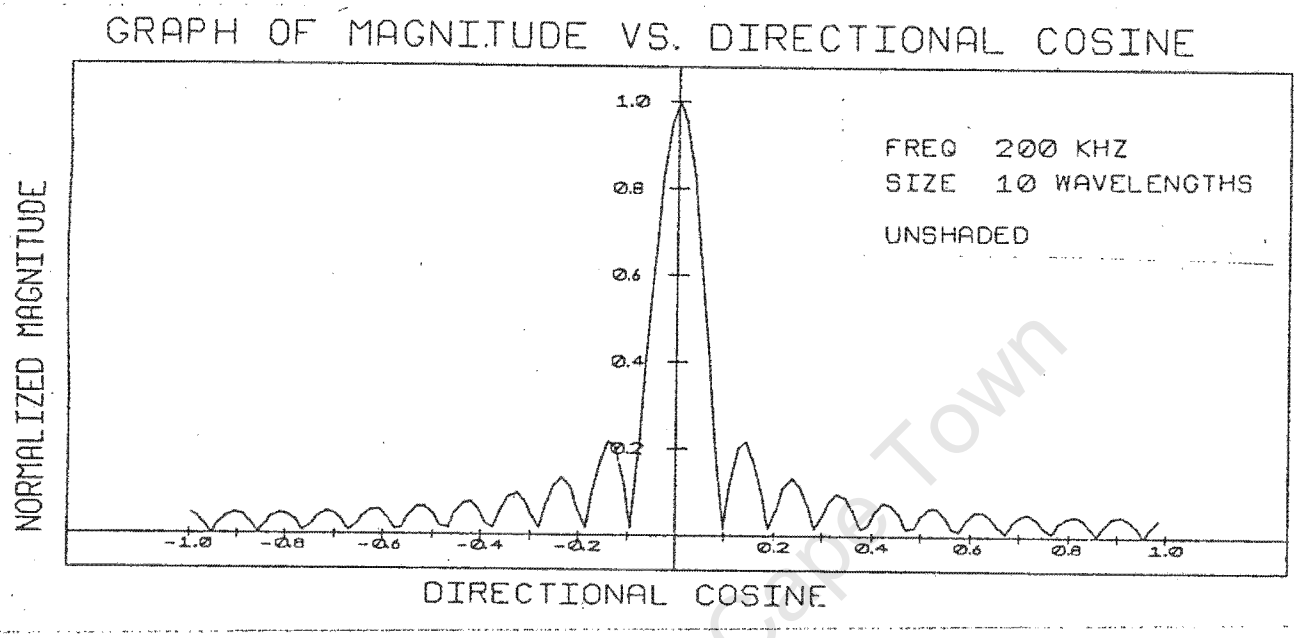


Figure 5.6: Fourier transform of rectangular source, showing unacceptable sidelobe levels at the edges of the bandwidth.

Inspection of Figure 5.6 confirms that for a directional cosine of 1 the magnitude has not yet fallen to an acceptable level. The reason for this is that the rectangular source aperture chosen has sharp edges, resulting in high frequency components. There are two methods of solving this problem:

(i) Increase the bandwidth until the high frequency components have reduced to an acceptable level.

(ii) Modify the source such that the magnitude of the frequency components has reduced to an acceptable level at the edge of the bandwidth.

Method (i) is not practical, because of the physical constraint that the plane waves can only propagate in front of the source.

The second option is implemented by superimposing an amplitude window on the source. The window chosen is shown in Figure 5.7.

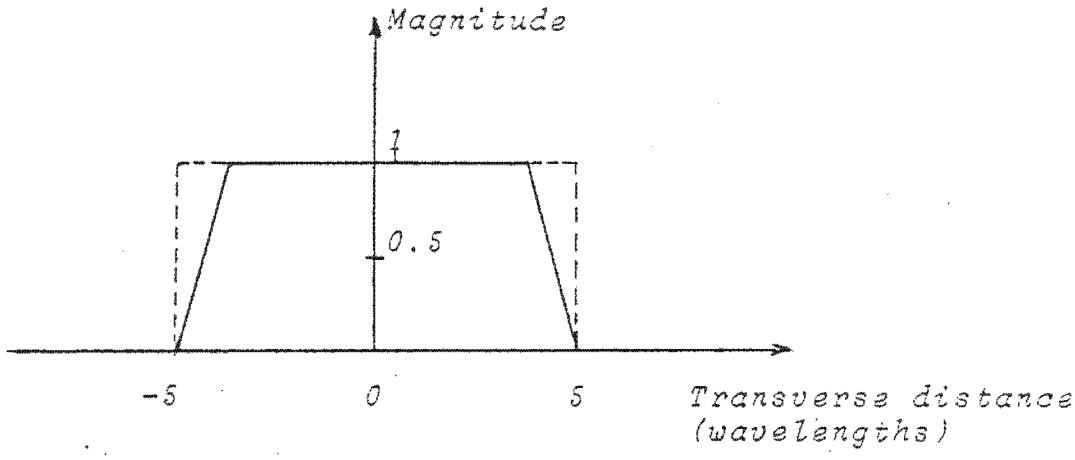


Figure 5.7: Source with amplitude window.

The rectangular function, represented in Figure 5.7 by the dotted lines, is multiplied by the amplitude window, represented by the solid line. The window is similar to the source, except that the leading and trailing edges are tapered. The addition of this window decreases the sidelobes to an acceptable level, as shown in Figure 5.8.

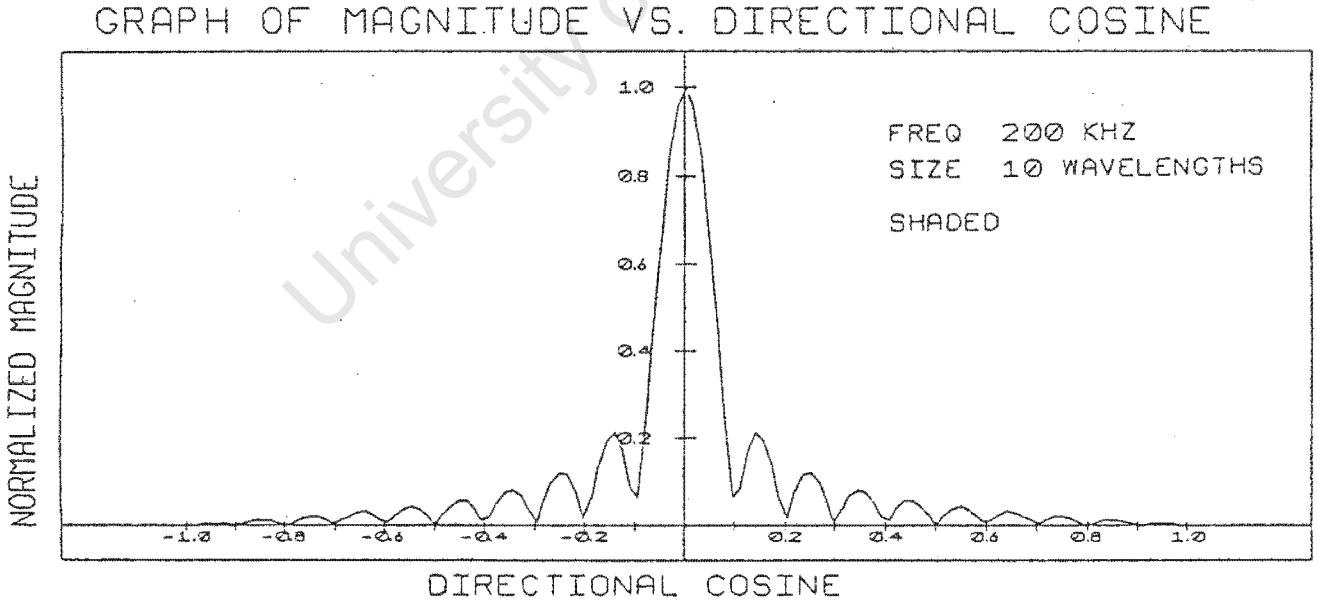


Figure 5.8: Fourier transform of windowed source function, showing acceptable sidelobe levels at the edges of the bandwidth.

Figures 5.9 to 5.12 show the pressure patterns at varying distances from the windowed source.

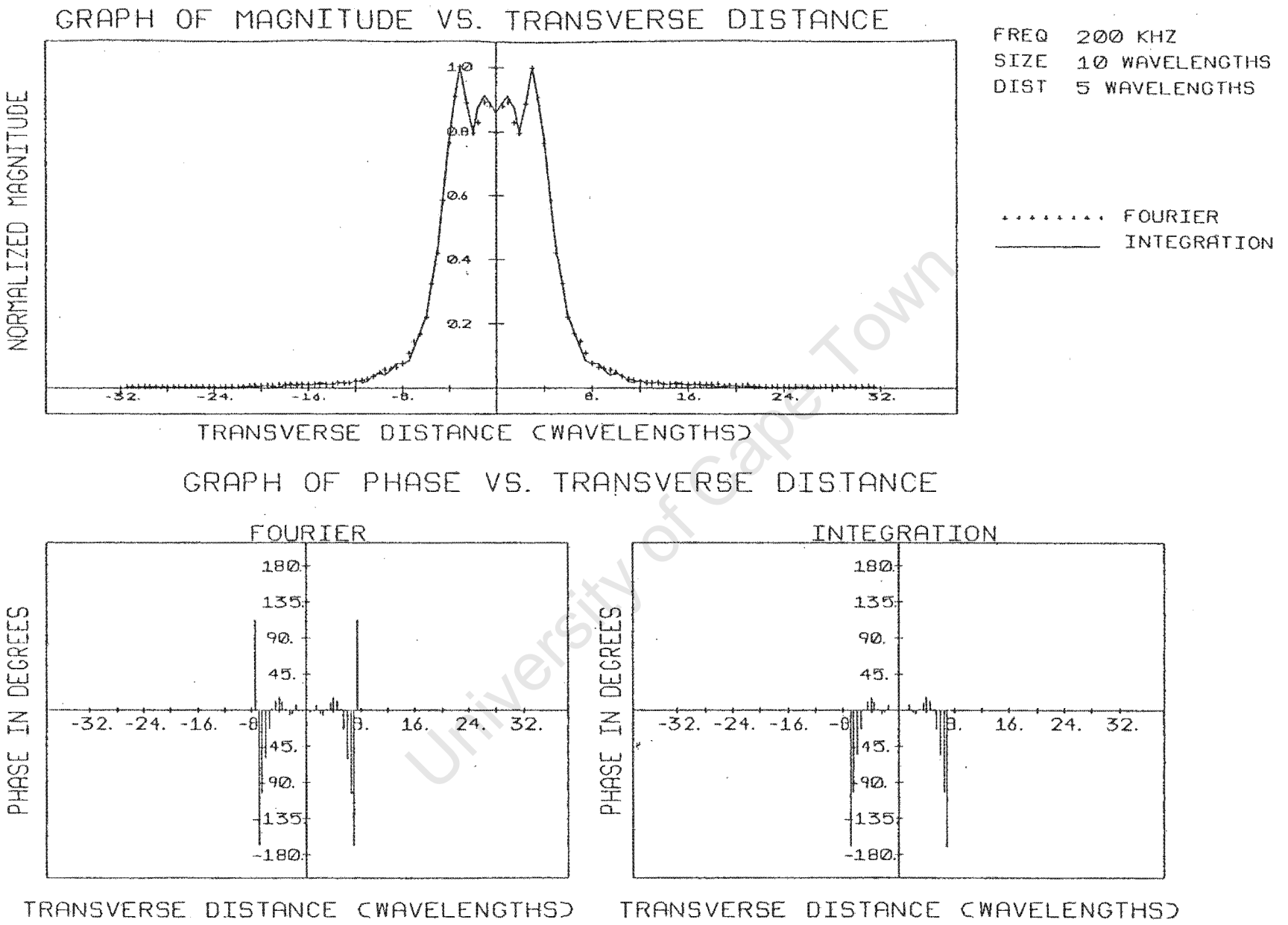


Figure 5.9: Magnitude and phase of projected pressure at 5λ from the source.

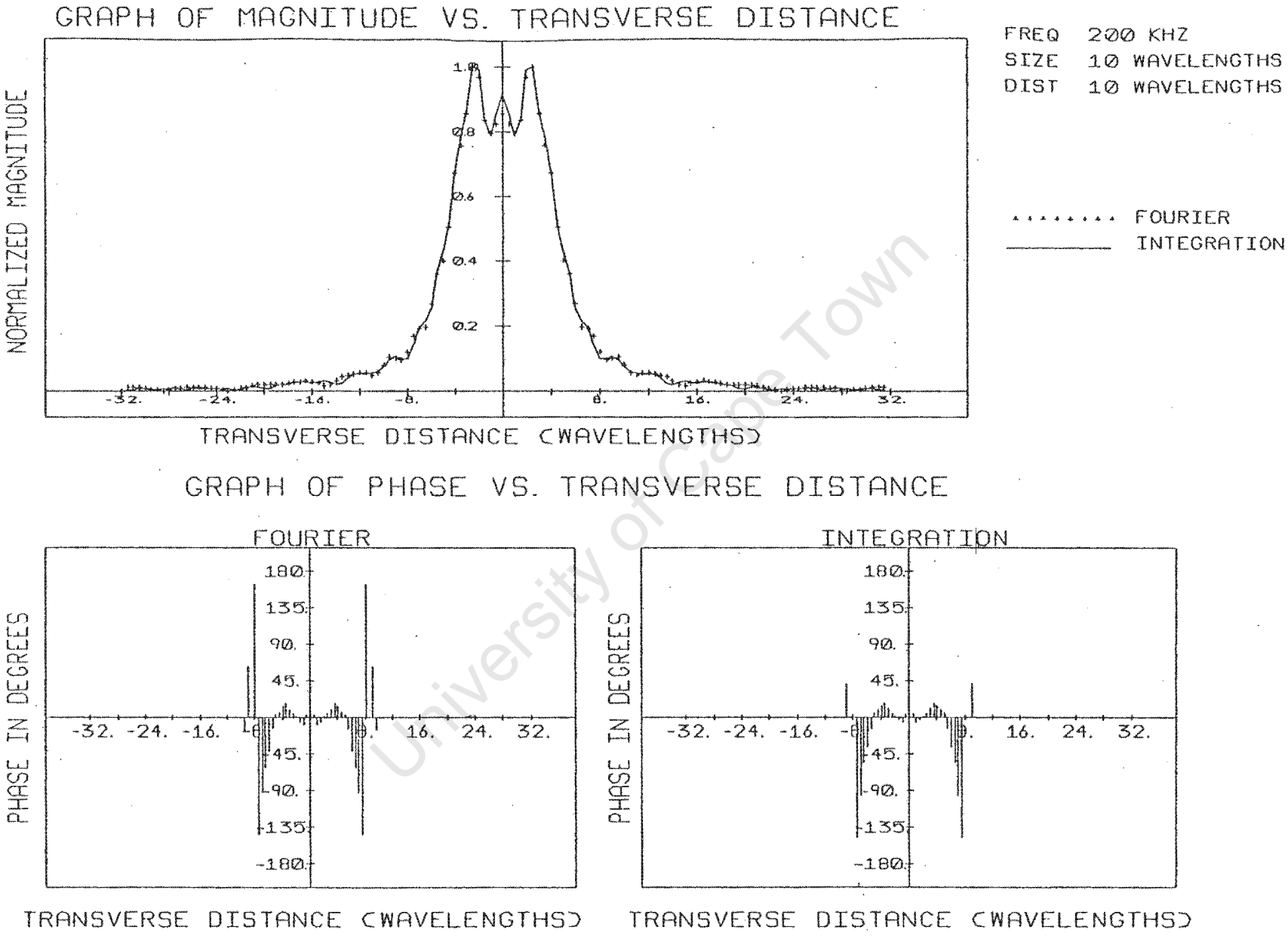


Figure 5.10: Magnitude and phase of projected pressure at 10λ from the source.

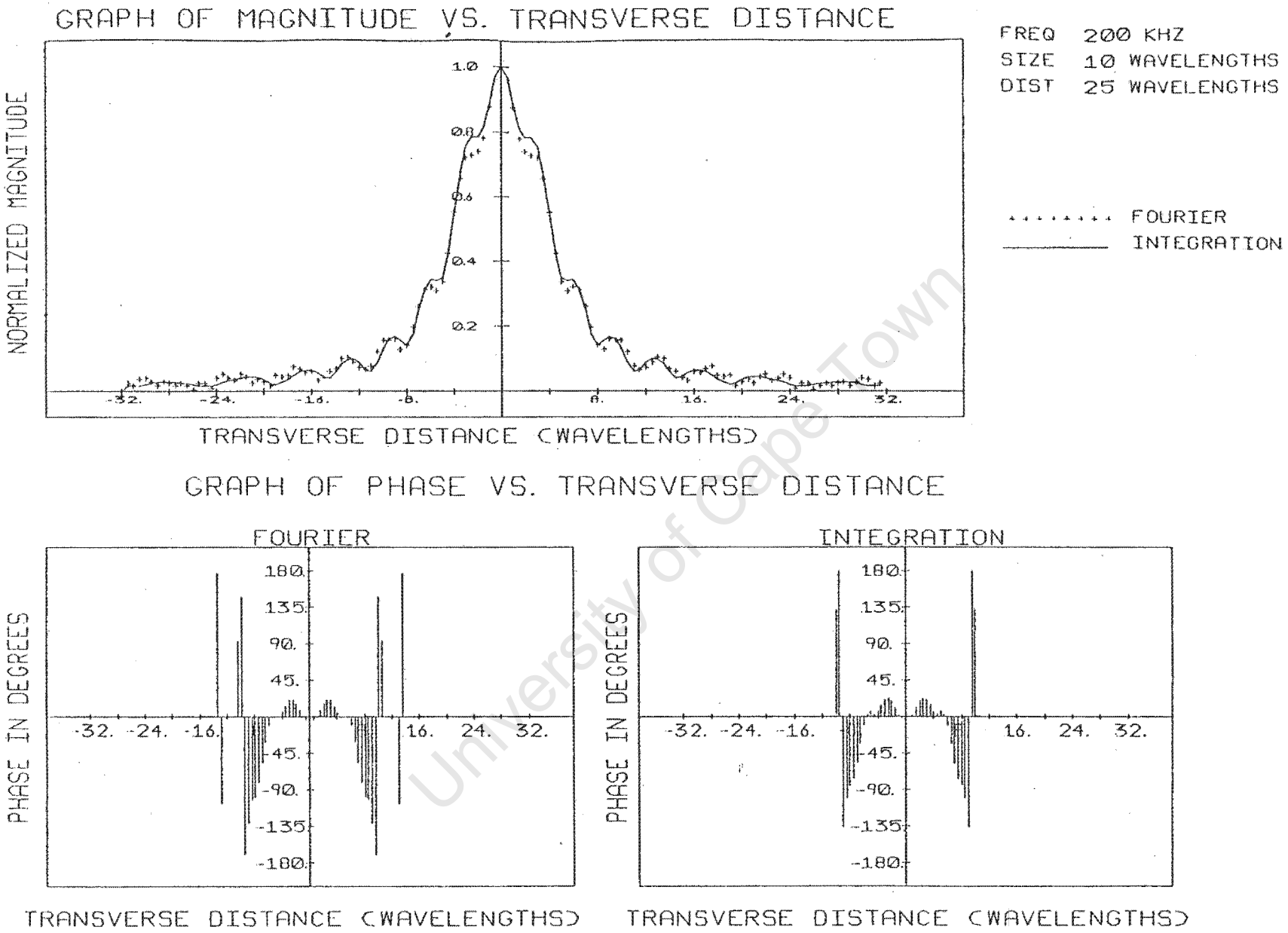


Figure 5.11: Magnitude and phase of projected pressure at 25λ from the source.

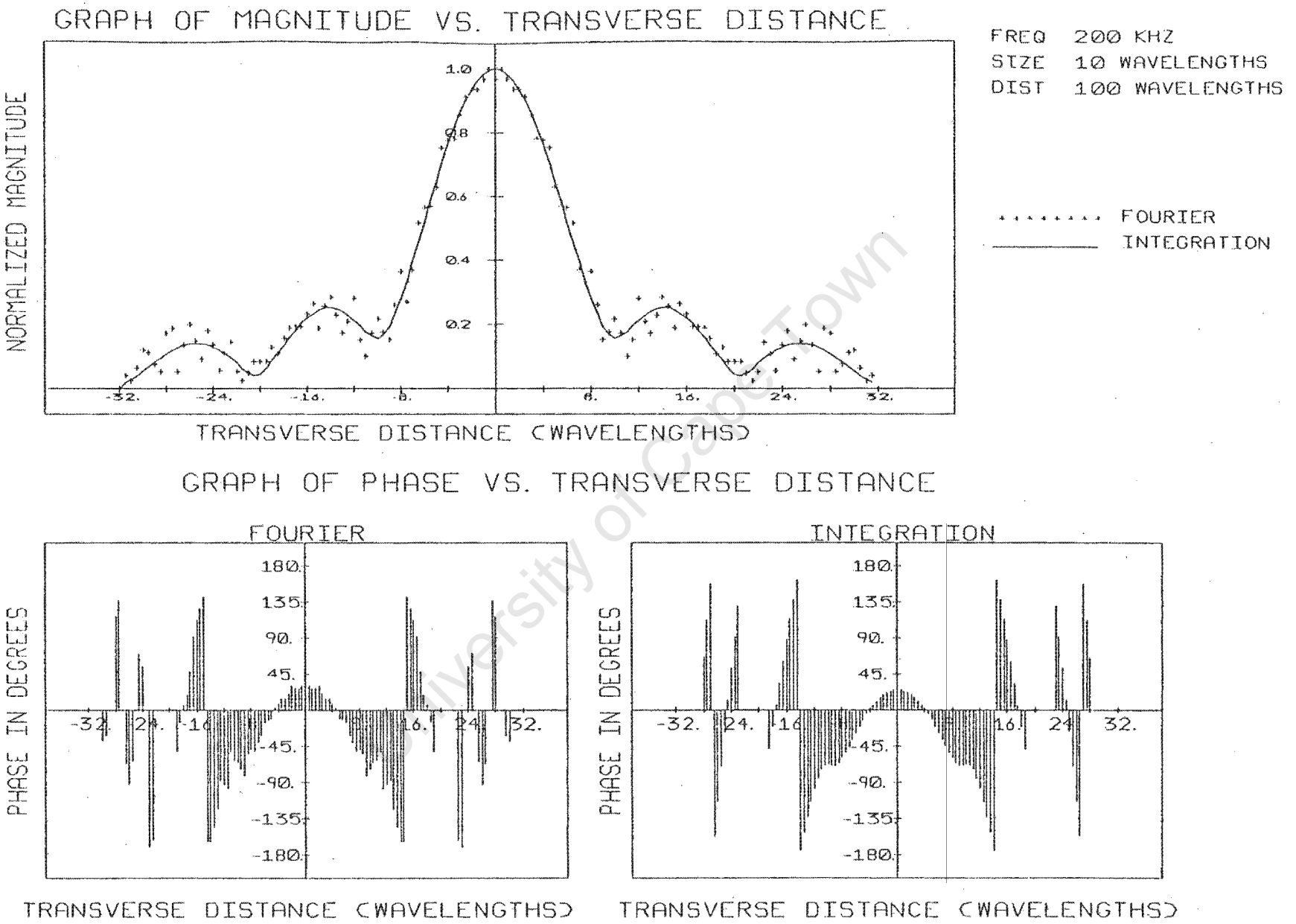


Figure 5.12: Magnitude and phase of projected pressure at 100λ from the source.

Figures 5.9 and 5.10 illustrate that there is satisfactory agreement between the integrated and Fourier techniques for both magnitude and phase. The two techniques do not correspond as well, however, in Figures 5.11 and 5.12. This can be explained in terms of the equation for energy overlap:

For $L = 25\lambda$, $\varepsilon_1 = 0.038 \times 25 = 0.95\%$
For $L = 100\lambda$, $\varepsilon_1 = 3.8\%$,

This energy overlap, together with that due to aliasing (1%), is too high. For large projection distances, the major error is due to the repeated sources being too close. The solution to the problem is to use more points, thereby increasing the guard band region.

Figures 5.13 and 5.14 show the effects of having insufficient auxiliary zeros. In these simulations, 64 points ($N_p = 64$) were used for the Fourier technique. There are two limitations in using fewer points. The first is that the beamwidth is limited by the number of points used. Figure 5.14 shows that the beam has been truncated at 16 wavelengths from the centre of the transducer. The second limitation is that the energy overlap between repeated sources increases as the number of points is decreased. The energy overlap ($\varepsilon_1 = 4.6\%$), using 64 points, for a distance of 50λ from the source is greater than when 128 points are used at a distance of 100λ from the source ($\varepsilon_1 = 3.8\%$). The errors introduced by using 32 points are unacceptably high , as shown in Figures 5.15 and 5.16.

The preceding discussion shows that the beamwidth and energy tolerances must be defined before the Fourier technique can be used to project fields from one plane to another, so that sufficient points are used to prevent errors.

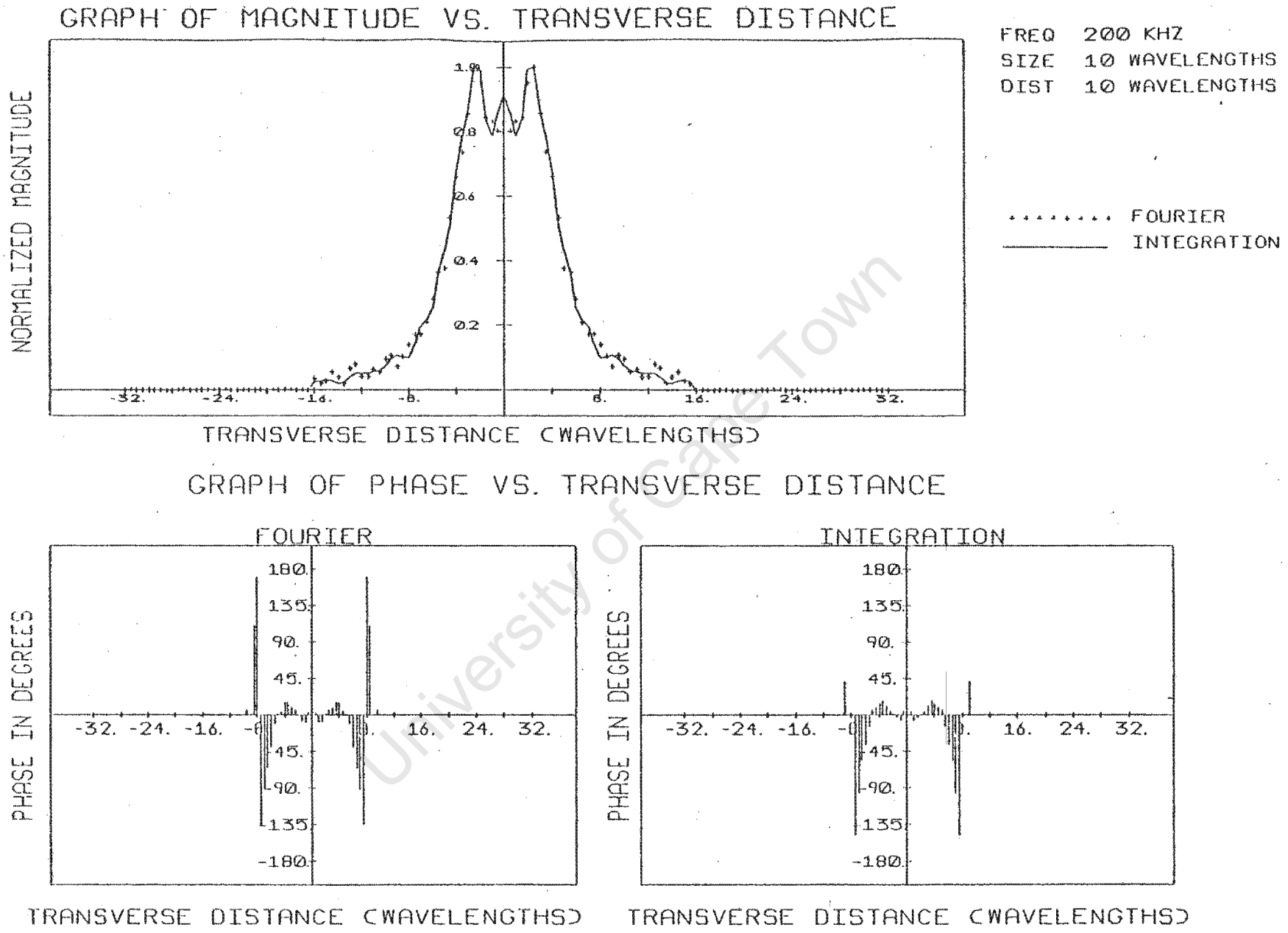


Figure 5.13: Magnitude and phase of projected pressure using 64 points at a distance of 10λ from the source.

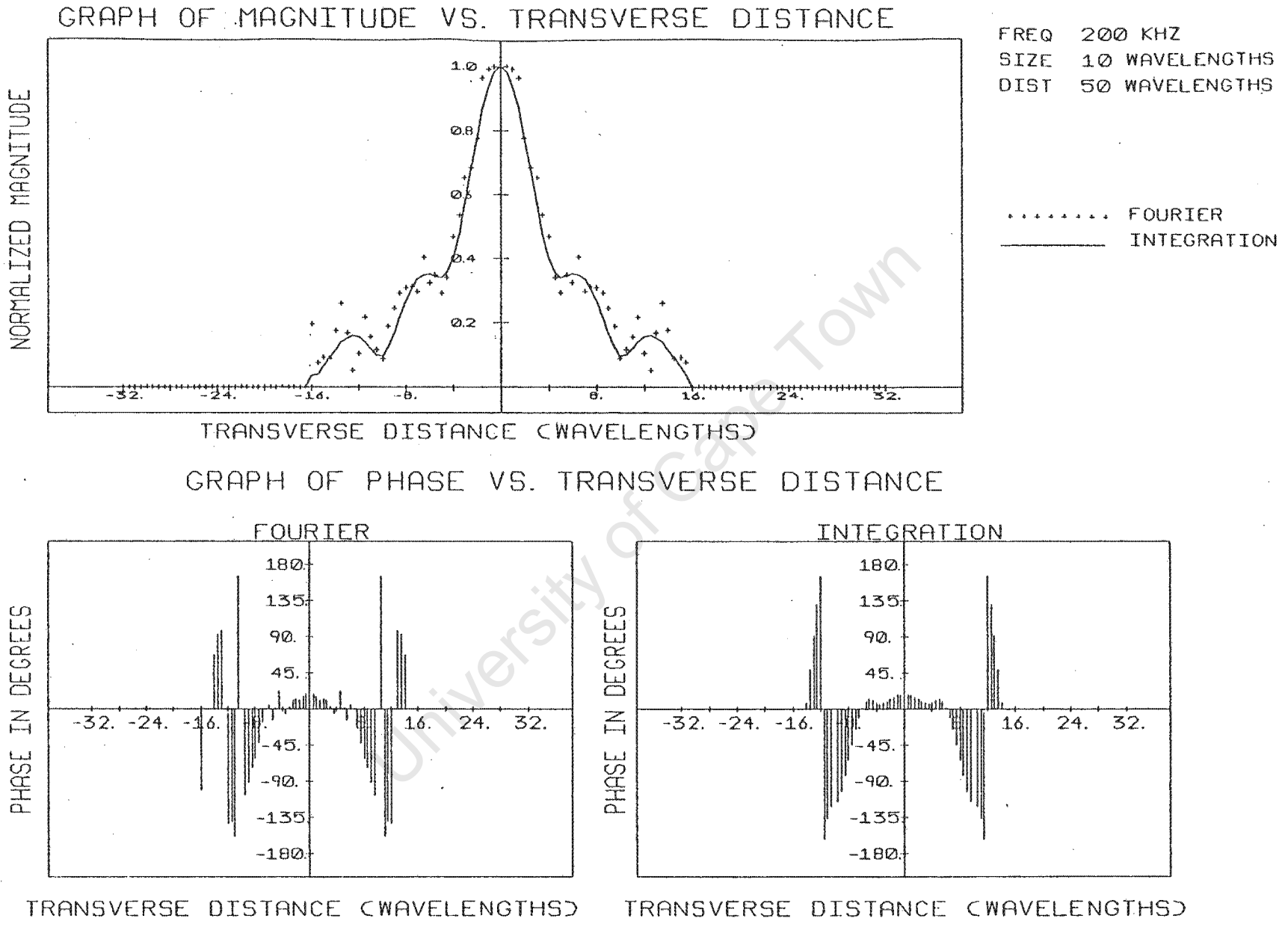


Figure 5.14: Magnitude and phase of projected pressure using 64 points at a distance of 50λ from the source.

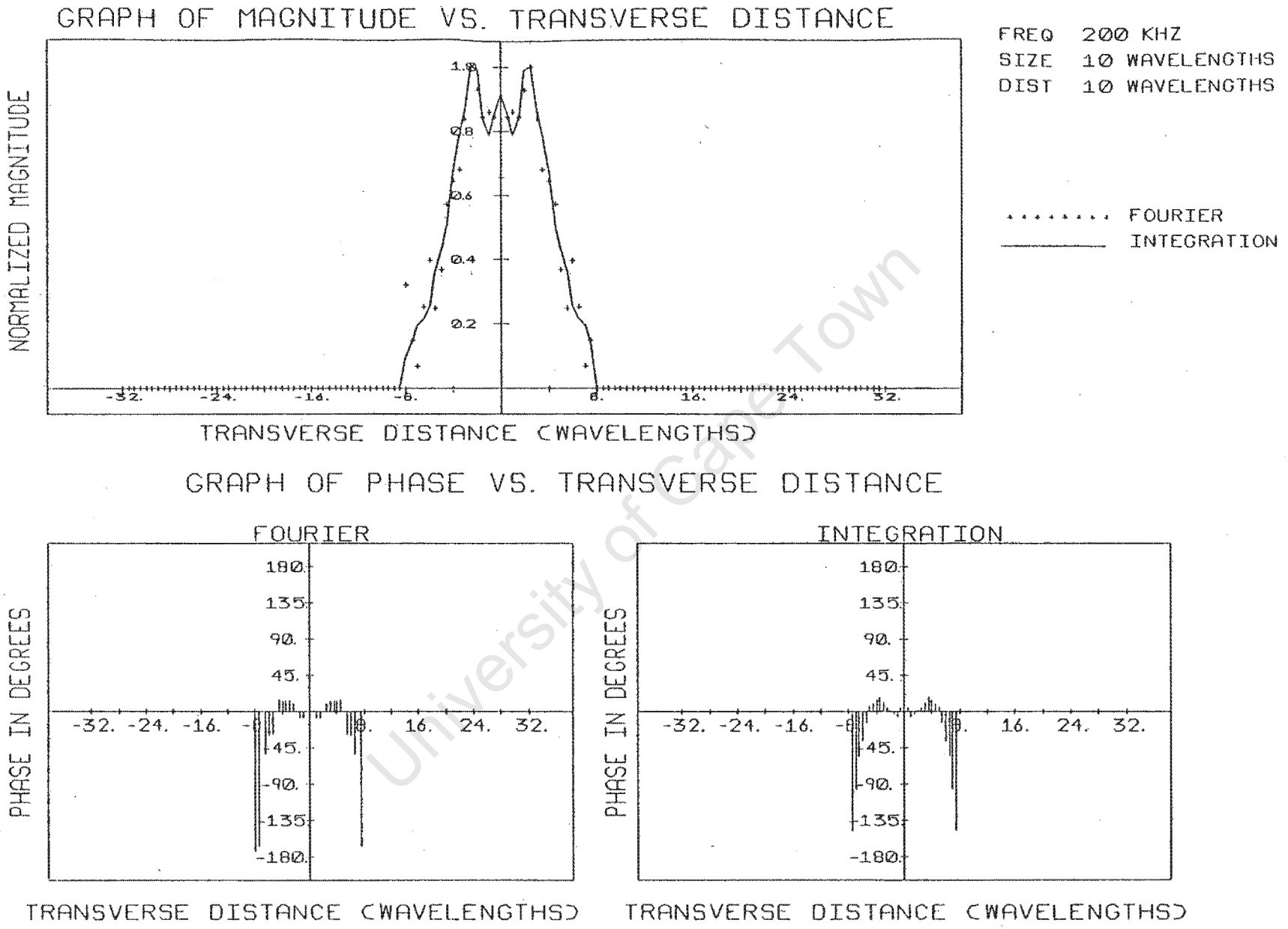


Figure 5.15: Magnitude and phase of projected pressure using 32 points at a distance of 10λ from the source.

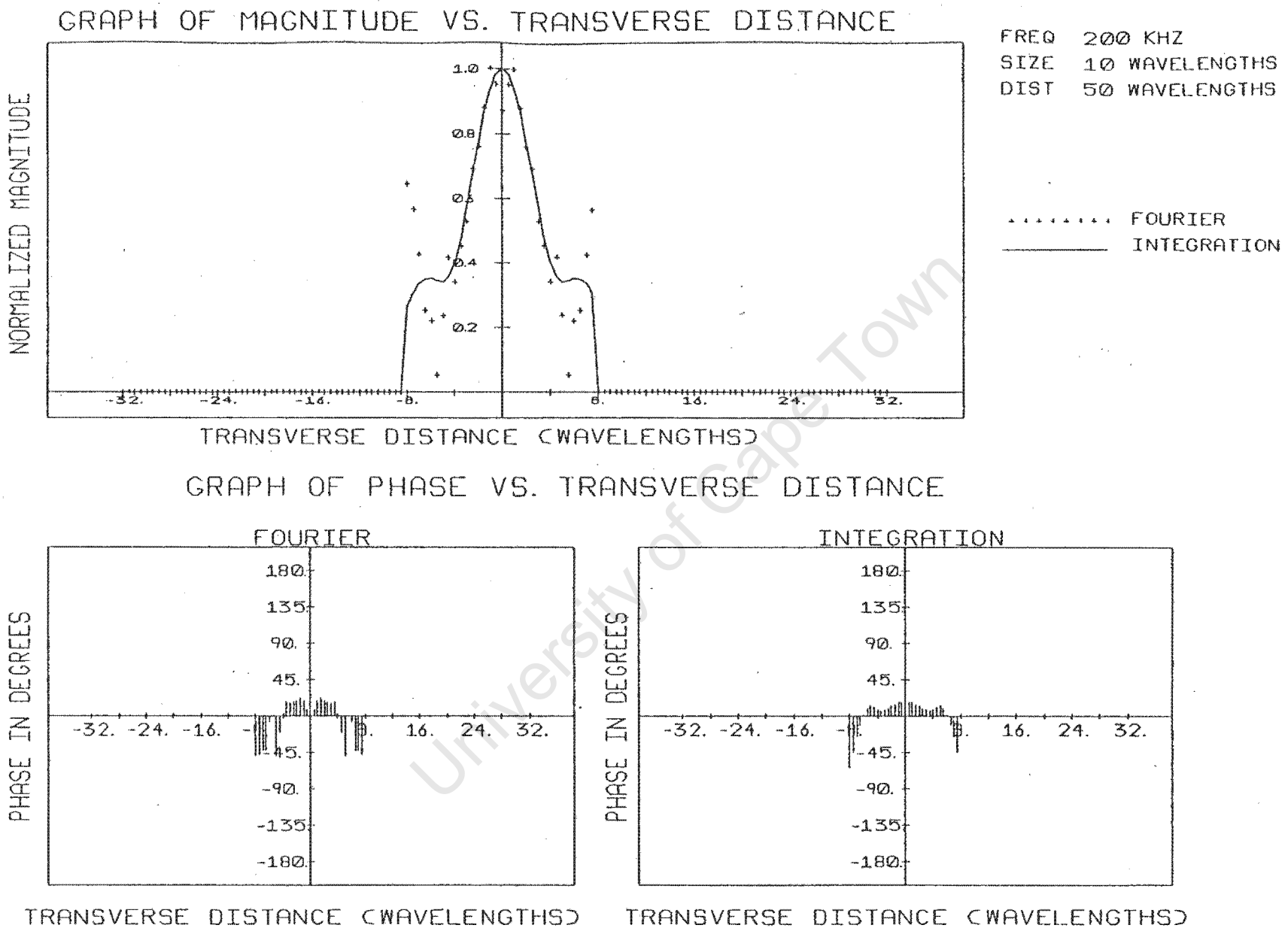


Figure 5.16: Magnitude and phase of projected pressure using 32 points at a distance of 50λ from the source.

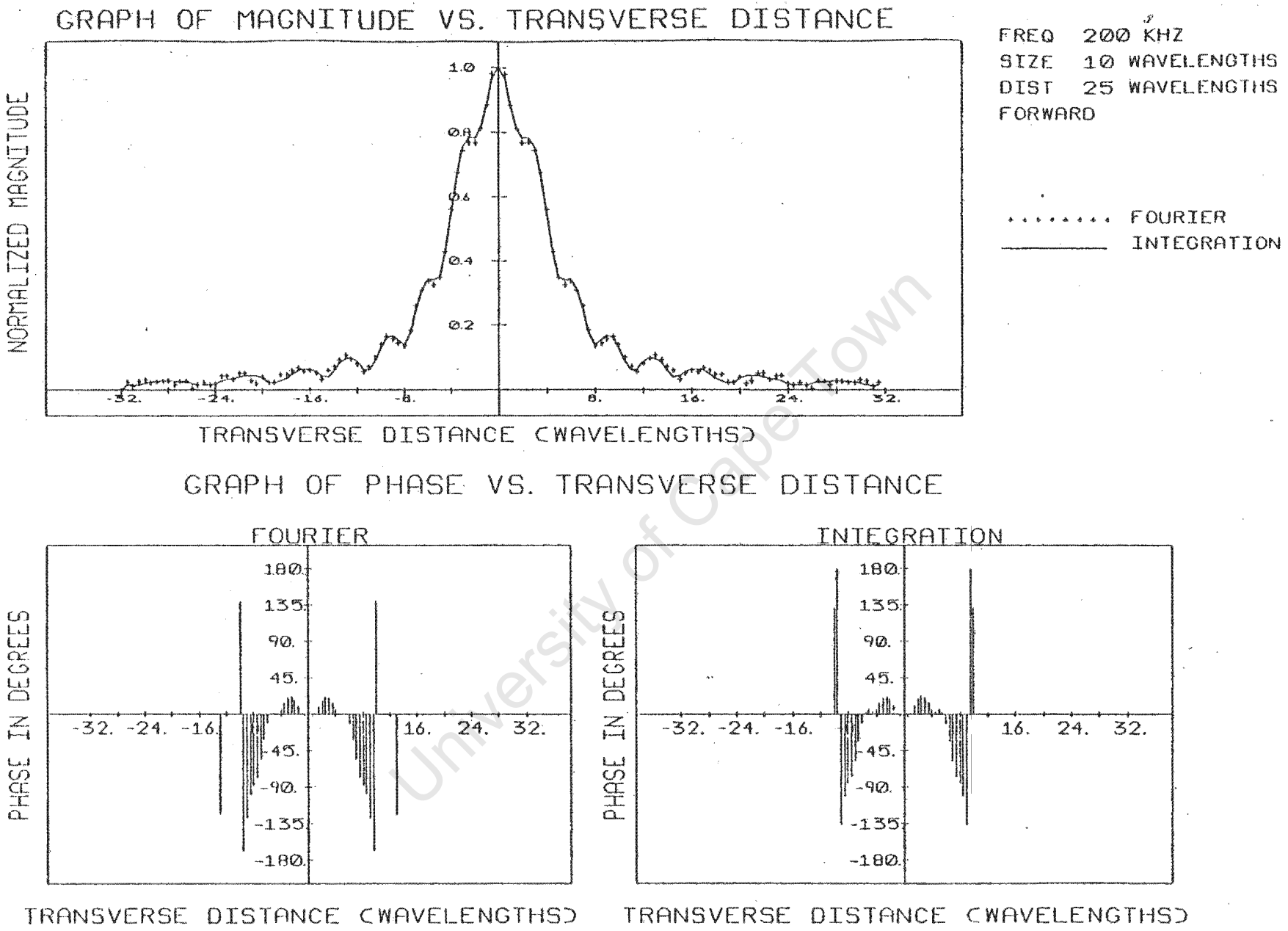


Figure 5.17: Comparison of integrated pressure at 25λ from the source, and the pressure obtained from forward projecting the integrated pressure from 10λ to 25λ using the Fourier technique.

The back projection technique was used in a similar way. The pressure distribution was calculated at 50λ from the source using the integration technique. This pressure data was used by the Fourier technique to find the pressure field at 25λ from the source (l=-25λ). This data was compared to the integrated pressure calculated for l = 25λ. It can be seen that these pressures do not correspond as well as those in Figure 5.17. This is as a result of the larger projection distance (15λ in Figure 5.17 and 25λ in Figure 5.18), which caused the phase transformation factor to change too rapidly for the sampling rate chosen. Graphs of the real part of this factor against directional cosine are shown in Figures 5.19(a), (b) and (c) for l=5λ, l=15λ and l=25λ respectively.

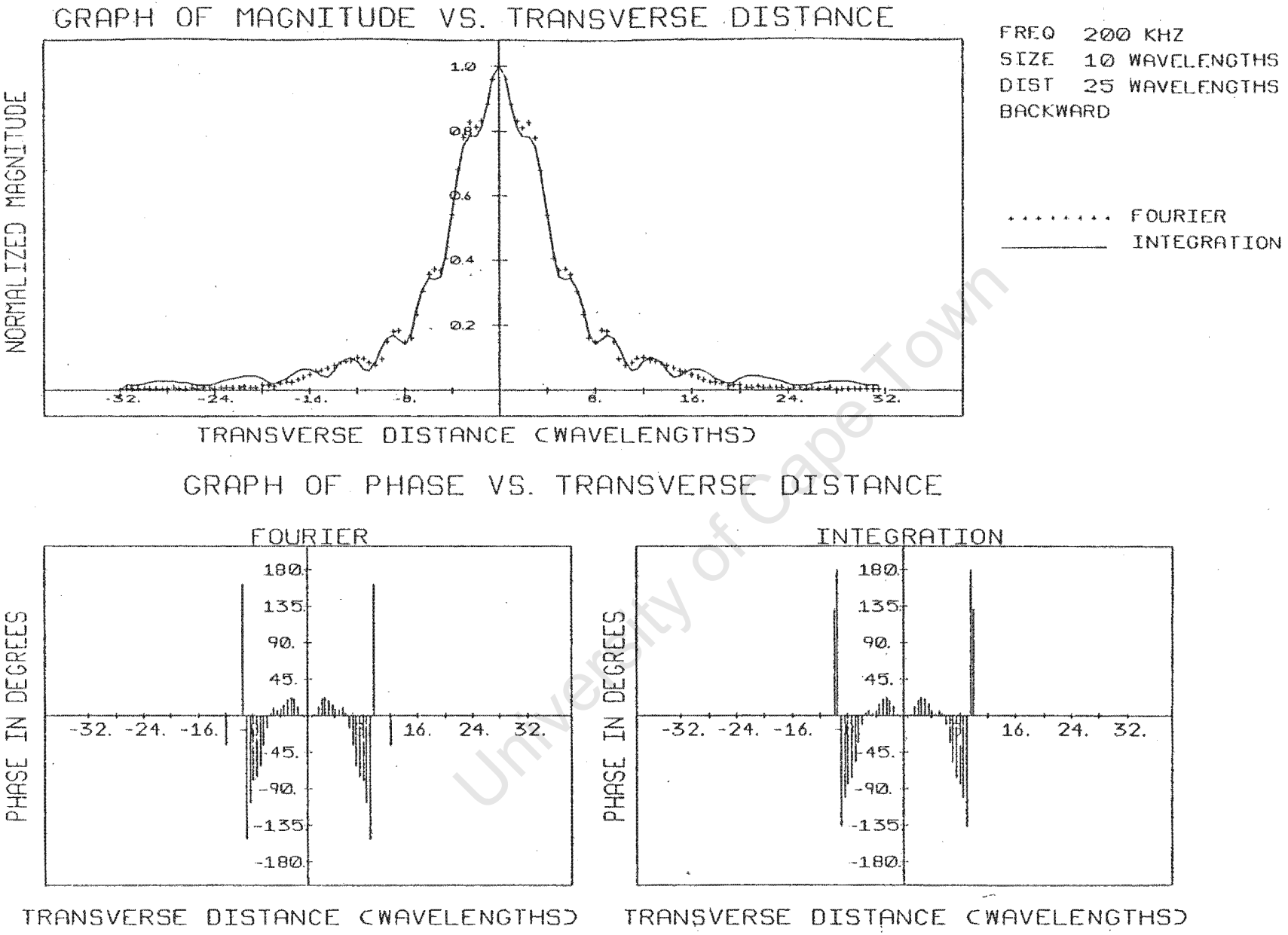


Figure 5.18: Comparison of integrated pressure at 25λ from the source, and the pressure obtained from backward projecting the integrated pressure from 50λ to 25λ using the Fourier technique.

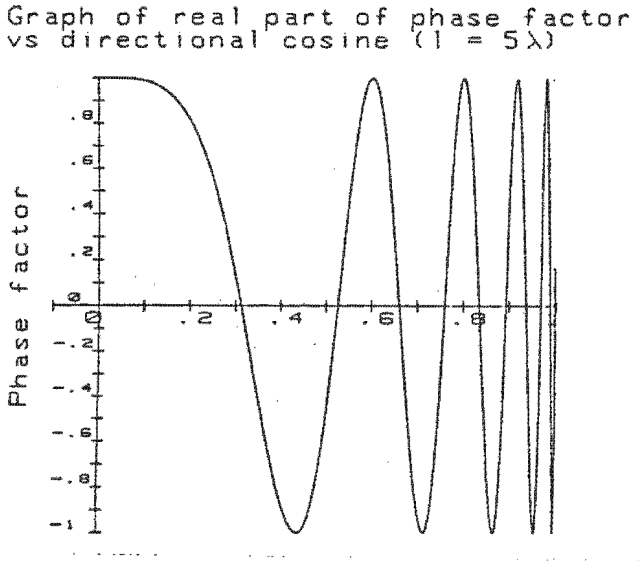


Figure 5.19(a): Phase factor for ℓ=5λ.

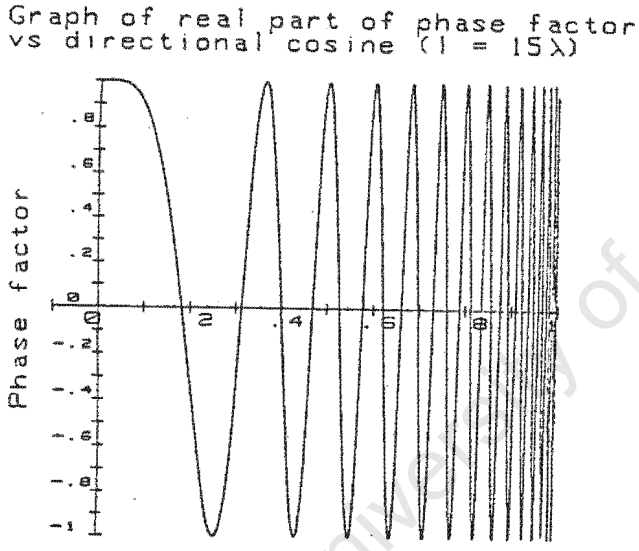


Figure 5.19(b): Phase factor for ℓ=15λ.

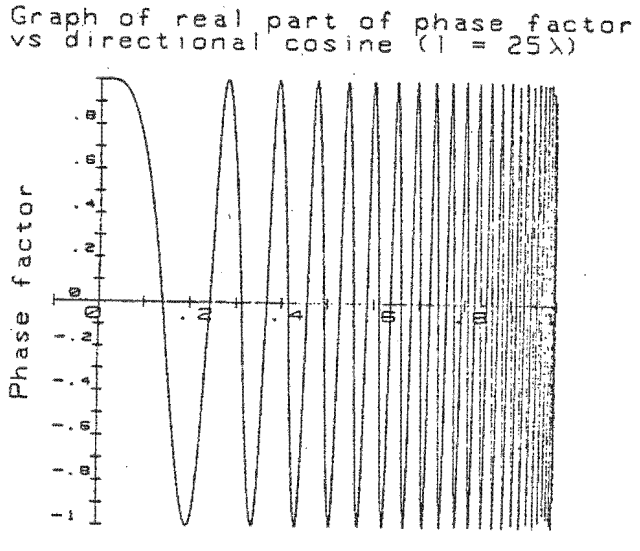


Figure 5.19(c): Phase factor for ℓ=25λ.

It can be seen from the graphs that the number of cycles is equal to the length, ℓ, in wavelengths. It can be seen that as the directional cosine approaches 1, the number of cycles increases. In the band 0.8 to 1, therefore, there are not enough points to satisfy the sampling requirement. Hence errors are introduced when a field is projected a large distance. This error can be eliminated by using more points to satisfy the Nyquist criterion.

Figure 5.20 shows the velocity field at the source, which has been back projected from a pressure field (calculated using the integration technique) at 25 wavelengths from the source. The characteristic Gibbs-type phenomenon can be seen on the plot of magnitude. This is, in part, caused by the finnite spatial bandwidth of the reconstruction process - ie. only those plane waves which satisfy the criterion $(k^2 - k_1^2)^{1/2} > 0$ were utilized. Another cause of the ripple is the undersampling of the phase factor. The phase in Figure 5.20 is as predicted (it should be zero across the source).

5.7. Determination of the farfield pressure pattern using a single Fourier transform.

The large distance between the near- and farfields is the major source of error when using the Fourier technique to project from one plane to the other. It is evident from Figure 5.12 that the larger the distance between the planes, the more auxiliary zeros, and hence more points are needed. If only the farfield pressure pattern is required, however, there is a short cut. In section 3.2 it was shown that the farfield pressure pattern was the Fourier transform of the source velocity. Figure 5.6 is therefore the farfield pressure pattern, with the horizontal axis interpreted as Θ, instead of the directional cosine. The farfield pressure pattern of a rectangular source is $\frac{\sin x}{x}$, as shown in the figure, which agrees with the theory of farfields.

In practice, the pressure pattern at a distance from the source is required to be measured. To prove that the farfield pressure pattern can be obtained directly from measurements of pressure in a plane of distance ℓ from the transducer, consider the block diagram given in Figure 5.21. This diagram gives the steps taken to back project a measured pressure field to a velocity field at the source.

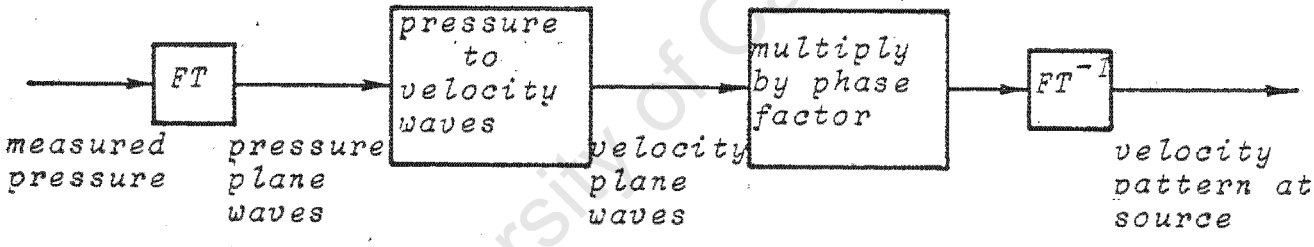


Figure 5.21: Block diagram showing back projection of the pressure field to obtain the velocity field at the source.

The last block which should be added to Figure 5.21 to obtain the farfield pressure is to take the Fourier transform of the velocity at the source.

The block diagram will then be:

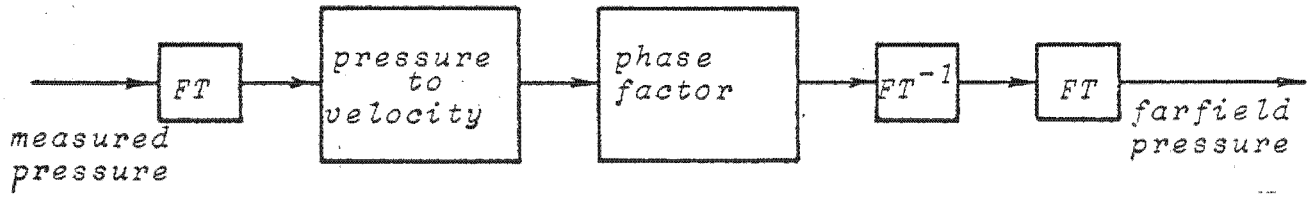


Figure 5.22: Block diagram showing the steps needed to find the farfield pressure from the measured pressure.

The above figures can be simplified, because the Fourier transform of the inverse Fourier transform of a field is the original field. Further, the phase factor can be neglected because in the farfield ony the magnitude is important. Thus by firstly taking the Fourier transform of the measured pressure, then by converting the pressure waves to velocity waves, the farfield pressure pattern can be obtained. Figure 5.23 shows the farfield pressure pattern obtained by applying the FFT to the data generated by the integration technique at a distance of 10λ from the source.

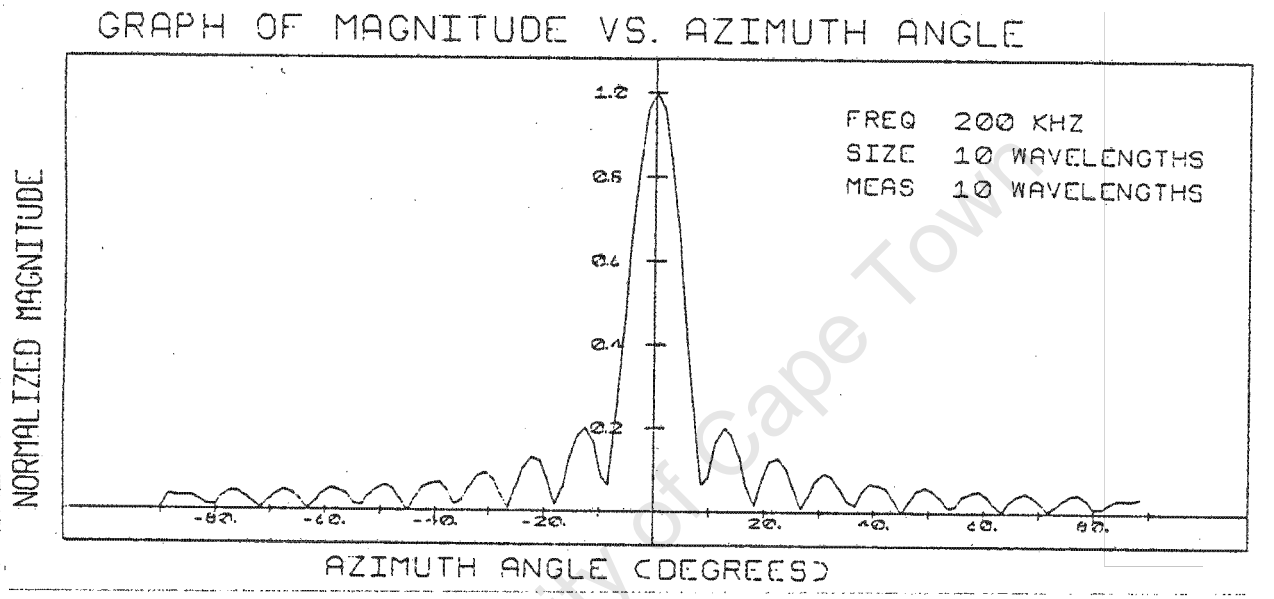


Figure 5.23: Farfield pressure obtained directly from nearfield pressure by performing a single FFT.

5.8. Back projection with element failure

The back projection technique can be used to check a line array for defects. By projecting the pressure measured a distance ℓ from the source back to the velocity field on the array, it is possible to determine where the defect is located.

The integration formula was used to find the pressure a distance of 10λ from the source with a defect at element number 3.0, as shown in Figure 5.24. As the array is infinite in the y-direction, the entire line at x=3 is not radiating.

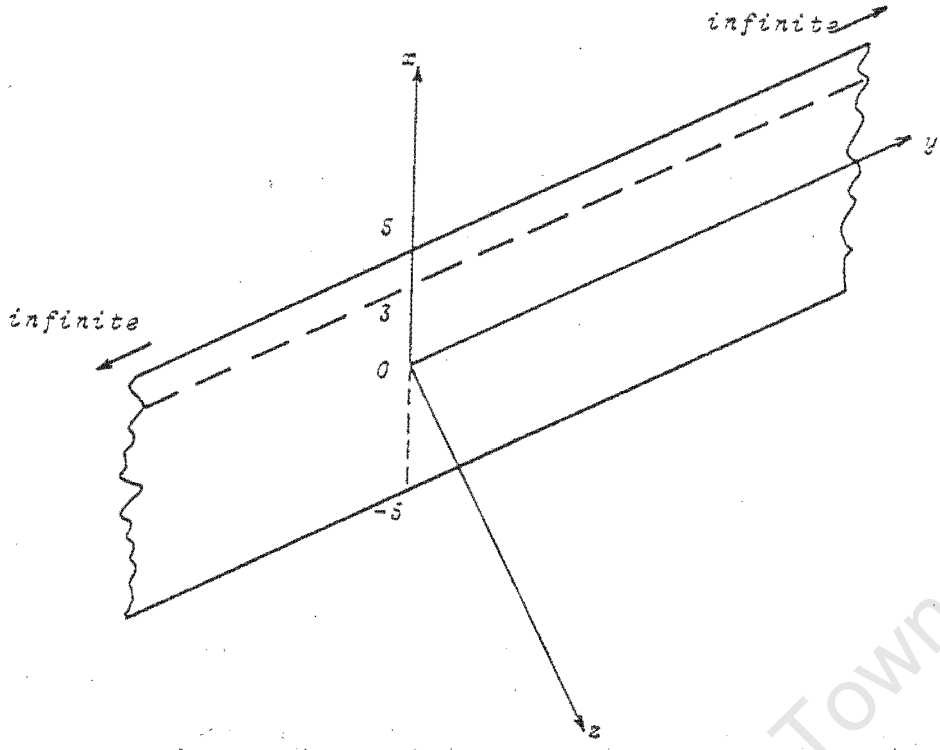


Figure 5.24: Showing the transducer with element failure at x=3.

Figure 5.25 gives the magnitude and phase of the back projected velocity distribution of the source.

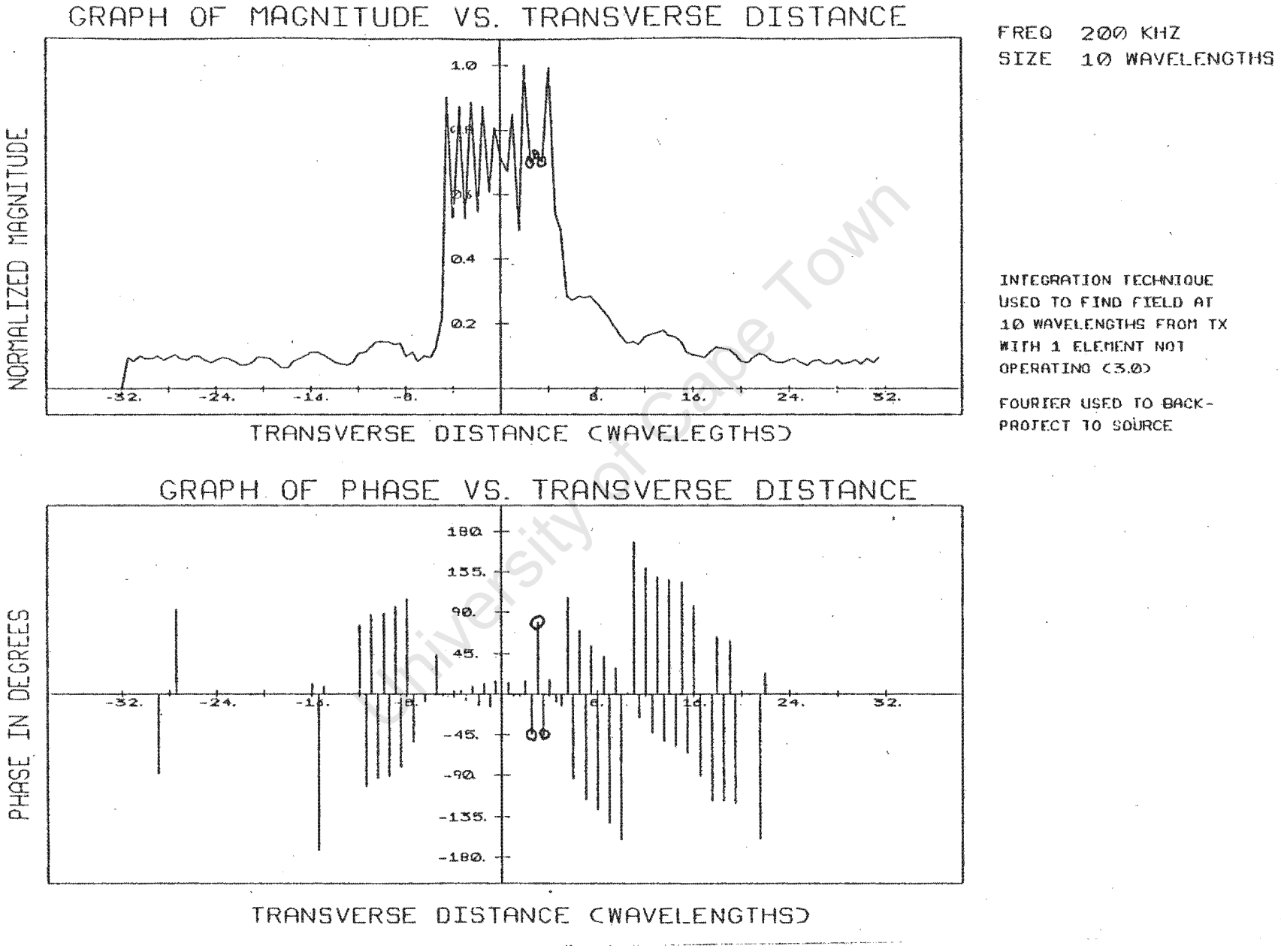


Figure 5.25: Detection of element failure using the Fourier back projection technique.

With no a priori information about the defect, the non radiating element can be located in the positive half of the velocity distribution, as shown in the figure. The phase plot shows that for the transverse distance equal to 2.5 to 3.5 (circles on the plots), there is possibly an error. The exact location of the defect, however, cannot be determined. The reason for this uncertainty is that the discontinuity caused by the null in the velocity results in the spatial frequency exceeding the bandwidth limitations. This causes a truncation in the spatial frequency domain, thus producing ripples (Gibbs-type phenomena) in the spatial domain.

The simulation of a line array containing 2 defects (positions 2.5 and 3.0) was simulated in the same way as for one defect. The defect is larger, hence it can be detected. The magnitude in Figure 5.26 shows clearly which elements are in failure, and the phase plot confirms this.

Thus it can be concluded that as long as the defect is not too small, it can be detected by the back projection technique.

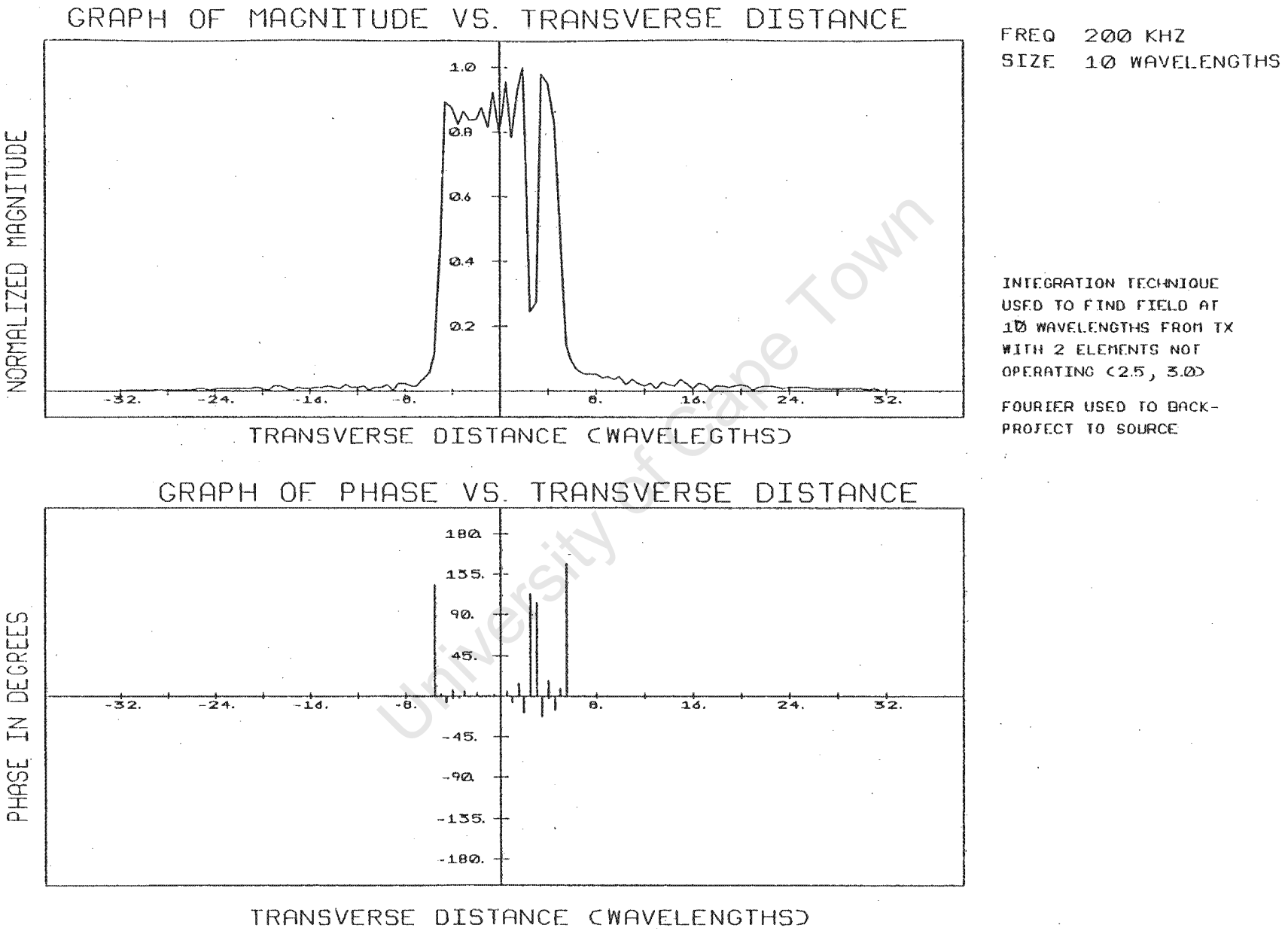


Figure 5.26: Detection of element failure using the Fourier back projection technique.

5.9. Simulation of two-dimensional source giving a three-dimensional distribution

The simulations performed to produce three-dimensional plots of pressure are essentially the same as those performed for a two-dimensional pressure pattern. The source is, however, finite in both dimensions. Two complications are introduced in the three-dimensional case. Firstly, the swopping routine has to deal with four sectors instead of two to get the data in the correct form for computer processing. Figure 5.27 shows how the swopping is performed.

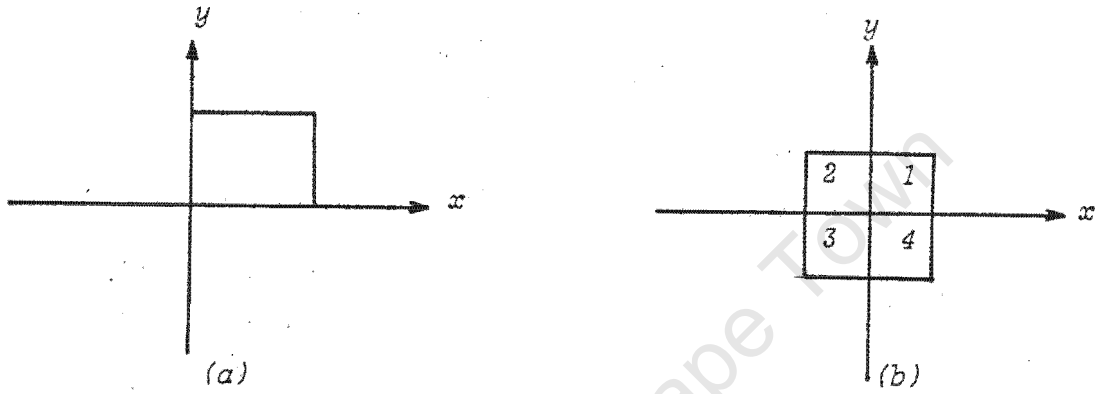


Figure 5.27: (a) Form of input required for computer processing.
(b) Form of the input data.

The swopping routine in the program exchanges data in quadrants 2 and 4. This routine is given in listing C.5. Similarly the data in quadrants 1 and 3 are exchanged, as shown in Figure 5.27 (b).

The second complication arises from the bandlimited nature of the Fourier transform technique. It was shown in Chapter 3 that for $(k^2 - k_1^2 - k_2^2)^{1/2} < 0$, the plane waves were strongly attenuated for distances greater than a few wavelengths. Thus it shall be assumed that $(k^2 - k_1^2 - k_2^2)^{1/2} > 0$. This limits k_1 and k_2 to the shaded section shown in Figure 5.28.

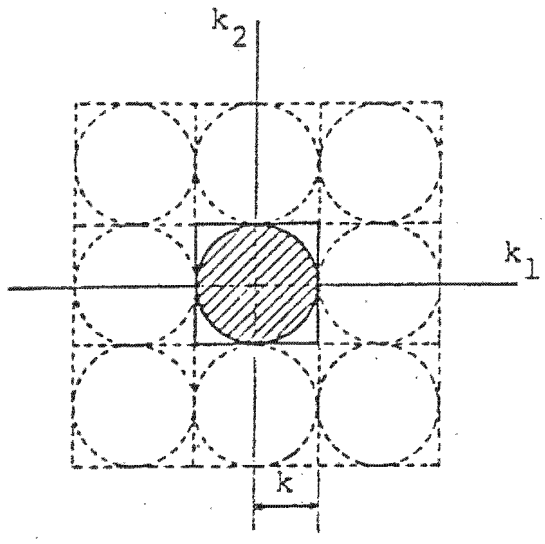


Figure 5.28: Spatial frequency plane (k_1 and k_2) showing band limited nature of frequencies.

The two-dimensional band limited function can be uniquely reconstructed if the Nyquist sampling rate is used. For the circular band limited region the sampling scheme becomes very complicated. If the region is a square, however, the two-dimensional sampling theorem is a direct extension of the usual one-dimensional sampling theorem. So functions are limited to the circumscribing square (this requires a sampling rate only about 12% higher than the theoretical minimum (2)).

The plots which follow compare the integration and Fourier techniques. It can be seen that at 10λ from the source, the two techniques agree fairly well, but at 25λ there is significantly more ripple on the Fourier transform generated plot. At 100λ, the sidelobes are almost lost in the 'noise', and the main lobe is unacceptably distorted.

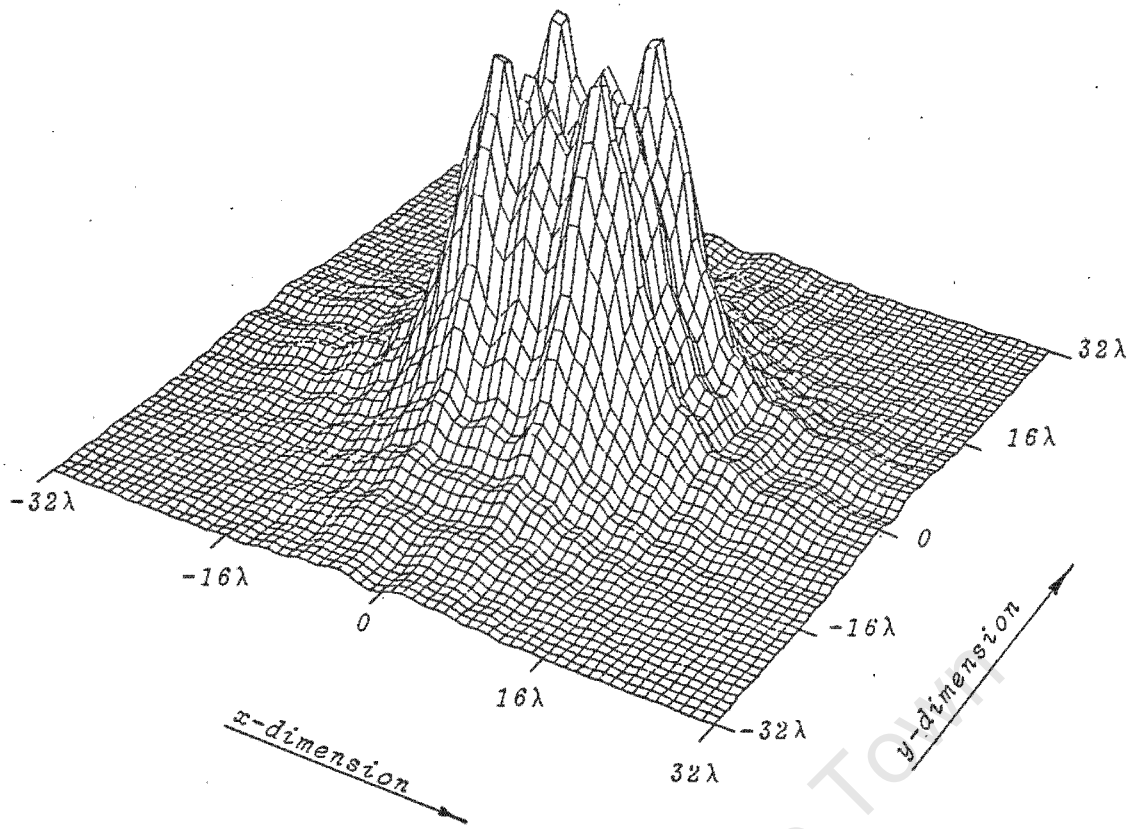


Figure 5.29(a): Pressure at 10λ from the source using integration technique.

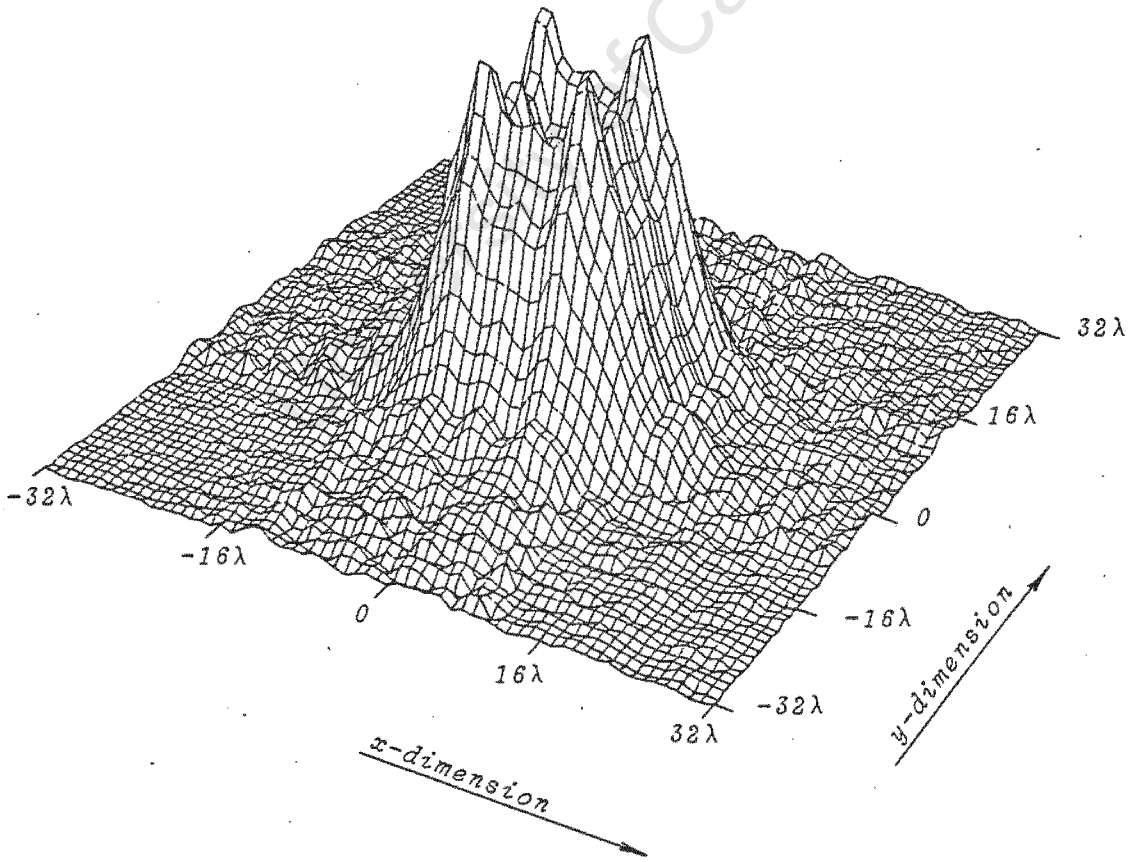


Figure 5.29(b): Pressure at 10λ from the source using Fourier technique.

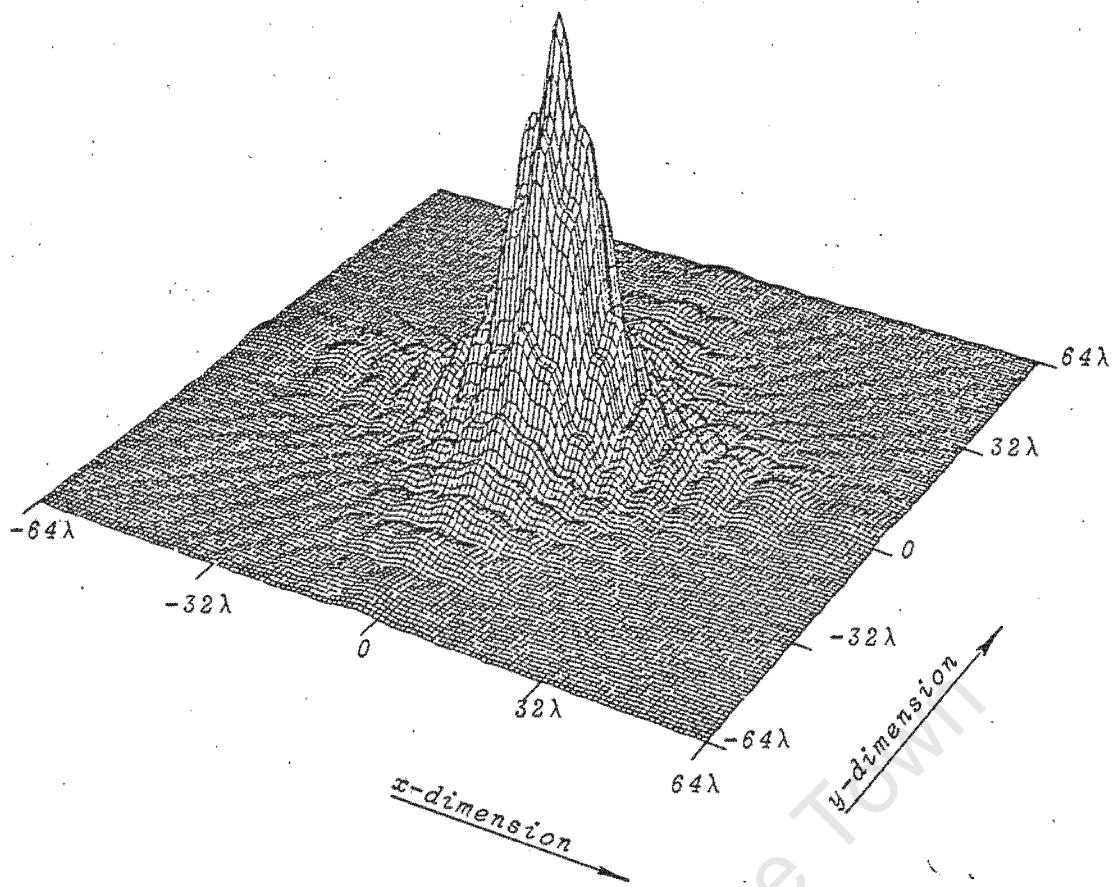


Figure 5.30(a): Pressure at 25λ from the source using integration technique.

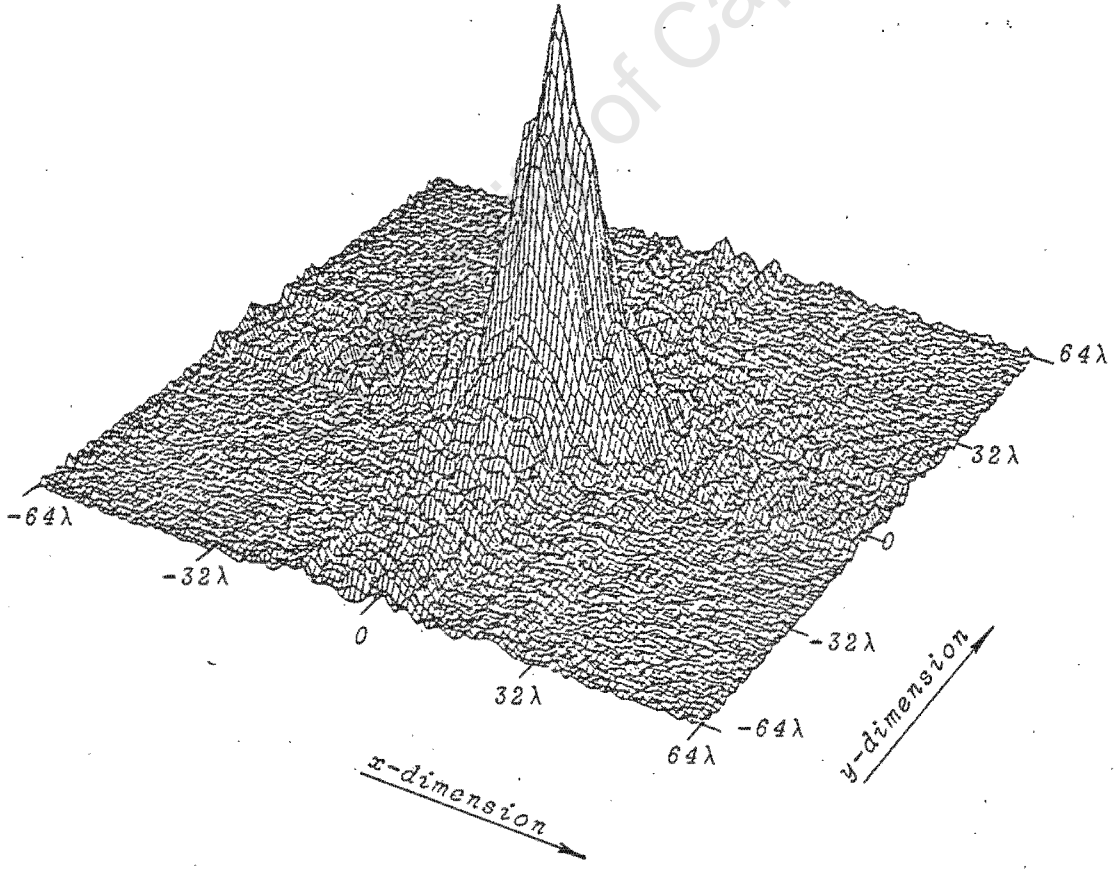


Figure 5.30(b): Pressure at 25λ from the source using Fourier technique.

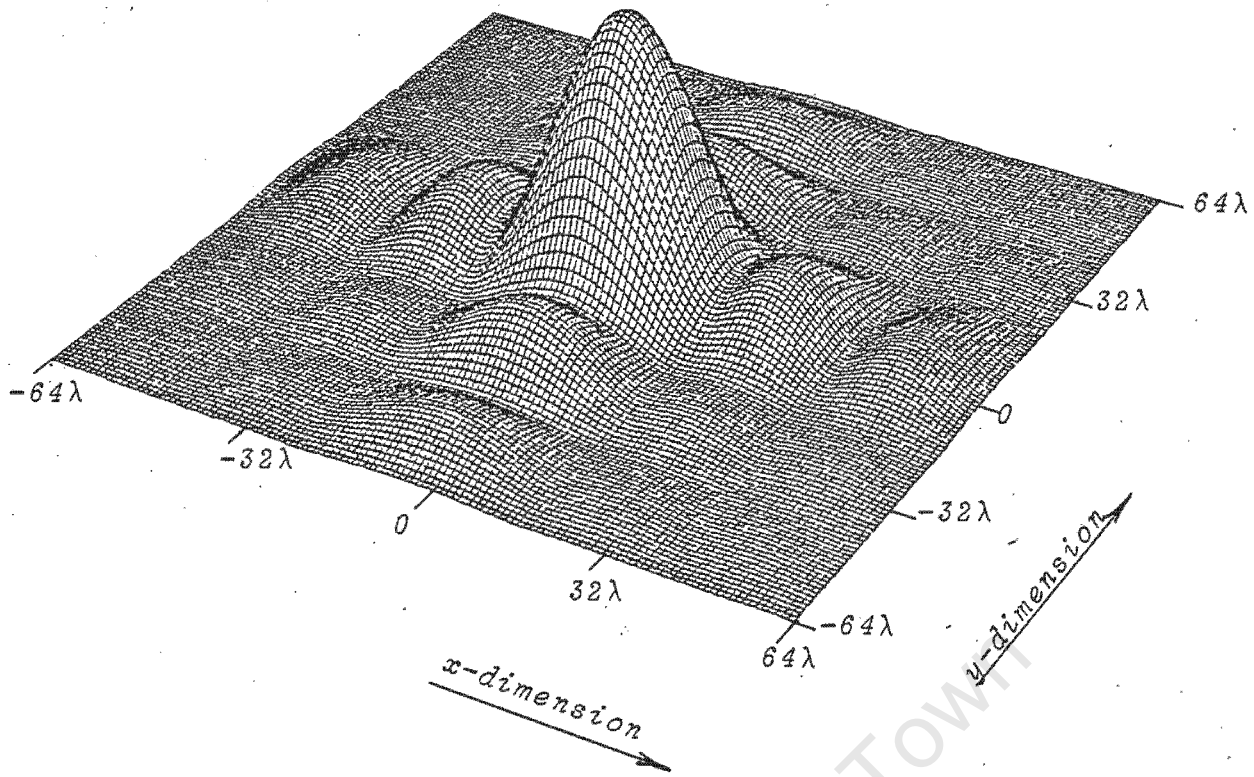


Figure 5.31(a): Pressure at 100λ from the source using integration technique.

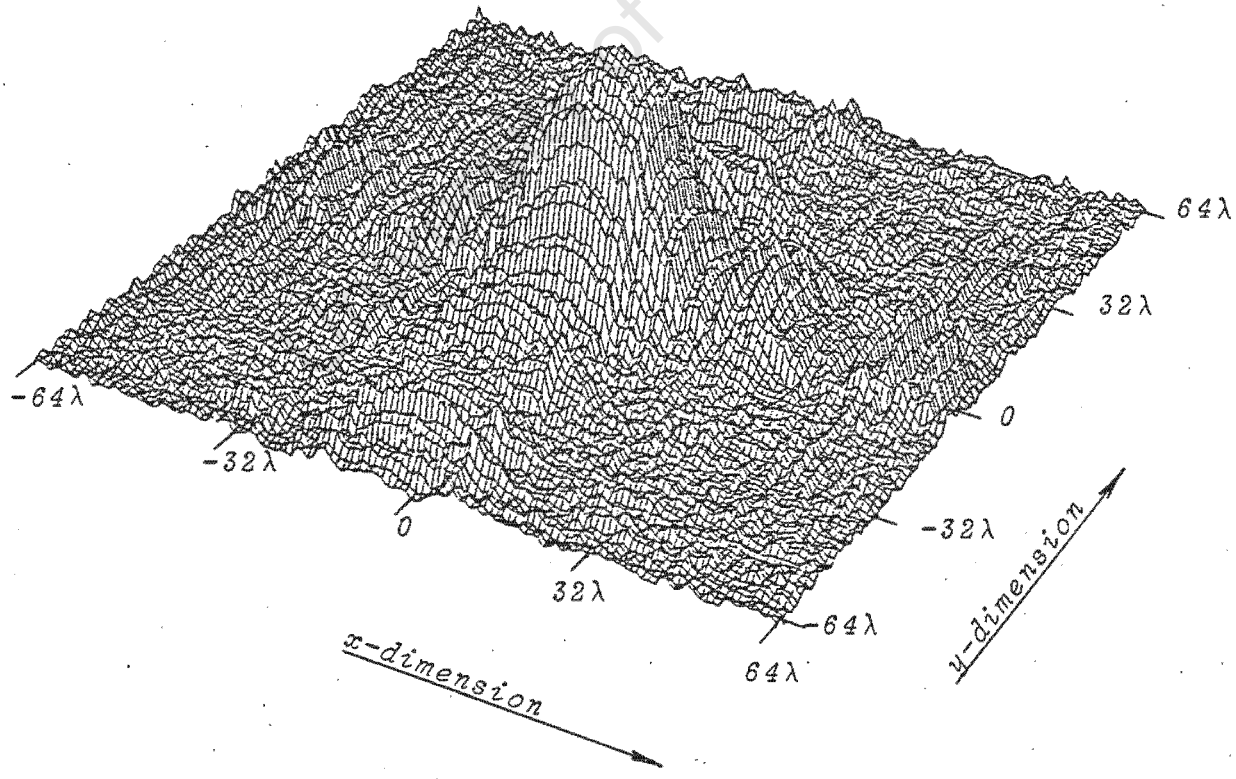


Figure 5.31(b): Pressure at 100λ from the source using Fourier technique.

As in the two-dimensional case, an alternative technique was used to determine the farfield pressure pattern. The Fourier transform of the velocity distribution at the source is shown in Figure 5.32(a), this being the farfield pressure. The Fourier transform of the nearfield pressure (calculated at 10λ from the source using the integration technique), followed by a conversion from pressure to velocity, gives the farfield pressure in Figure 5.32(b). Since the two plots in Figure 5.32 are similar, it can be concluded that the Fourier transform of the pressure at any plane, followed by a conversion of pressure to velocity, gives the farfield pressure.

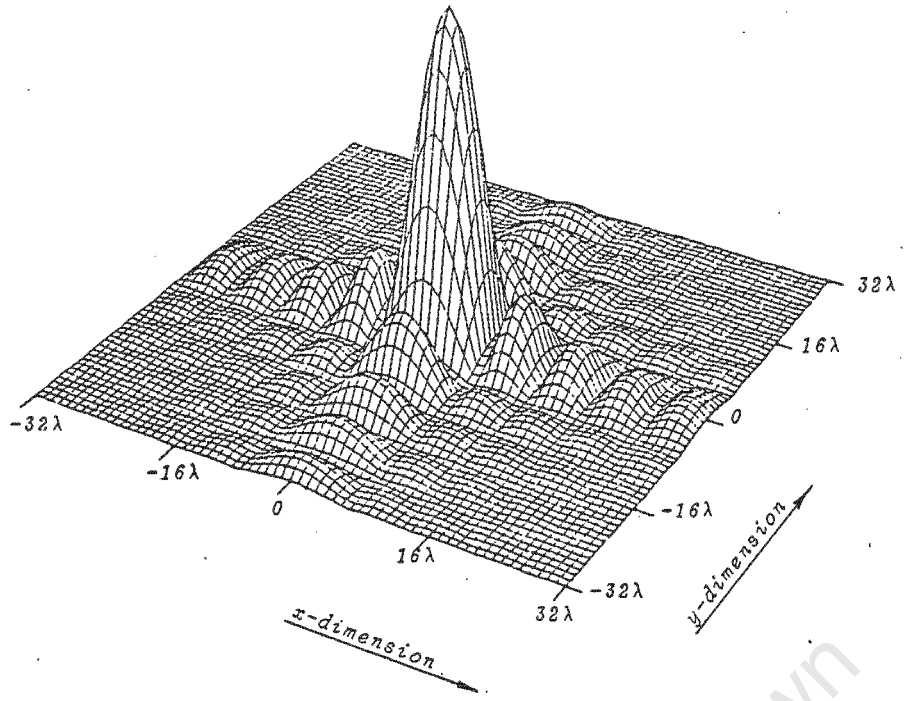


Figure 5.32(a): Fourier transform of velocity field on transducer gives farfield pressure pattern.

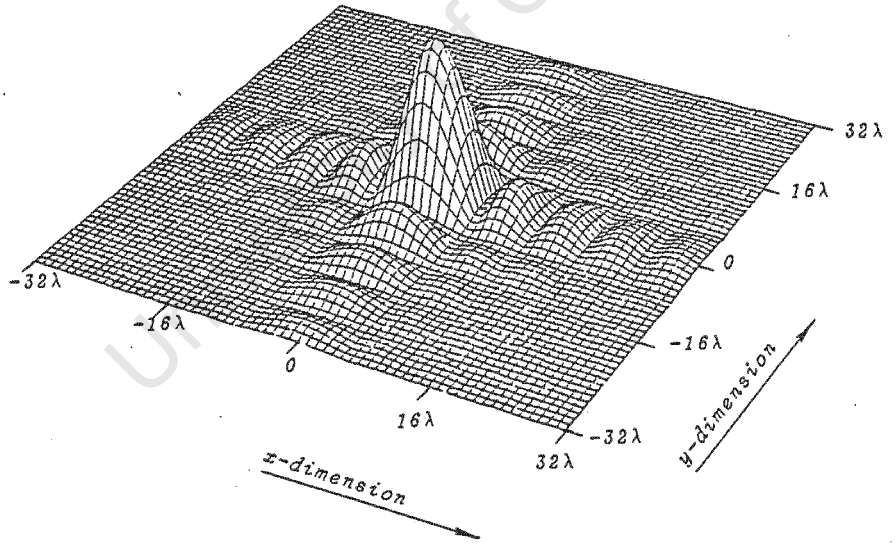


Figure 5.32(b): Farfield pressure pattern obtained from taking Fourier transform of pressure at 10λ from source.

The back projection technique was used to reconstruct the velocity distribution of the source, as given in Figure 5.33. The pressure field was calculated at 10λ from the source using the integration technique, then back projected using the Fourier technique. When back projecting a pressure field to obtain the velocity at the source, there are ripples in the velocity distribution due to the Gibbs-type phenomenon. This, as in the two-dimensional example, is due to the finite spatial bandwidth of the reconstruction process.

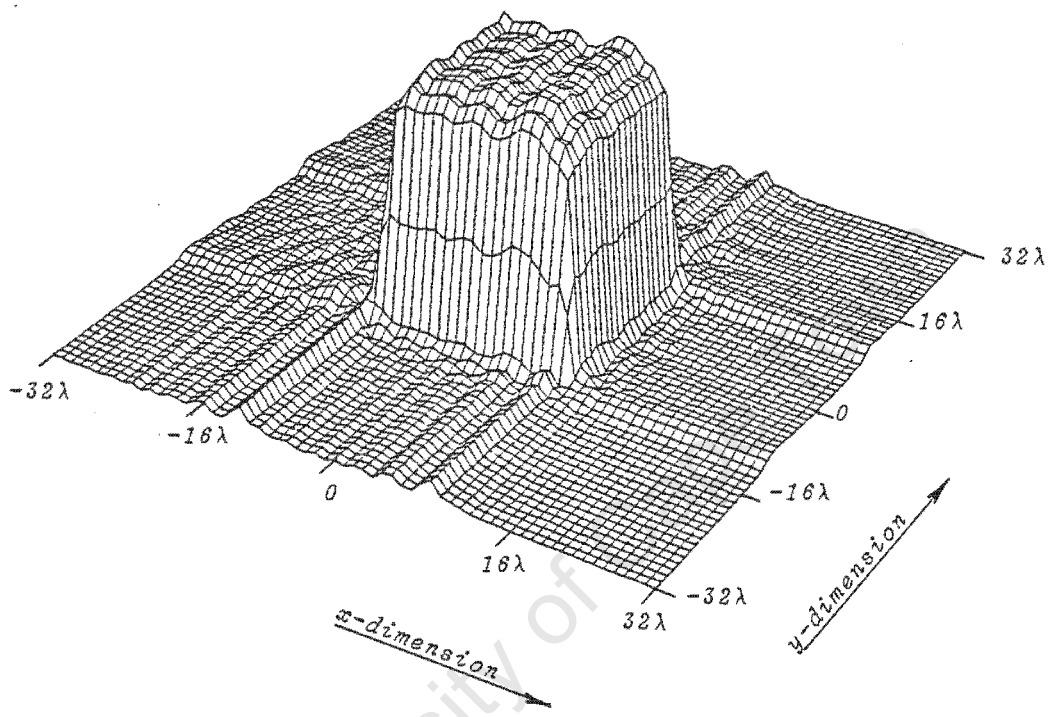


Figure 5.33: Velocity distribution of the source, determined by back projecting the integrated pressure data at 10λ from the source.

then by using the identity

$$\text{Cos A Cos B} = \frac{\text{Cos (A-B) + Cos (A+B)}}{2} \qquad (6.1)$$

the real component is given by

$$A \cos(\omega_o t) \cos(\omega_o t + \phi)$$

$$= \frac{A}{2} (\cos\phi + \cos(2\omega_o t + \phi))$$

By using a low pass filter with a cut off frequency of below $(2\omega_o t + \phi)$, the real component of the pressure will be:

$$\text{Real component} = \frac{A}{2} \cos\phi \qquad (6.2)$$

The imaginary component is found by using the identity

$$\text{Sin A Cos B} = \frac{\text{Sin (A-B) + Sin (A+B)}}{2} \qquad (6.3)$$

Delaying $\cos(\omega_o t)$ by 90° gives

$$\cos(\omega_o t + 90^\circ) = \sin(\omega_o t)$$

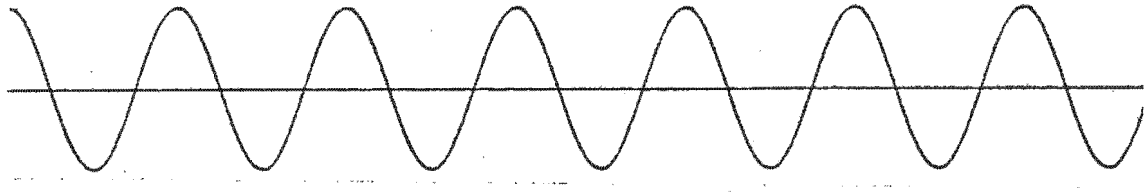
Multiplication yields

$$A \sin(\omega_o t) \cos(\omega_o t + \phi) = \frac{A}{2} (\sin\phi + \sin(2\omega_o t + \phi))$$

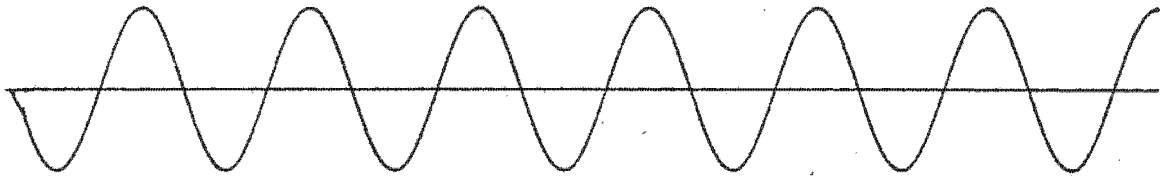
which when applied to a low pass filter gives

$$\text{Imaginary component} = \frac{A}{2} \sin\phi$$

This is shown diagramatically in figure 6.1 for $\phi=0$ and $A=1$ (i.e. the return signal is identical to the transmit signal).

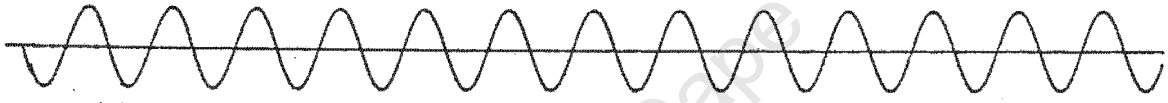


(a) Transmit signal = received signal



(b) 90° phase shifted transmit signal

(c) Transmit signal * received signal



(d) Transmit signal shifted by 90° * received signal

Figure 6.1: Calculation of complex pressure

Thus, if the signal in figure 6.1(c) is low pass filtered, the real valued component will be ½. A low pass filter applied to the signal given in figure 6.1(d) will yield 0 for the imaginary component. Circuitry to perform these multiplication and filtering operations was available at the Central Acoustics Laboratory. The operating frequency of this equipment was 200 kHz. For simplicity, this circuitry will be referred to as the multiplication box.

6.3 The experimental set-up

The source which was used was a 67mm diameter Ferruno echo sounder transducer which had a resonance close to 200 kHz. The transducer is shown in figure 6.2 and its farfield directivity response is given in Figure 6.3.

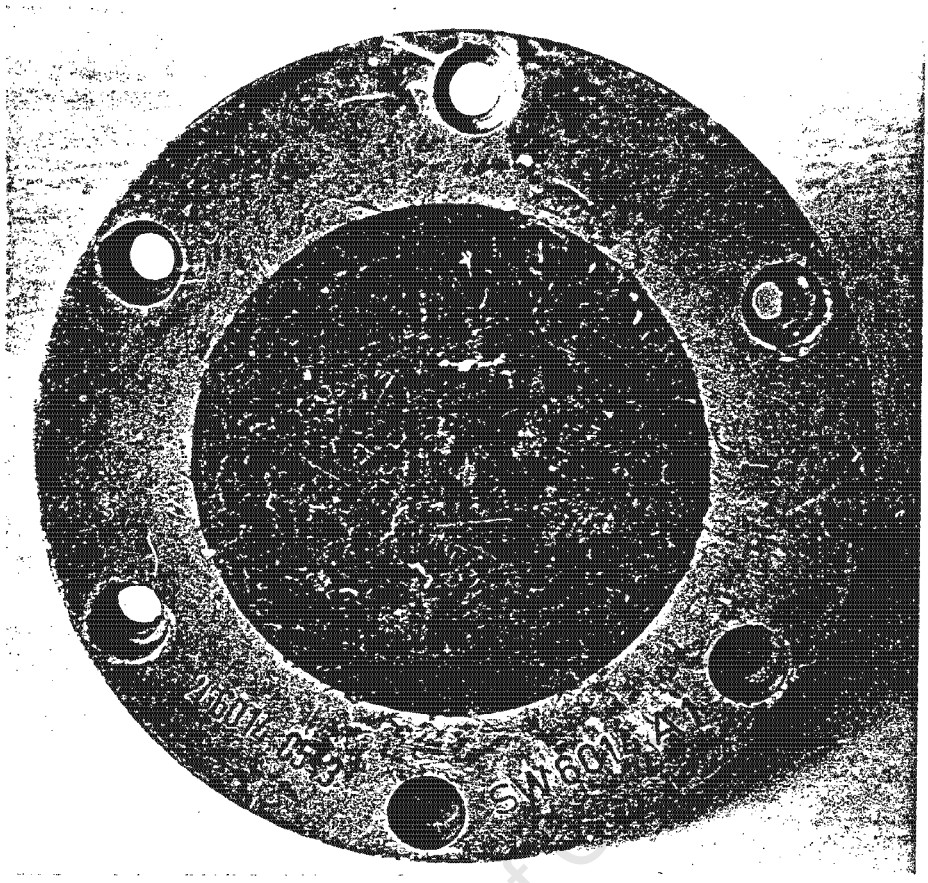


Figure 6.2: Ferruno echo sounder transducer used for the measurements

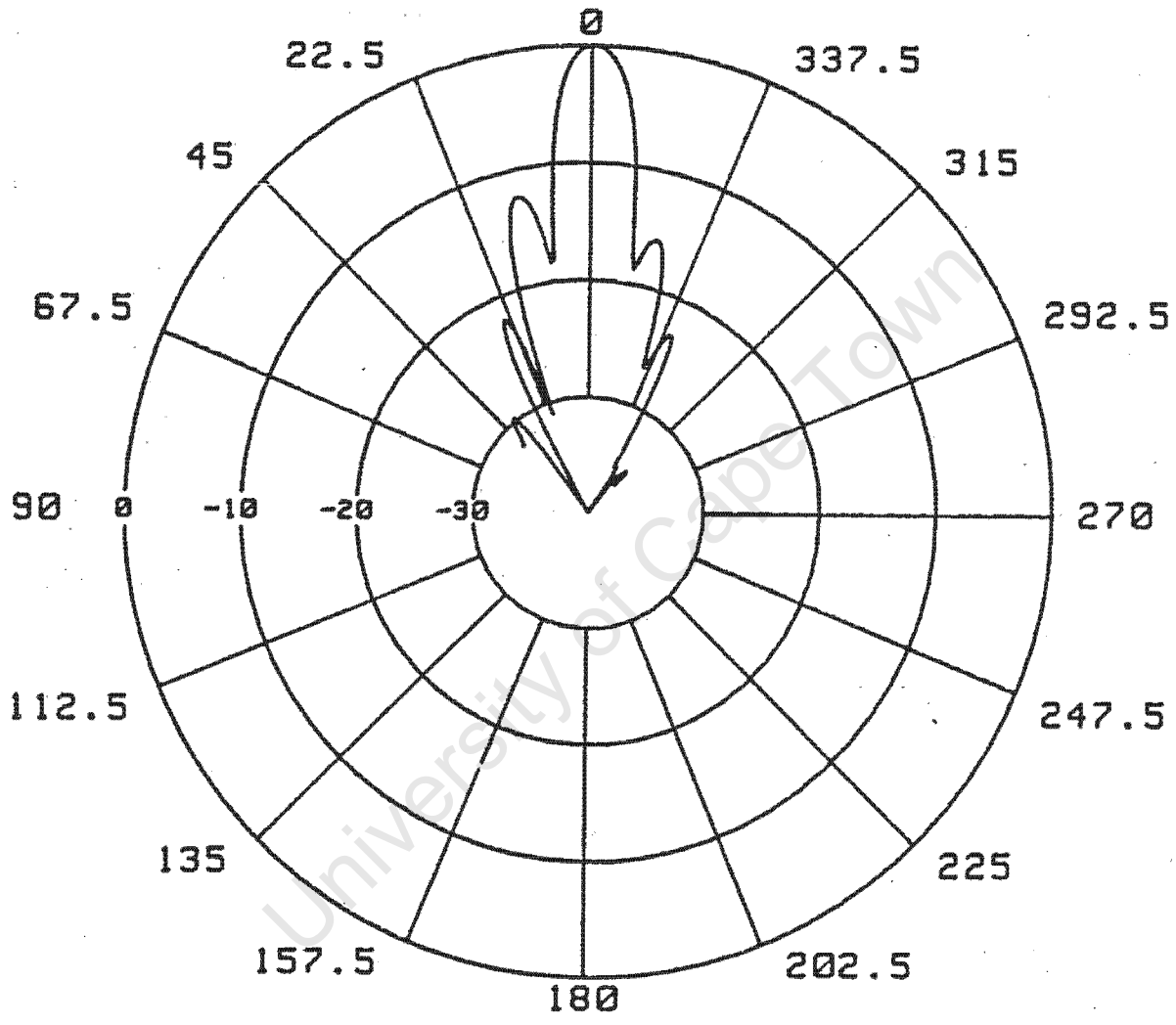


Figure 6.3: Farfield directivity response of Ferruno echo sounder transducer.

The probe used has an active face of 1.6mm by 1.6mm. It is a hollow cylindrical piezoelectric transducer made of PZ27 material.

The transmit signal was generated by an oscillator in the multiplication box. It was used as a reference signal for the multiplications. The received signal was amplified using a Brookdeal amplifier and fed into the multiplication box. The outputs of the box were the real and imaginary pressure components. These components were sampled using 2-channel Nicolete storage oscilloscope. The stored waveforms were then transferred to an HP-85. Figure 6.4 shows the measurement setup.

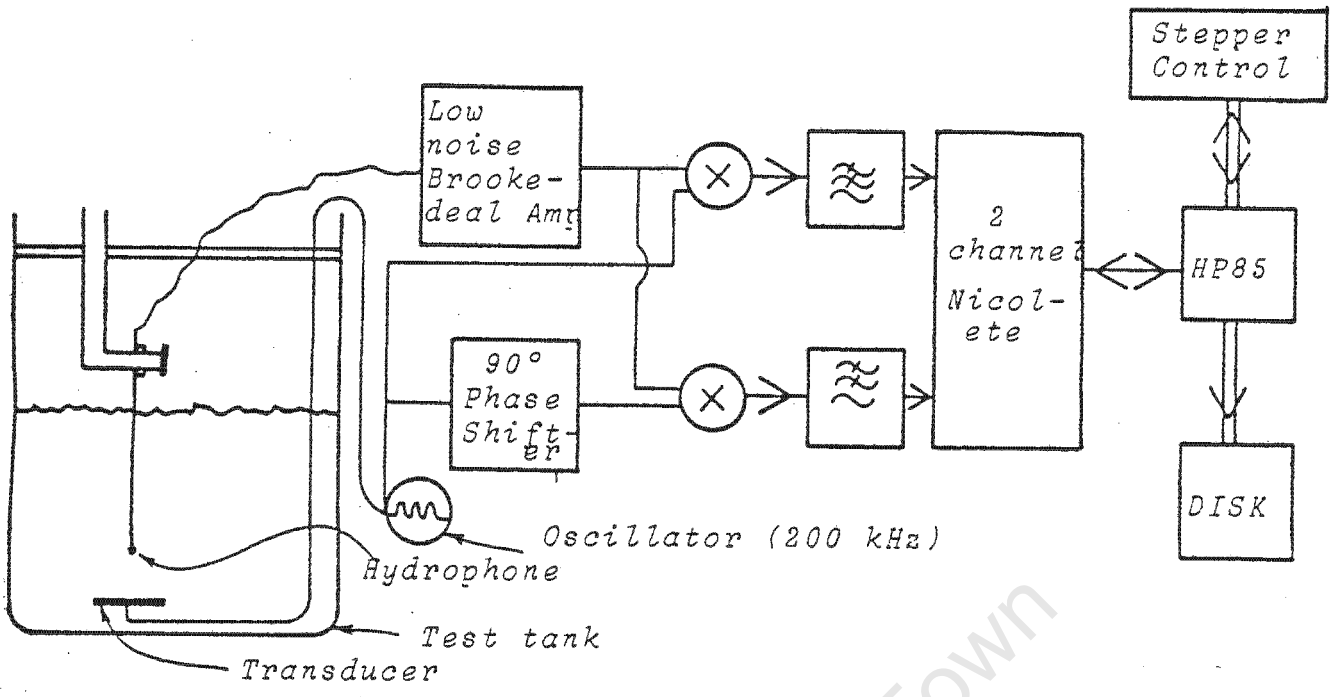


Figure 6.4: The measurement setup.

Typical waveforms at the input to the Nicolete are given in figure 6.5.

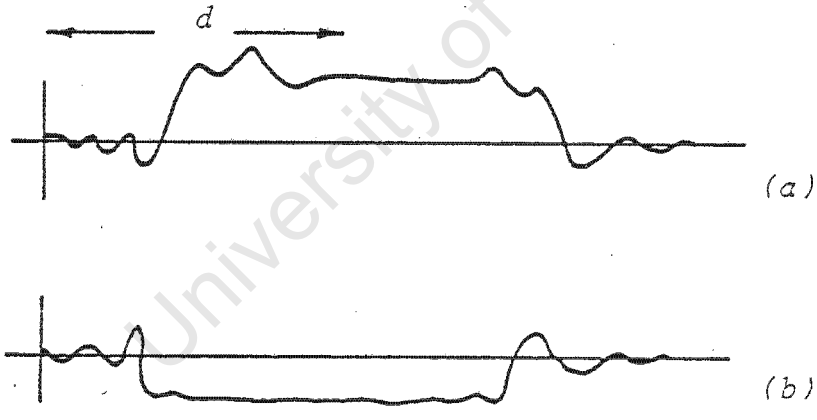


Figure 6.5: (a) Real component of the pressure
(b) Imaginary component of the pressure

The waveforms which were transferred to the HP-85 comprised 1024 samples each. The delay, d, as shown in figure 6.5 was programmed into the computer. This delay was used to eliminate transients from the signal. 50 samples after the delay were used, these being averaged to obtain single values for each of the real and imaginary components. These values were stored in an array.

An array of 32 x 32 real and imaginary values for pressure were obtained in this way. Once all of the points had been stored, the 1024 real and imaginary values were transferred to a data file on magnetic disk.

The control software, together with the program for transferring data from the Nicolete to the HP-85 are given in appendix C (listing C.1.)

When the 2048 points had been stored on disk, the file was transferred to a VAX-VMS computer for processing. The pressure field was plotted 3-dimensions.

6.4 Results

The first experiment was conducted to verify that the theoretically developed technique works in practice. The pressure was measured in two different planes; the first plane was at 59.2mm (8λ) from the source, and the other plane was at 81.4mm (11λ) from the source. The plots of these pressure measurements are given in Figures 6.6 and 6.7.

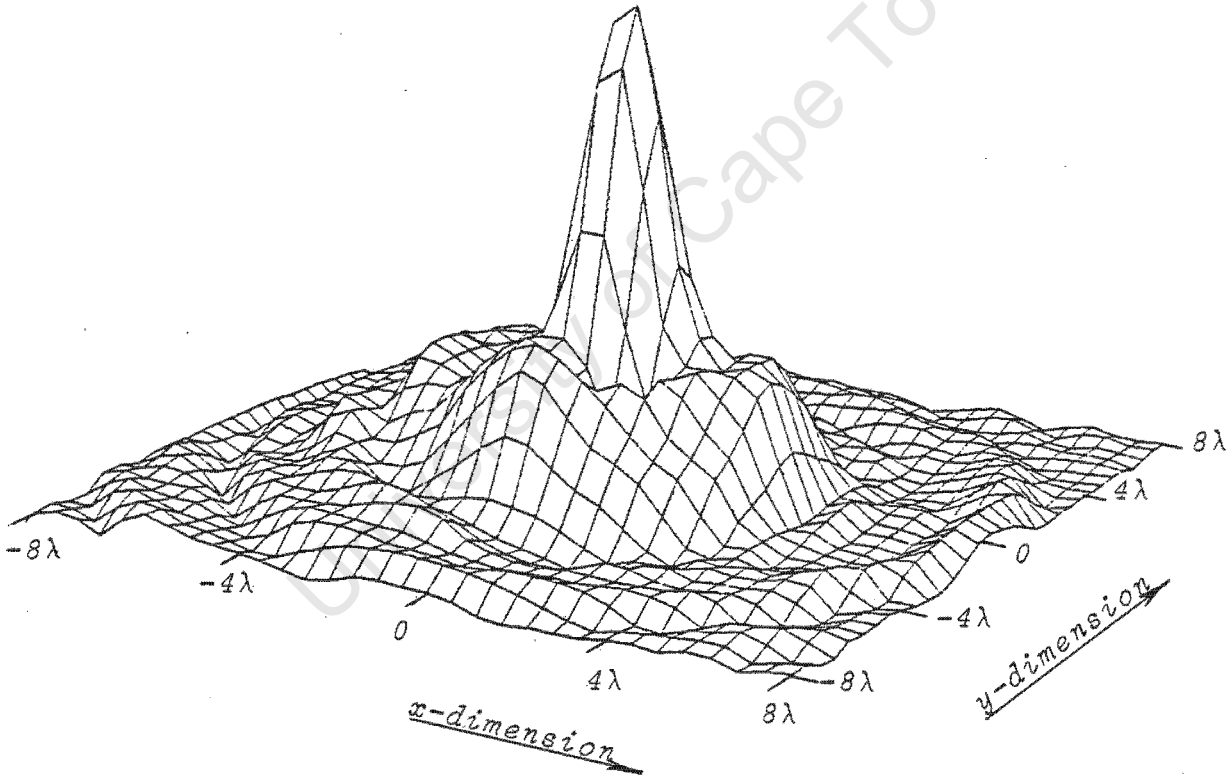


Figure 6.6 Pressure measurements at 59.2mm (8λ) from the transducer.

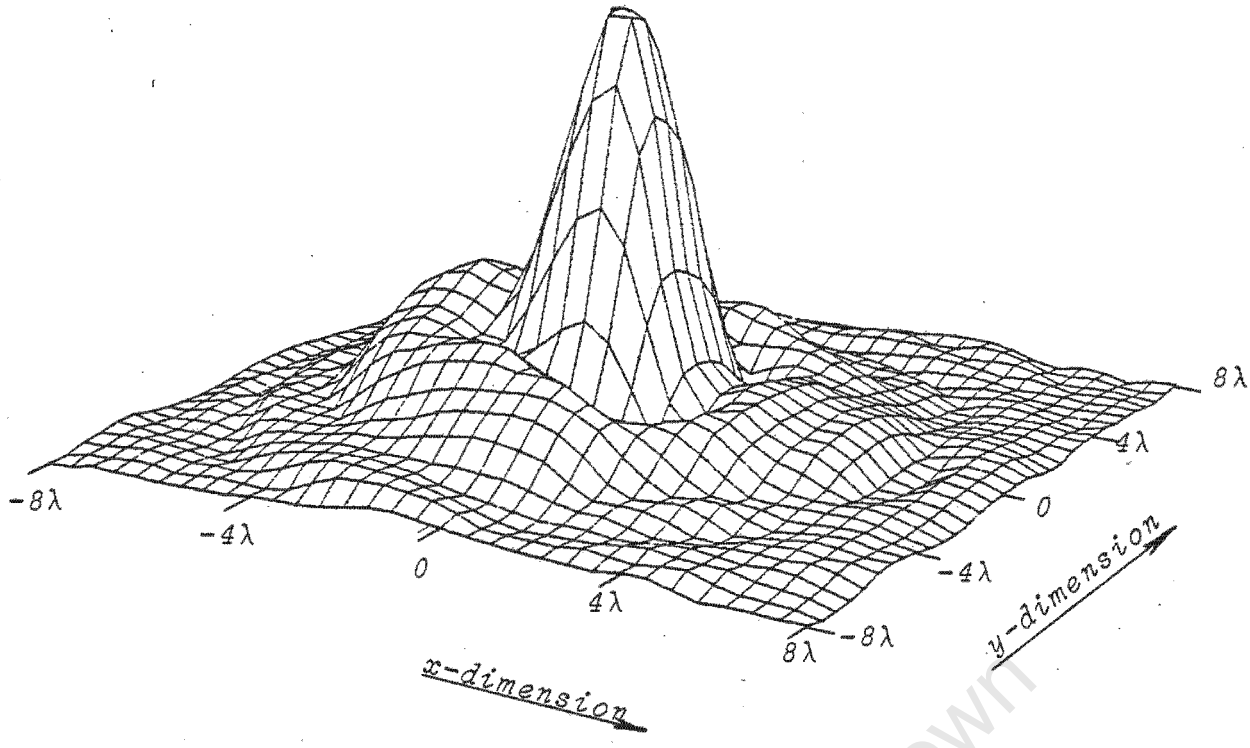


Figure 6.7: Pressure measurements at 81.4mm (11λ) from the transducer

The Fourier transform technique was used to forward project the measurements in the plane at 8λ from the source to obtain the pressure at the plane 11λ from the source. This projected field is shown in figure 6.8. The pressure data in the plane at 11λ from the source were backprojected to obtain the pressure at a plane 8λ from the source, as shown in figure 6.9.

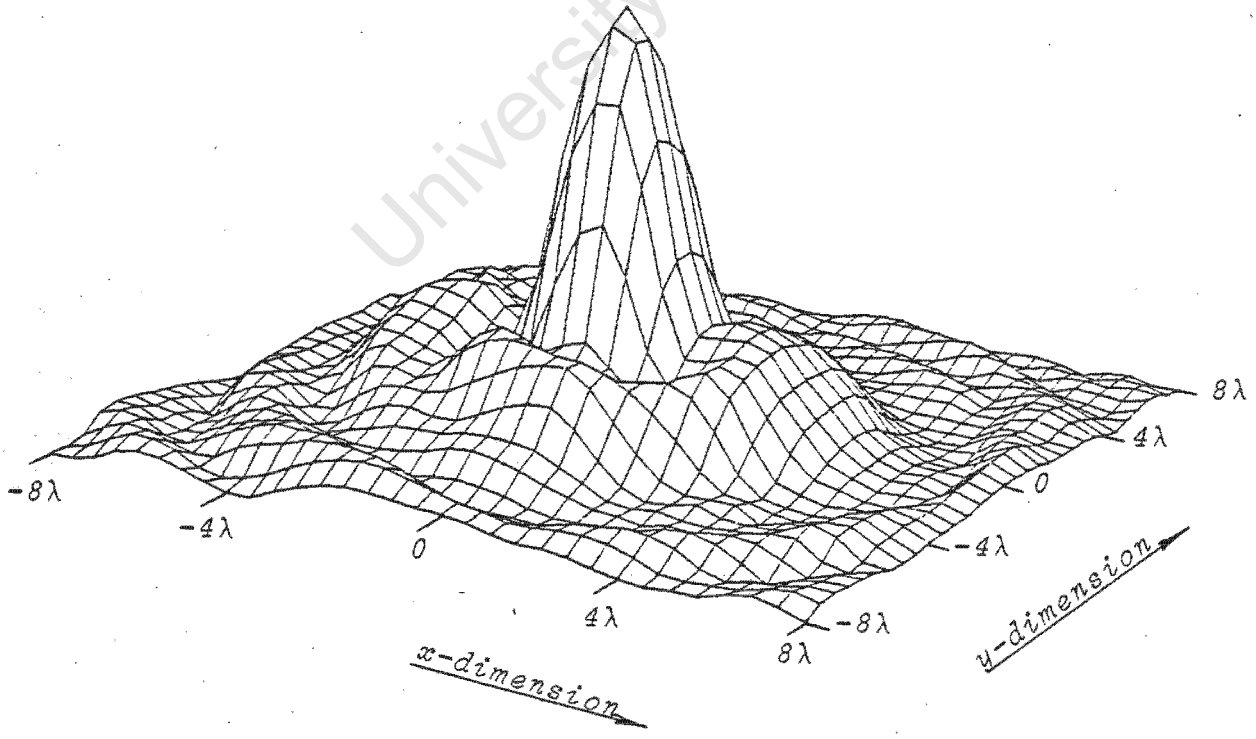


Figure 6.8: Pressure field projected from 8λ to 11λ.

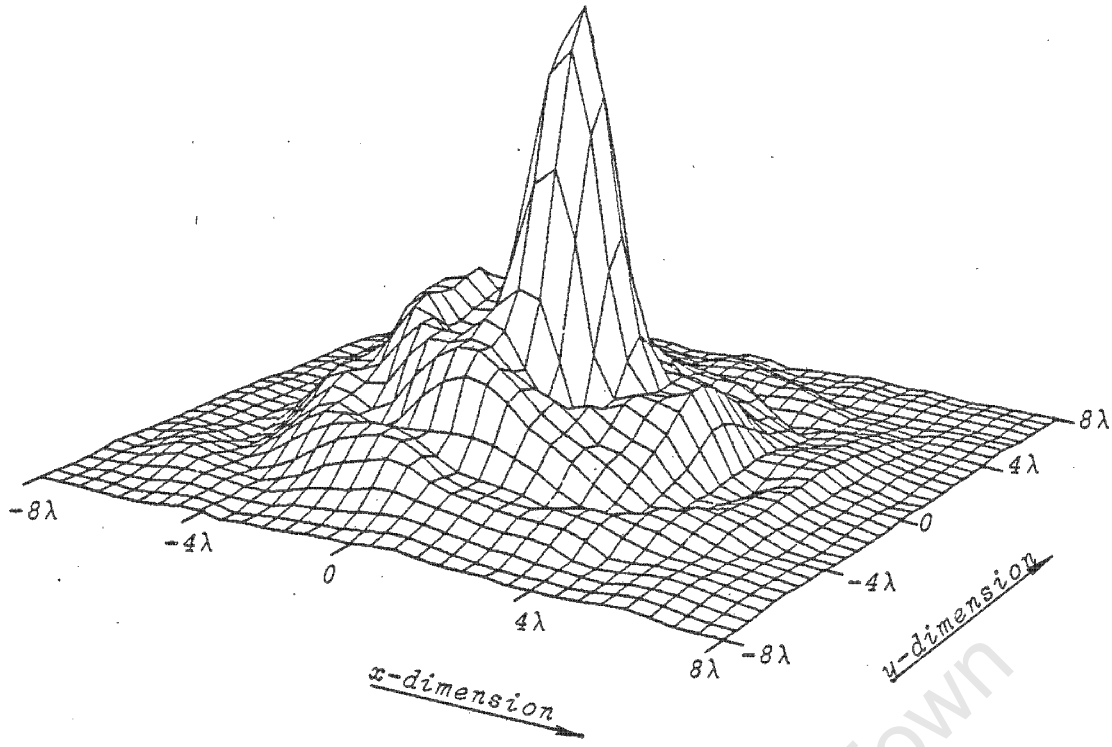


Figure 6.9: Pressure field projected from 11λ to 8λ .

There is good correlation between the measured and projected results. In particular, it can be seen that the beamwidth is approximately equal for the measured and projected data at 8λ from the source. This is also the case at 11λ from the source.

The velocity distribution of the source was found by back projecting the measured pressure field at 8λ from the source. This is shown in Figure 6.10

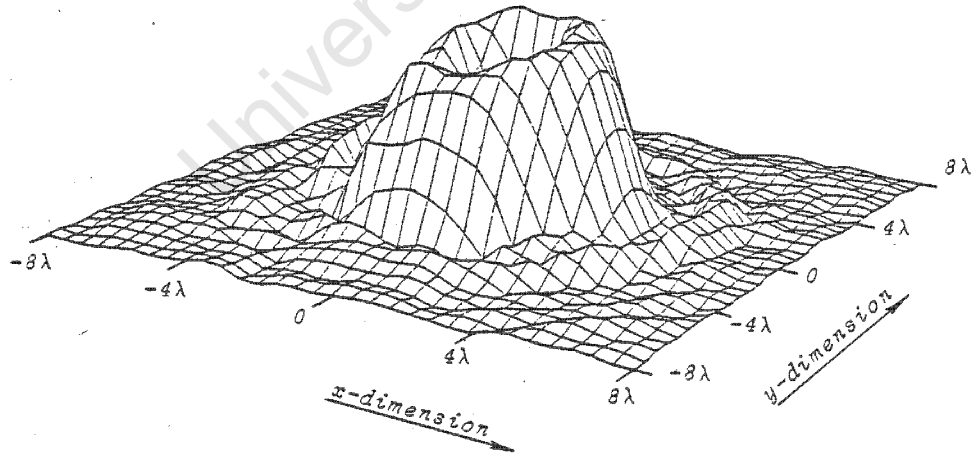


Figure 6.10: Back projected velocity on face of transducer

The calculated velocity distribution is not as flat, and the sides are not as steep as expected. There are two reasons for this. Firstly, the spectrum was truncated before the frequency components had decreased to an acceptable level. Secondly, not enough measurements were taken in the transverse direction to fully cover the significant part of the beam. It was found that the pressure at the edge of the measurement plane was still relatively high - 10% of the maximum pressure.

The next experiment attempted to correct these two limitations. Measurements were made closer to the source, thereby reducing the spread of the beam and thus including more information. The magnitude of the pressure at the edge of the plane was reduced in this way to 2% of the maximum pressure. The measurements were taken at 44.4mm (6λ) from the source. The pressure pattern is shown in figure 6.11.

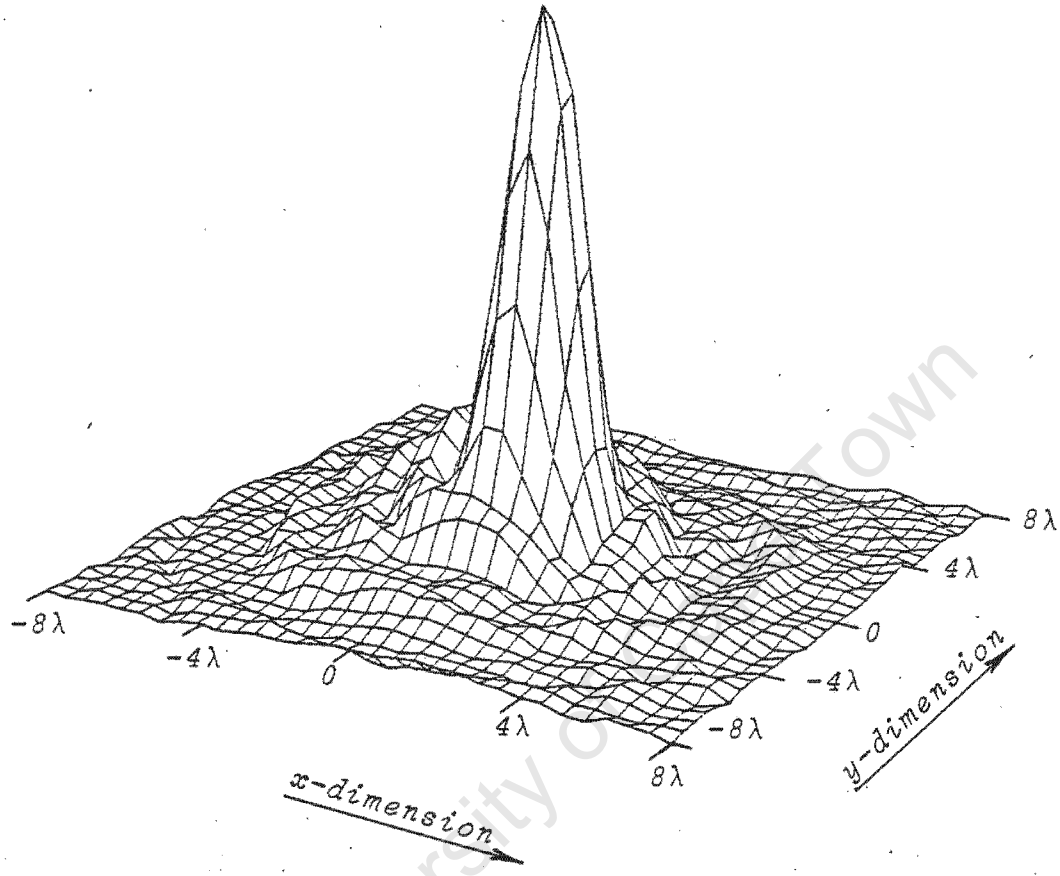


Figure 6.11: Measurements at a plane 44.4mm (6λ) from the transducer

The back projected velocity is given in Figure 6.12. It can be seen from this plot that the dip in the velocity magnitude has been reduced. The velocity distribution is more similar to that which was expected.

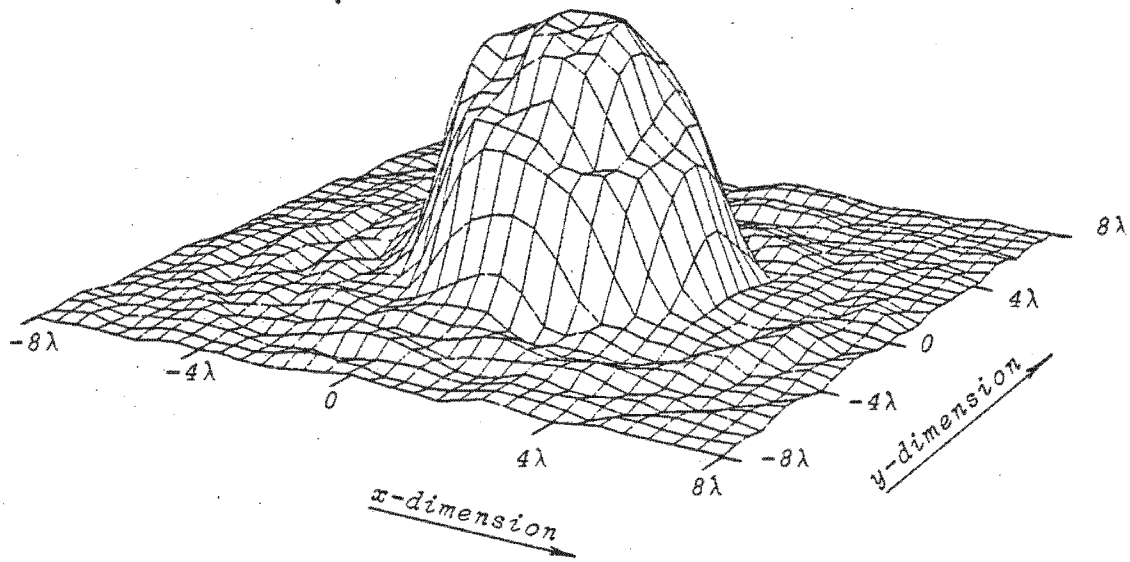


Figure 6.12: Velocity distribution of source determined by back projecting the nearfield pressure.

However, the prediction of the farfield pressure from nearfield measurements using the forward projection technique introduced ripples of unsatisfactory magnitude. This pressure pattern shown in Figure 6.13.

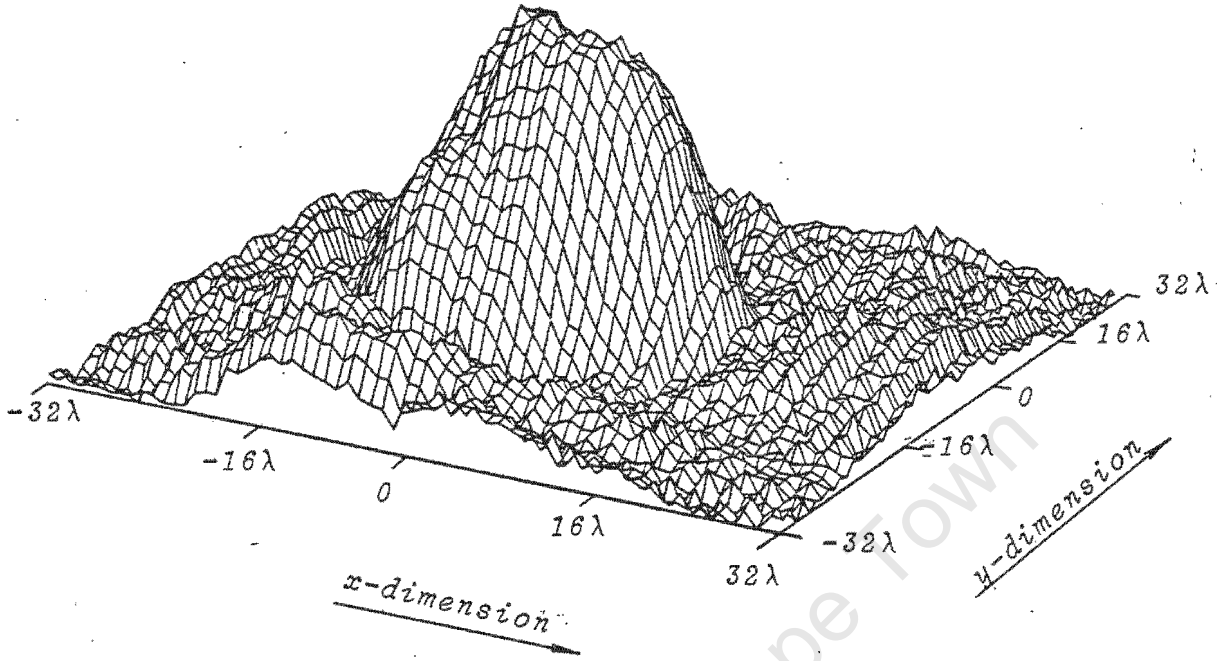


Figure 6.13: Farfield pressure determined from forward projecting the nearfield pressure.

The technique of taking the Fourier transform of the nearfield pressure, as described in Chapter 3 was used to obtain the farfield pressure. The results are correct, as shown in Figure 6.14.

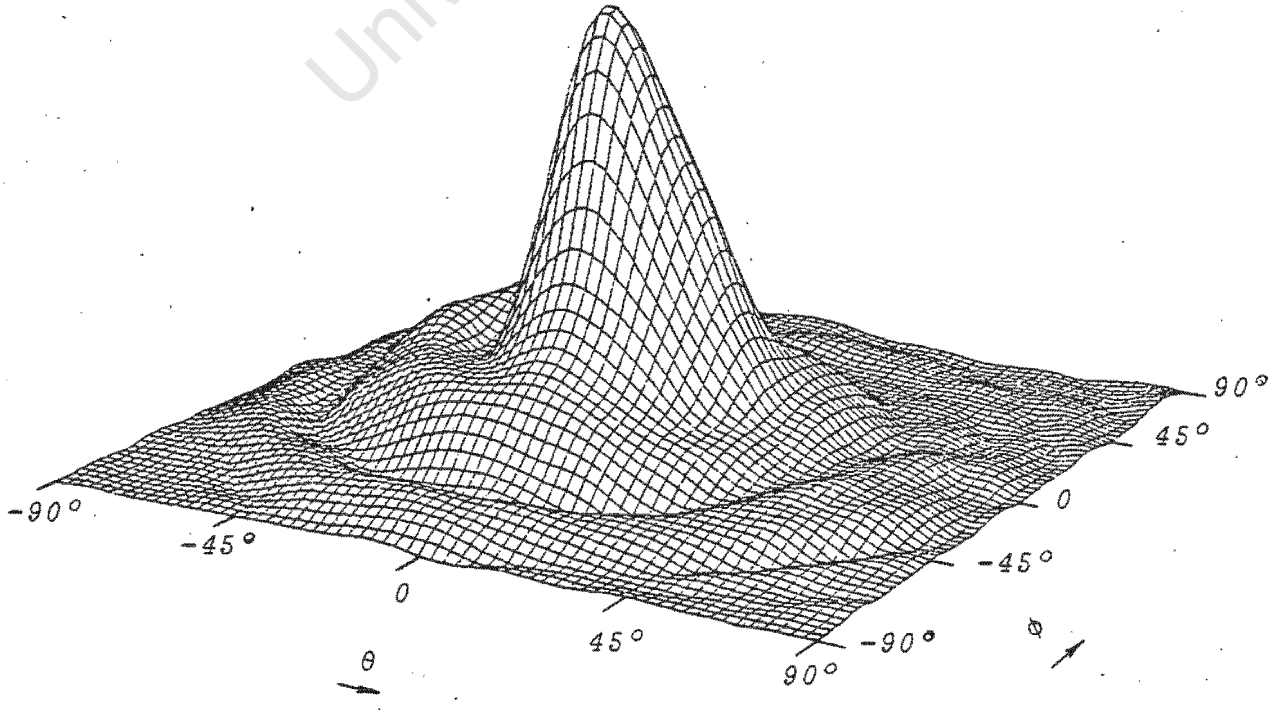


Figure 6.14: Farfield pressure pattern determined by taking the Fourier transform of the nearfield pressure measurements.

The final set of experiments used non-porous material to mask of a part of the transducer. The measured pressure was back projected to find the velocity distribution on the face of the transducer. A photograph of the transducer with a strip of the non-porous material is shown in Figure 6.15.

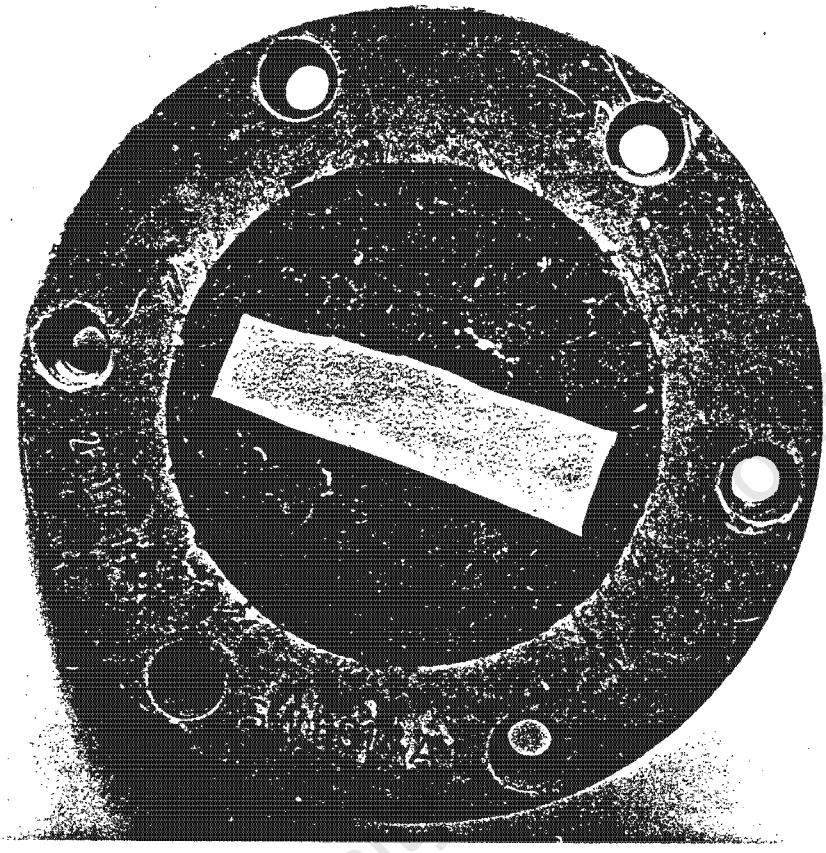


Figure 6.15: Photograph of transducer masked with a strip of non-porous material.

The measured pressure pattern at a distance of 59.2mm (8λ) from the source is presented in Figure 6.16.

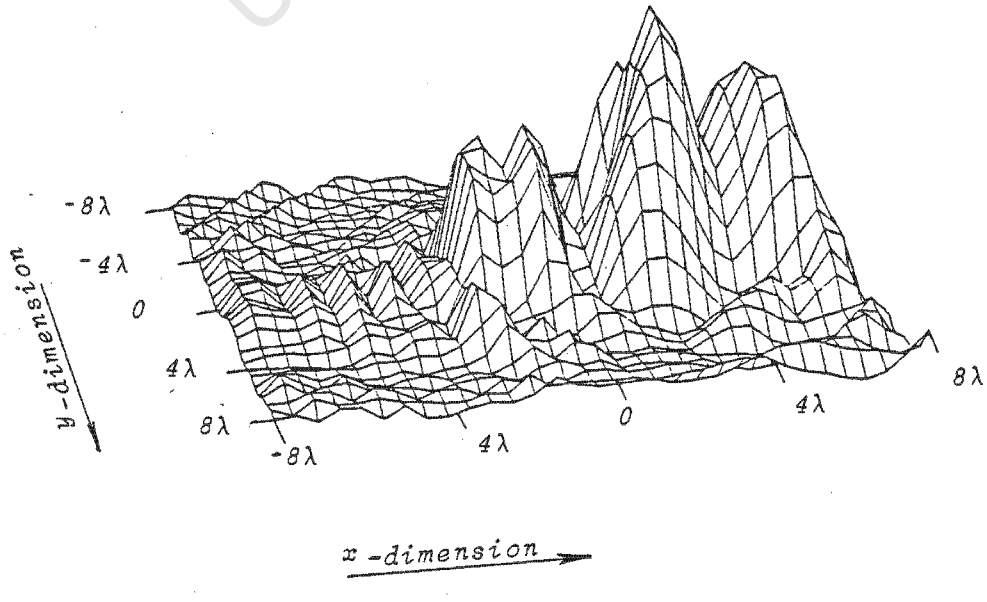


Figure 6.16: Measured pressure field of masked transducer at a distance of 59.2mm (8λ) from the source.

This pressure distribution was back projected to find the velocity distribution on the face of the transducer. This velocity distribution, given in Figure 6.17, agreed with the expected distribution. The transducer seems to be offset from the centre of the plot. This is due to the fact that the centre of the measured points was not alligned with the acoutic centre.

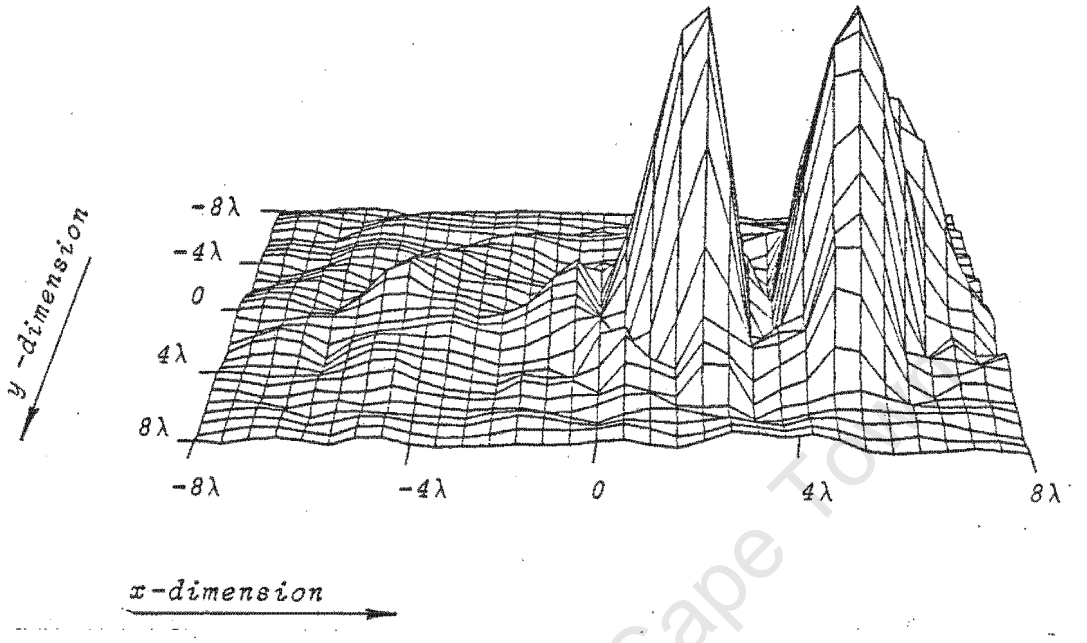


Figure 6.17: Back projected velocity distribution with strip of material masking a portion of the transducer.

An alternate representation of the velocity is given by the contour plot in Figure 6.18.

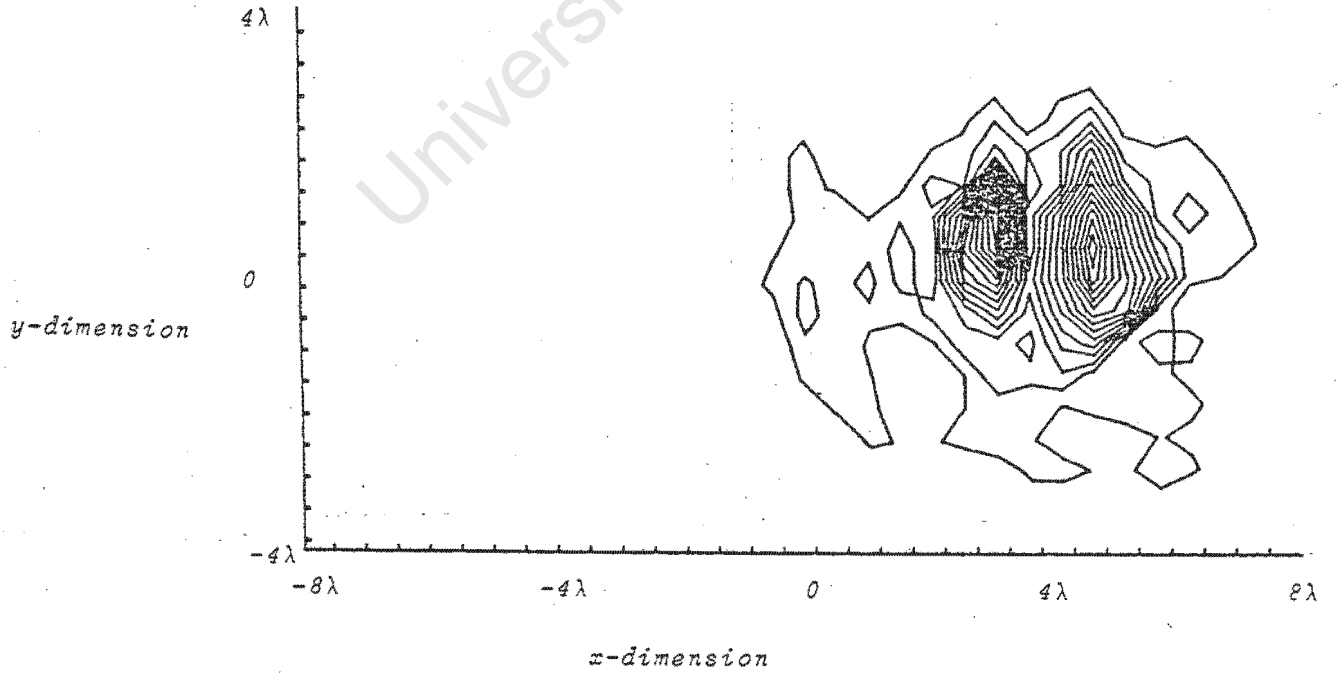


Figure 6.18: Contour plot of source velocity with non-porous material masking a strip of the transducer.

A letter 'E' was cut out of the non-porous material and placed firmly on the transducer face as shown in Figure 6.19.

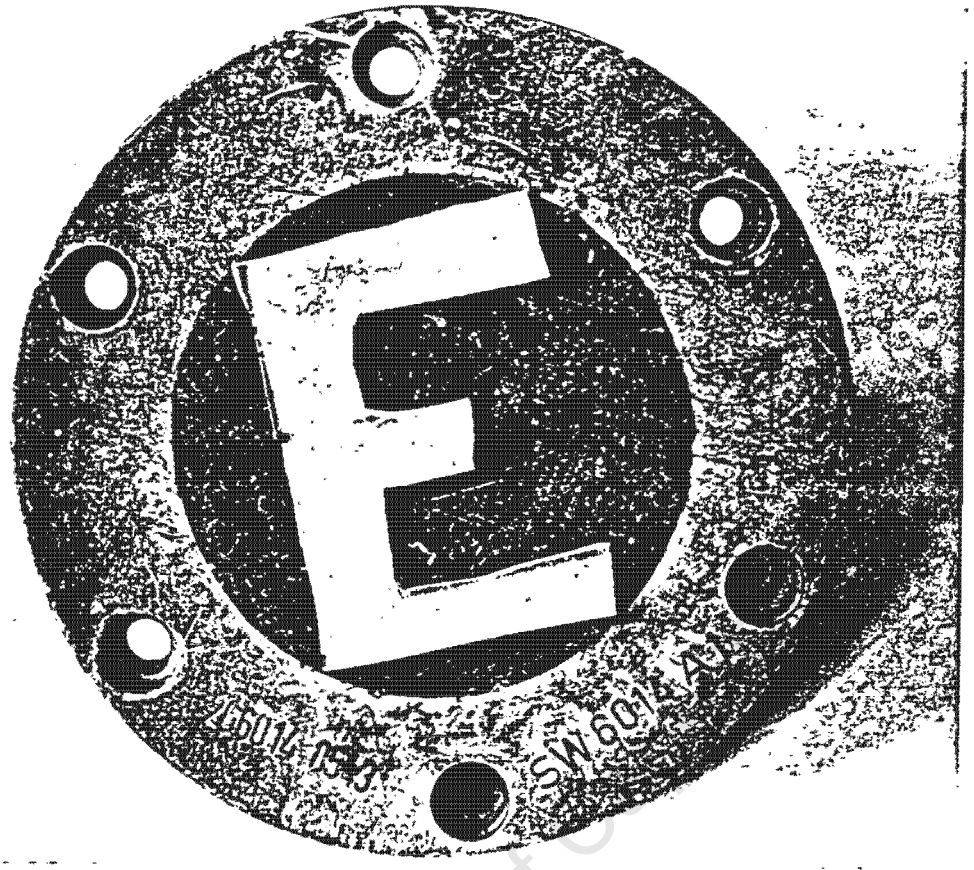


Figure 6.19: Photograph of the transducer with the letter 'E' placed on it.

The measurements were made at a distance of 44.4mm (6λ) from the source as shown in Figure 6.20.

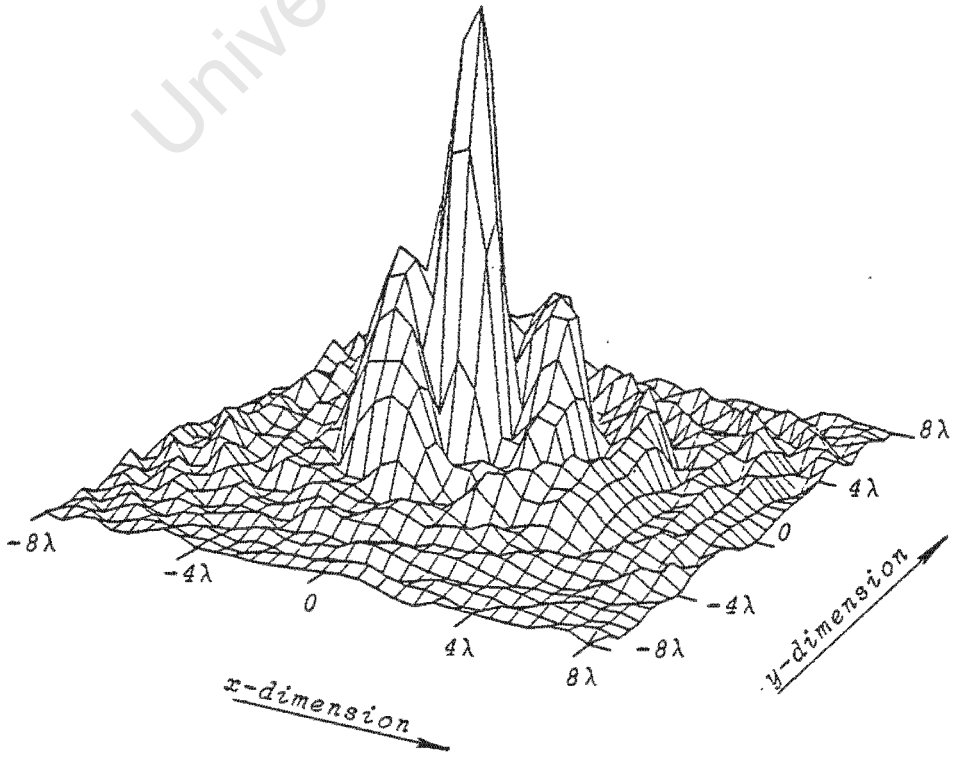


Figure 6.20: Measured pressure field at a distance of 44.4mm (6λ) from the masked source.

The back projected velocity pattern is shown in Figures 6.21 and 6.22 by a three-dimensional and a contour plot respectively.

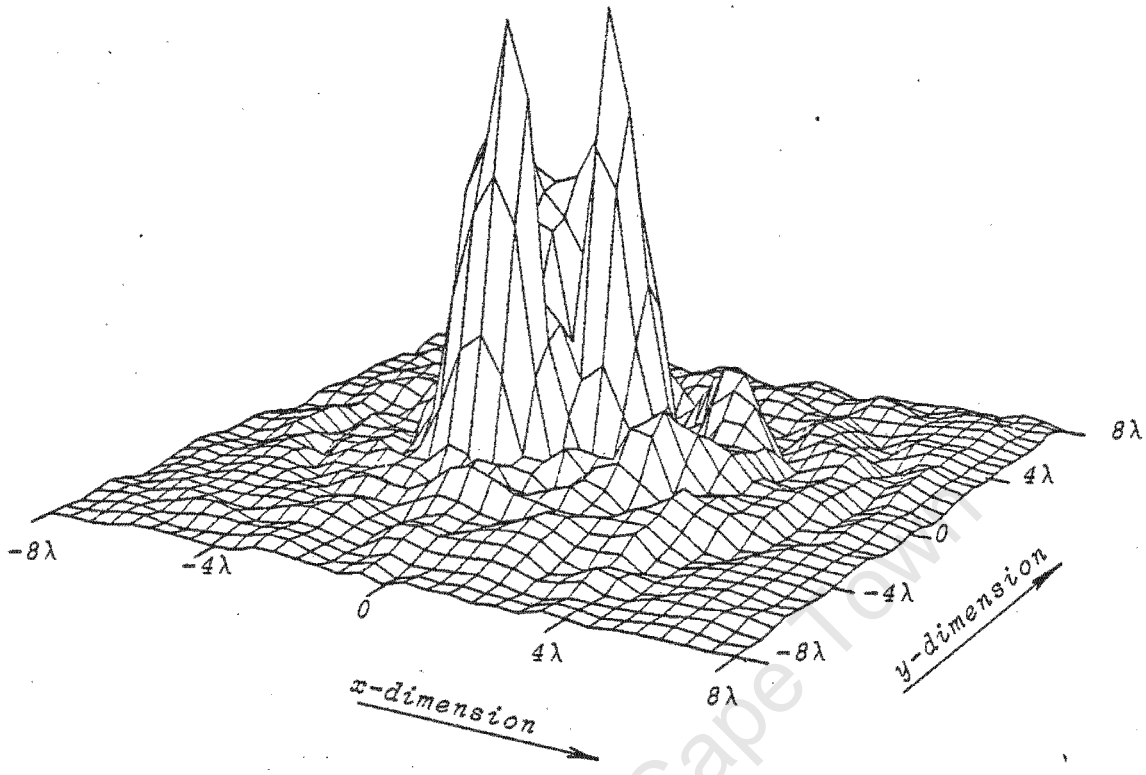


Figure 6.21: Back projected velocity distribution of transducer with 'E' placed on the face.

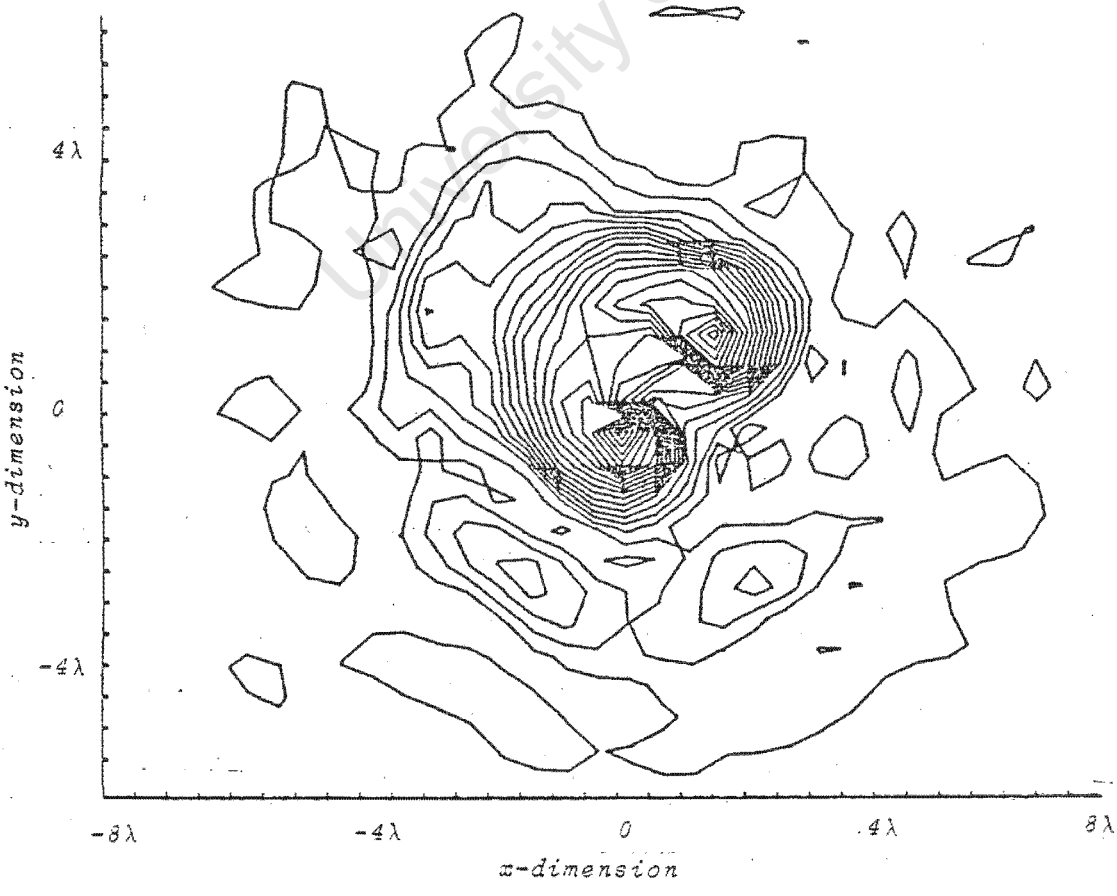


Figure 6.22: Contour plot of the back projected velocity distribution with 'E' placed on the face of the transducer.

The central portion of the 'E' is shown very clearly. The outer sections of the 'E', however, did not show up in the velocity plots. This is because the edges of the 'E' were at the physical edge of the transducer. Since the radiating element is usually 5-10% smaller than the casing, there was effectively no propagation beyond the limits of the 'E'.

CHAPTER 7

CONCLUSIONS

The theory of the forward and backward projection of acoustic fields using the Fourier transform has been developed. The aim of this presentation was twofold. Firstly, to give a simple explanation of the technique which was used to project acoustic fields. Secondly, to show that the technique works in practice, and to give the limitations of this method.

The basic objective of this thesis was to obtain the farfield pressure pattern using nearfield pressure measurements. The forward projection formulae as developed in Chapter 2 were used to achieve this aim. It was found that this method was not suitable because of the large projection distance. By performing a single Fourier transform on the measured nearfield pressure, however, the farfield beam pattern could readily be obtained.

If the pressure distribution at a plane other than the farfield is required, the entire forward projection process must be used. The further the projection distance, the more points are needed. The pressure patterns which are obtained in this way correlated with expected results, for relatively small projection distances.

The back projection of the measured nearfield pressure to the velocity distribution on the source yielded interesting results. It was found that the entire beam had to be covered in the measurement plane in order to obtain the correct velocity distribution at the source. The masking of the transducer with non-porous material showed that this method can be used to reconstruct the velocity distribution at the source.

There are limitations to the use of the Fourier transform technique. Firstly, the source must be large enough so that the plane waves propagate in front of the transducer. One cannot use a point source, for example, which is much smaller than a wavelength. This source causes plane waves to propagate at angles greater than 90° and less than -90°. Secondly, because of the property of the DFT that the source is periodic, ther is interference from repeated sources which are too close. The solution to this problem is to use more points, thus increasing the period of the repeated sources.

One must also not forget that an accurate probe positioning experimental set-up is required to make the nearfield measurements. This was available for the experiments performed for this thesis, but it would be costly to set up a similar system.

The area of interest for further investigations is the use of the back projection technique. This process has been shown to be a very powerful tool for analysing the source of radiation. By back projecting nearfield pressure measurements, further insight can be provided of the mechanics of the radiating source.

The back projection technique was shown to be capable of detecting a non-radiating element of an array. Additional research could investigate methods to enhance the resolution of this detection process.

REFERENCES

1. Bobber, R.J., "Underwater Electroacoustic Measurements", (Naval Res. Lab., U.S. G.P.O., Washington, D.C., 1970), Chapter 4.

2. Sondhi, M.M., "Reconstruction of Objects from Their Sound-Diffraction Patterns", J. Acoust. Soc. Am. 46, 5(2), 1158-1164, (1969).

3. Williams, E.G., Maynard, J.D., and Skudrzyk, E., "Sound source reconstructions using a microphone array", J. Acoust. Soc. Am. 68(1), 340-344 (July 1980).

4. Williams, E.G., "Numerical evaluation of the radiation from unbaffled, finite plates using the FFT", J. Acoust. Soc. Am. 74(1), 343-347 (July 1983).

5. Stepanishen, P.R. and Benjamin, K.C., "Forward and backward projection of acoustic fields using FFT methods", J. Acoust. Soc. Am. 71(4), 803-812 (April 1982).

6. Horton, C.W. and Innis, G.S. Jr., "The Computation of Far-Field Radiation Patterns from Measurements Made near the Source", J. Acoust. Soc. Am. 33(7), 877-880 (July 1961).

7. Van Buren, A.L., "Theoretical design of nearfield calibration arrays", J. Acoust. Soc. Am. 53(1), 192-199 (1973).

8. Rudgers, A.J., "Determination of the farfield radiation of a noise source from nearfield measurements made with a Trott array", J. Acoust. Soc. Am. 53(5), 1411-1416 (1973).

9. Clay, C.S., and Medwin, H., "Acoustical oceanography; principles and applications", (John Wiley, New York, 1977) pp. 153-155.

10. Zemanek, J., "Beam Behaviour within the Nearfield of a Vibrating Piston", J. Acoust. Soc. Am. 49(1) part 2, 181-191 (1971).

11. Ref. 10, pg. 181, Fig. 9(b).

12. Caruthers, J.W., "Fundamentals of Marine Acoustics", (Elsevier Publishing Company, Amsterdam, 1977), pp. 43-44.

13. Ref. 10, pg. 187, Fig. 1.

14. Kinsler, L.E., Frey, R., Coppens, A.B. and Sanders J.V., "Fundamentals of Acoustics", (Wiley, New York, 1982), pg. 98.

15. Higgins, F.P., Norton, S.J. and Linzer, M., "Optical interferometric visualization and computerized reconstruction of ultrasonic fields", J. Acoust. Soc. Am. 68(4), 1169-1176 (Oct. 1980).

16. Ref. 14, pg. 107.

17. Bergland, C.D., "A guided tour of the fast Fourier transform", IEEE Spectrum, vol. 6, 41-52 (July 1969).

18. Ref. 17, pg. 43, Fig. 3.

19. Ref. 14, pg. 173, eq. 8.26.

20. Sziklas, E.A. and Siegman, A.E., "Mode calculations in unstable resonators with flowing saturable gain. 2: Fast Fourier transform method", Applied Optics 14(8), 1874-1889 (August 1975).

21. Freedman, A., "Sound field of a rectangular piston", J. Acoust. Soc. Am. 32(2), 197-209 (February 1960).

22. Powers, J.P., "Computer simulation of linear acoustic diffraction", in Acoustical Holography, edited by Lawrence W. Kessler (Plenum Press, New York, 1977), Vol. 7, pp. 193-205.

23. Jongens, A.W.D., and Dean, D., "Underwater sound test facility at the Central Acoustics Laboratory", C. A. L. Internal Report no. 81-02, University of Cape Town, November 1981.

24. Junger, M.C. and Feit, D., "Sound structures, and their interaction", (Cambridge, Massachusetts, M. I. T. Press, 1972).

37. Sondhi, M.M., "Acoustic imaging by back propagation of the pressure field", Proceedings : Conference on Inverse Optics (SPIE : Arlington, Va) 150-152 (April 1983).

38. Johnson R.K., "Inverse propagation for acoustic transducer chatacterization", Proceedings: Conference on Inverse Optics (SPIE: Arlington, VA), 153-155 (April 1983).

39. Stremler, F.G., "Introduction to Communication Systems (Addison-Wesley, Massachusetts, 1977), chapters 1, 2 and 3.

40. Tjotta, J.N. and Tjotta, S., "An analytical model for the nearfield of a baffled piston transducer", J. Acoust. Soc. Am. 68(1), 334-339 (July 1980).

41. Baker, D.D., "Determination of Far-Field Characteristics of Large Underwater Sound Transducers from Near-Field Measurements", J. Acoust. Soc. Am. 34(11), 1737-1744 (Nov 1962).

42. Williams, E.G. and Maynard, J.D., "Numerical evaluation of the Rayleigh integral for planar radiators using the FFT", J. Acoust. Soc. Am. 72(6) 2020-2030 (Dec. 1982).

43. Yeager, D.M., "A computer simulation of acoustical image reconstruction", J. Acoust. Soc. Am. 73(1), 359-362 (Jan 1983).

44. Boyer, A.L. et al., "Reconstruction of ultrasonic images by backward projection", in Acoustical Holography, edited by A.F. Metherell, (Plenum Press, New York, 1971), vol 3, pp. 333-348.

APPENDICES

Page

APPENDIX A

Evaluation of the integral to obtain the farfield pressure using stationary phase methods

Equation (3.12) gives

$$P(x_1, x_2, x_3) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{ka\rho c\, \hat{V}(k_1, k_2, x_3'')}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} * \exp[-j(k_1 x_1 + k_2 x_2 + [(ka)^2 - k_1^2 - k_2^2](x_3 - x_3''))]dk_1 dk_2 \quad (A.1)$$

Let

$$x_1 = R\sin\theta\cos\Phi$$
$$x_2 = R\sin\theta\sin\Phi$$
$$(x_3 - x_3'') = R\cos\theta$$

Therefore

$$P(x_1, x_2, x_3) = \frac{ka\rho c}{(2\pi)^2} * \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\exp[-jR((ka)^2 - k_1^2 - k_2^2)^{1/2}\cos\theta + k_1\sin\theta\cos\Phi + k_2\sin\theta\sin\Phi]}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} * \hat{V}(k_1, k_2, x_3'')\, dk_1 dk_2 \quad (A.2)$$

Let

$$\Phi(k_1, k_2) = \hat{V}(k_1, k_2, x_3'')/((ka)^2 - k_1^2 - k_2^2)^{1/2} \quad (A.3)$$

$$\varphi(k_1, k_2) = -R[(ka)^2 - k_1^2 - k_2^2)^{1/2}\cos\theta + k_1\sin\theta\cos\Phi + k_2\sin\theta\sin\Phi] \quad (A.4)$$

The method of stationary phase is based on the assumption that the main contribution to the integral in the farfield is associated with the region in which the phase does not vary with the integration parameters, k_1 and k_2.

i.e.

$$\frac{d\varphi}{dk_1} = \frac{d\varphi}{dk_2} = 0$$

$$k_1 = \bar{k}_1$$

$$k_2 = \bar{k}_2 \qquad \text{at the point of stationary phase}$$

The stationary phase approximation to the double integral is (24, 25, 26)

$$I = \pm \frac{j\,2\pi}{|D(\bar{k}_1, \bar{k}_2)|^{1/2}} \Phi(\bar{k}_1, \bar{k}_2) \exp\,|\, j\,\varphi(\bar{k}_1, \bar{k}_2)\,| \qquad (A.5)$$

The alternate signs are associated with the corresponding sign of the determinant D, where D is given in eq. (A.6).

$$D(\bar{k}_1, \bar{k}_2) = \left(\frac{\delta^2\psi}{\delta k_1\,\delta k_2}\right)^2 - \frac{\delta^2\psi}{\delta k_1^2}\,\frac{\delta^2\psi}{\delta k_2^2} \qquad (A.6)$$

$$\frac{\delta\psi}{\delta k_1} = -\frac{1}{2} R\,[(ka)^2 - k_1^2 - k_2^2]^{-1/2}\,(-2k_1 \operatorname{Cos}\Theta) + R \operatorname{Sin}\Theta \operatorname{Cos}\Phi$$

$$= R\,[(ka)^2 - k_1^2 - k_2^2]^{-1/2}\,(k_1 \operatorname{Cos}\Theta) + R \operatorname{Sin}\Theta \operatorname{Cos}\Phi \qquad (A.7)$$

Therefore

$$\frac{\delta\psi}{\delta k_1} = R \operatorname{Cos}\Phi \operatorname{Sin}\Theta + \frac{k_1 R \operatorname{Cos}\Theta}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} \qquad (A.8)$$

Similarly,

$$\frac{\delta\psi}{\delta k_2} = R \operatorname{Sin}\Phi \operatorname{Sin}\Theta + \frac{k_2 R \operatorname{Cos}\Theta}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} \qquad (A.9)$$

The point of stationary phase is given by the above simultaneous roots:

i.e. where $\frac{\delta\psi}{\delta k_1} = \frac{\delta\psi}{\delta k_2} = 0$

i.e. $R\cos\Phi\sin\Theta + \frac{k_1 R\cos\Theta}{X} = R\sin\Phi\sin\Theta + \frac{k_2 R\cos\Theta}{X} = 0$

where

$$X = [(ka)^2 - k_1^2 - k_2^2]^{1/2}$$

Therefore

$$X\cos\Phi\sin\Theta + k_1\cos\Theta = X\sin\Phi\sin\Theta + k_2\cos\Theta = 0 \qquad \text{(A.10)}$$

Therefore

$$k_1\cos\Theta = -X\cos\Phi\sin\Theta$$

Hence

$$X = \frac{-k_1\cos\Theta}{\cos\Phi\sin\Theta} \qquad \text{(A.11)}$$

and now the RHS of Eq. (A.10)

$$X = \frac{-k_2\cos\Theta}{\sin\Phi\sin\Theta}$$

Therefore

$$\frac{k_1\cos\Theta}{\cos\Phi\sin\Theta} = \frac{k_2\cos\Theta}{\sin\Phi\sin\Theta}$$

So

$$k_1 = k_2\frac{\cos\Phi}{\sin\Phi}$$

Therefore

$$k_2 = k_1\tan\Phi \qquad \text{(A.12)}$$

Since spherical coordinates are being used:

$$k_1 = ka \sin\theta \cos\Phi$$

and

$$k_2 = ka \sin\theta \sin\Phi$$

So,

$$[(ka)^2 - k_1^2 - k_2^2)^{1/2} = [(ka)^2 - (ka)^2 \sin^2\theta \cos^2\Phi - (ka)^2 \sin^2\theta \sin^2\Phi]^{1/2}$$

$$= (ka)[1 - (\sin^2\theta \cos^2\Phi + \sin^2\theta \sin^2\Phi)]^{1/2}$$

$$= ka\,(1-\sin^2\theta)^{1/2}$$

$$= ka \cos\theta$$

So

$$\Phi(\bar{k}_1, \bar{k}_2) = \frac{\hat{V}(k_1, x_2, x_3'')}{ka \cos\theta} \qquad (A.13)$$

and

$$\psi(\bar{k}_1, \bar{k}_2) = -R\,[ka \cos^2\theta + ka \sin^2\theta \cos^2\Phi + ka \sin^2\theta \sin^2\Phi]$$

$$= -R\,ka\,[\cos^2\theta + \sin^2\theta\,(\cos^2\Phi + \sin^2\Phi)]$$

$$= -R\,ka\,[\cos^2\theta + \sin^2\theta]$$

Therefore

$$\psi(\bar{k}_1, \bar{k}_2) = -R\,ka \qquad (A.14)$$

Now, differentiating (A.8) with respect to k_2

$$\frac{\delta^2\psi}{\delta k_1\,\delta k_2} = -\frac{1}{2}(-2k_2)(k_1 R\cos\theta)((ka)^2 - k_1^2 - k_2^2)^{-3/2}$$

$$= \frac{k_1 k_2 R\cos\theta}{[(ka)^2 - k_1^2 - k_2^2]^{3/2}}$$

Substituting for k_1 and k_2

$$\frac{\delta^2\psi}{\delta k_1\,\delta k_2} = \frac{R(ka)^2 \sin^2\theta\cos\Phi\sin\Phi\cos\theta}{(ka\cos\theta)^3} \qquad (A.15)$$

Thus,

$$\frac{\delta^2\psi}{\delta k_1\,\delta k_2} = \frac{R}{(ka)}\tan^2\theta\cos\Phi\sin\Phi$$

Now, differentiating Eq. (A.8) with respect to k_1 gives

$$\frac{\delta^2\psi}{\delta k_1^2} = \frac{R\cos\theta}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} + \frac{k_1^2 R\cos\theta}{[(ka)^2 - k_1^2 - k_2^2]^{3/2}} = \frac{R}{(ka)}(1 + \tan^2\theta\cos^2\Phi) \qquad (A.16)$$

and differentiating Eq. (A.9) with respect to k_2 gives

$$\frac{\delta^2\psi}{\delta k_2^2} = \frac{R\cos\theta}{[(ka)^2 - k_1^2 - k_2^2]^{1/2}} + \frac{k_2^2 R\cos\theta}{[(ka)^2 - k_1^2 - k_2^2]^{3/2}} = \frac{R}{(ka)}(1 + \tan^2\theta\sin^2\Phi) \qquad (A.17)$$

So substituting equations (A.15, A.16, A.17) into Eq. (A.6) gives

$$D(\bar{k}_1, \bar{k}_2) = \left(\frac{R}{(ka)}\right)^2 \mathrm{Tan}^4\,\theta\,\mathrm{Cos}^2\,\Phi\,\mathrm{Sin}^2\,\Phi - \left(\frac{R}{(ka)}\right)^2 (1 + \mathrm{Tan}^4\theta\,\mathrm{Sin}^2\Phi\,\mathrm{Cos}^2\Phi + \mathrm{Tan}^2\,\theta)$$

$$= -\left(\frac{R}{(ka)}\right)^2 (1 + \mathrm{Tan}^2\theta)$$

and since

$$1 + \mathrm{Tan}^2\theta = \frac{1}{\mathrm{Cos}^2\theta}$$

$$D(\bar{k}_1, \bar{k}_2) = -\left(\frac{R}{ka\,\mathrm{Cos}\,\theta}\right)^2 \qquad \text{(A.18)}$$

Therefore substituting equations (A.13), (A.14) and (A.18) into Eq. (A.5) gives

$$I = \frac{j\,2\,\pi}{\left[\left(\frac{R}{ka\,\cos\theta}\right)^2\right]^{1/2}} * \frac{\hat{V}(k_1, k_2, x_3'')}{ka\,\cos\theta} * \exp|-j\,ka\,R|$$

Therefore

$$I = \frac{j\,2\,\pi\,\hat{V}(k_1, k_2, x_3'')}{R} \exp|-j\,ka\,R| \qquad \text{(A.19)}$$

The pressure is given by

$$P = \frac{ka\,\rho c}{(2\pi)^2} I$$

Therefore

$$P = \frac{ka\,\rho c}{(2\pi)^2} \frac{j\,2\pi\,\hat{V}(k_1, k_2, x_3'')\,\exp(-j\,ka\,R)}{R}$$

Therefore

$$P = j\,ka\,\rho c \frac{e^{(-j\,ka\,R)}}{2\pi R} \hat{V}(k_1, k_2, x_3'') \qquad \text{(A.20)}$$

So

$$P(R, \theta, \Phi) = j\,ka\,\rho c \left(\frac{e^{-j\,ka\,R}}{2\pi R}\right) \hat{V}(ka\,\sin\theta\,\cos\Phi, ka\,\sin\theta\,\sin\Phi, x_3'') \qquad \text{(A.21)}$$

APPENDIX B

FOURIER OPTICS

In this appendix, the equations for the optical equivalent to the acoustic projection techniques are derived. A lens is shown to display the spectrum of the light entering it. An example of a coherent-optical processor is given.

B.1 Propagation of plane waves

A unit amplitude plane with wave vector k is described by a complex amplitude

$$B(\underline{r}) = e^{-j\,\underline{k}\cdot\underline{r}} \qquad (B.1)$$

and corresponds to a real wave of the form

$$b(\underline{r}, t) = \text{Cos}\,(2\pi\nu t - \underline{k}\cdot\underline{r})$$

In a linear homogeneous medium the wave vector has magnitude $|\underline{k}| = k$ and its direction is in the direction of propagation of the wave.

Let

$$\underline{k} = k_x \underline{i}_x + k_y \underline{i}_y + k_z \underline{i}_z$$

where $\underline{i}_x$, $\underline{i}_y$, $\underline{i}_z$ are unit vectors parallel to the Cartesian coordinate axes. Now, a set of direction cosines α, β, γ can be defined with related angles $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$ according to

$$\alpha \equiv \text{Cos}\,\hat{\alpha} = (\underline{k}\cdot\underline{i}_x)/k = \frac{k_x}{k}$$

$$\beta \equiv \text{Cos}\,\hat{\beta} = (\underline{k}\cdot\underline{i}_y)/k = \frac{k_y}{k}$$

$$\gamma \equiv \text{Cos}\,\hat{\gamma} = (\underline{k}\cdot\underline{i}_z)/k = \frac{k_z}{k}$$

It can be seen from Figure B.1 that $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$ are the angles between $\underline{k}$ and the Cartesian unit vectors $\underline{i}_x$, $\underline{i}_y$ and $\underline{i}_z$ respectively.

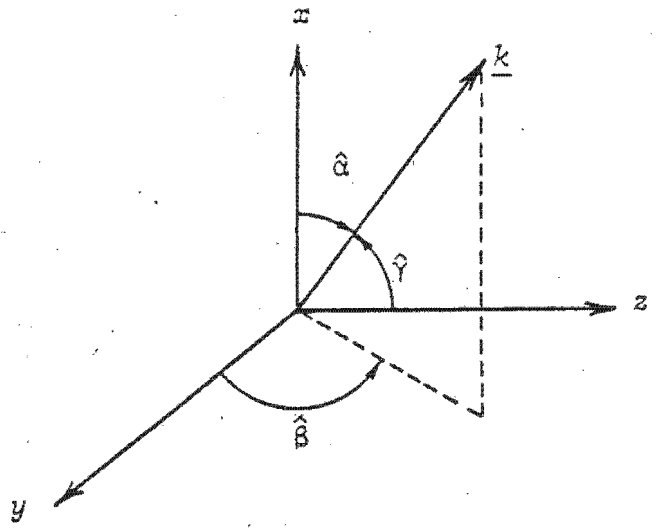


Figure B.1: Angles $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$.

Since

$$\frac{\underline{k} \cdot \underline{k}}{k^2} = 1, \quad \alpha^2 + \beta^2 + \gamma^2 = 1,$$

and because

$$\underline{r} = \underline{i}_x x + \underline{i}_y y + \underline{i}_z z,$$

Eq. (B.1) can be rewritten

$$B(x, y, z) = e^{-j(k_x x + k_y y + k_z z)} = e^{-jk(\alpha x + \beta y)} e^{-jk(1-\alpha^2-\beta^2)^{1/2} z} \qquad (B.2)$$

Now using the notation

$B_z(x,y) = B(x,y,z)$, we get

$$\frac{B_z(x,y)}{B_o(x,y)} = \exp\left[-j2\pi\left(\left(\frac{\alpha x}{\lambda} + \frac{\beta y}{\lambda}\right) + \frac{(1-\alpha^2-\beta^2)^{1/2}}{\lambda^2} - \left(\frac{\alpha x}{\lambda} + \frac{\beta y}{\lambda}\right)\right)\right]$$

$$= \exp\left[-j2\pi\left(\frac{1}{\lambda^2} - \left(\frac{\alpha}{\lambda}\right)^2 - \left(\frac{\beta}{\lambda}\right)^2\right)^{1/2} z\right] = H_z\left(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}\right) \qquad (B.3)$$

Substituting

$$u = \frac{\alpha}{\lambda},\ v = \frac{\beta}{\lambda},\ w = \frac{\gamma}{\lambda},$$

$$H_z(u,v) = \exp\left[-j2\pi\left(\frac{1}{\lambda}^2 - u^2 - v^2\right)^{1/2} z\right]$$

$$= \exp\left[-j\frac{2\pi}{\lambda}\left(1 - \lambda^2 u^2 - \lambda^2 v^2\right)^{1/2} z\right] \qquad (B.4)$$

where $H_z(u,v)$ is the free-space transfer function describing the propagation of a plane wave with direction cosines α, β, γ over distance z.

For propagation it is required that

$$u^2 + v^2 < \frac{1}{\lambda}^2$$

For $u^2 + v^2 > \frac{1}{\lambda}^2$, evanescence of the waves occurs.

Now that the free space transfer function has been derived, a wave travelling along the z-axis and incident on an x-y plane at z=0 as shown in Figure B.2, can be considered.

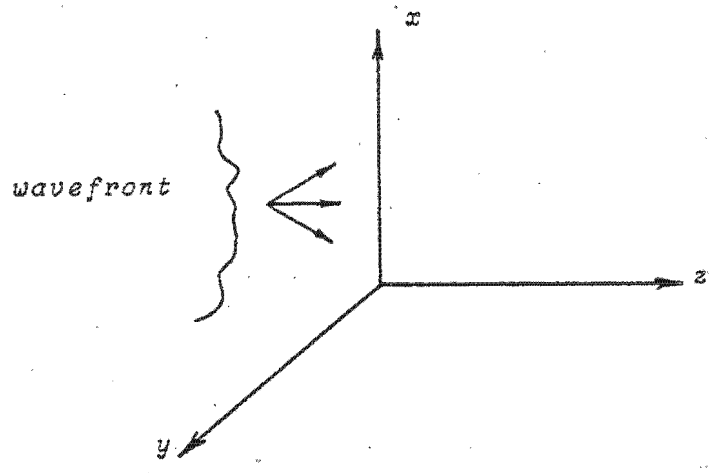


Figure B.2: A wave travelling along the z-axis.

Such a wave can be represented in terms of an infinite number of infinitesimal plane waves according to

$$U_o(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A_o(u,v)\, e^{-j(2\pi(ux + vy))}\, du\, dv \qquad (B.5)$$

where $U_o(x,y)$ is the complex amplitude across the x-y plane at z=0 and

$$dU_o \equiv A_o(u,v)\, du\, dv\, e^{-j2\pi(ux + vy)} \qquad (B.6)$$

represents a plane wave of infinitesimal amplitude $A_o(u,v)\, du\, dv$. The directional cosines of travel are given by relating Eq. (B.2) to Eq. (B.6),

$$\alpha = u\lambda,\ \beta = v\lambda,\ \gamma = (1 - (u\lambda)^2 - (v\lambda)^2)^{1/2}$$

Similarily, at a distance z away, the complex amplitude in a transversal xy-plane can be decomposed into infinitesimal plane waves

$$dU_z \equiv A_z(u,v)\, du\, dv \exp(-j2\pi(ux + uy)) \qquad (B.7)$$

according to

$$U_z(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A_z(u,v)\, e^{-j2\pi(ux + vy)}\, du\, dv \qquad (B.8)$$

Eq. (B.8) gives the distribution $U_z(x,y)$ being composed of the plane waves $A_z(u,v)\,du\,dv$.

From Eqs. (B.3) and (B.4) the relationship between dU_z and dU_o is obtained.

$$dU_z = H_z(u,v)\,dU_o$$

from which

$$A_z(u,v) = A_o(u,v)\exp\left[-j2\pi\left(1-\lambda^2u^2-\lambda^2v^2\right)^{1/2}z\right] \qquad (B.9)$$

Now to obtain the plane waves originating from z=0, the Fourier transform is used

$$A_o(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_o(x',y')\,e^{-j2\pi(ux'+vy')}\,dx'\,dy' \qquad (B.10)$$

where x' and y' are the co-ordinates for the plane at z=o.

By combining Eqs. (B.8), (B.9) and (B.10), a relationship between $U_z(x,y)$ and $U_o(x',y')$ can be obtained.

$$U_z(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_z(u,v)e^{-j2\pi(ux+vy)}\,du\,dv$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_o(u,v)\exp\left[-j2\pi\left(1-\lambda^2u^2-\lambda^2v^2\right)^{1/2}z\right]e^{-j2\pi(ux+vy)}\,du\,dv \qquad (B.11)$$

Assuming that the wave travelling from z=0 can be represented by its Fourier components, one gets

$$U_z = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} U_o(x', y')\, e^{-j\,2\pi\,(ux' + vy')}\, dx'\, dy' \right) *$$

$$\exp\left(-j\,\frac{2\pi}{\lambda}\,(1-\lambda^2 u^2 - \lambda^2 v^2)^{1/2}\, z\right)\; e^{-j\,2\pi\,(ux + vy)}\, du\, dv \qquad (B.12)$$

Eq. (B.11) can be rewritten by letting

$$k_1 = \frac{2\pi\alpha}{\lambda} = 2\pi u$$

$$k_2 = \frac{2\pi\beta}{\lambda} = 2\pi v$$

$$U_z(x, y) =$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A_o(u,v)\;\exp\;(-j(k^2 - k_1^2 - k_2^2)^{1/2}\, z)\;\exp\;(-j(k_1 x + k_2 y))\, dk_1\, dk_2 \qquad (B.13)$$

which is identical to Eq. (3.8) which was determined for the acoustic case.

B.2 Fourier transforming properties of lenses

The most important components of optical imaging and optical data-processing systems are lenses. The lens is composed of optically dense material, in which the propagation velocity of light is less than in air. If a ray enetering the lens at co-ordinates (x,y) on one face emerges at approximately the same co-ordinates on the opposite face, the lens is termed a thin lens. A thin lens simply delays an incident wavefront by an amount proportional to the thickness of the lens.

Goodman (27) gives the total phase delay undergone by the wave at co-ordinates (x,y) in passing through the lens as

$$\phi(x,y) = kn\Delta(x,y) + k(\Delta_o - \Delta(x,y)) \qquad (B.14)$$

where

n	is the refraction index of the lens material
Δ_o	is the maximum thickness of the lens
$kn\Delta(x,y)$	is the phase delay introduced by the lens
$k(\Delta_o - \Delta(x,y))$	is the phase delay introduced by the remaining region of free space between the two planes
$\Delta(x,y)$	is the thickness at co-ordinates (x,y) as shown in Figure B.3

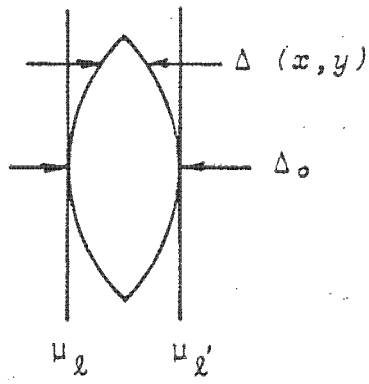


Figure B.3: The thickness function.

Thus the lens may be regarded as a multiplicative phase transformation of the form

$$t_\ell (x,y) = \exp [j k \Delta_o] \exp (j k (n-1) \Delta (x,y)) \qquad (B.15)$$

The complex field $U_\ell (x,y)$ can be obtained

$$U_\ell' (x,y) = t_\ell (x,y) U_\ell (x,y) \qquad (B.16)$$

The phase transformation can be found by considering the portions of the wavefront lying near the lens axis i.e. consider only paraxial rays.

Goodman (28) derives this to be

$$t_\ell (x,y) = \exp [j k n \Delta_o] \exp (- j \frac{k}{2f} (x^2 + y^2)) \qquad (B.17)$$

where f is the focal length of the lens.

Consider a unit amplitude plane wave incident on the lens. Using Eq. (B.16)

$$U_\ell' (x,y) = t_\ell (x,y) U_\ell (x,y)$$

Since the field distribution ($U_\ell (x,y)$) in front of the lens is unity,

$$U_\ell' (x,y) = \exp (j k n \Delta_o) \exp (-j \frac{k}{2f} (x^2 + y^2)) \qquad (B.18)$$

The first term of the above expression is a constant phase delay. The second term is a quadratic approximation to a spherical wave. Thus a lens composed of spherical surfaces maps a plane wave into a spherical wave.

Consider a plane object with amplitude transmittance $t_o(x,y)$ to be placed a distance d_o in front of a converging lens of focal length f, as shown in Figure B.4. The object is illuminated by a normally incident plane wave of amplitude A.

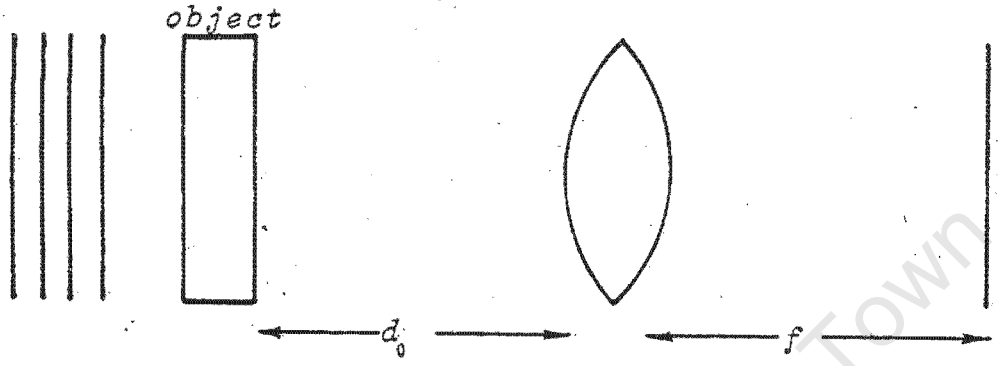


Figure B.4: Object placed in front of lens.

The resulting field at the focal plane behind the lens is given by Goodman (29):

$$U_f(x_f, y_f) = \frac{A \exp\left[j \frac{k}{2f} \left(1 - \frac{d_o}{f} \right) (x_f^2 + y_f^2) \right]}{j \lambda f} \; * \;$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} t_o(x_o, y_o) \exp\left(-j \frac{2\pi}{\lambda f} (x_o x_f + y_o y_f) \right) dx_o dy_o \qquad \text{(B.19)}$$

If the object is placed at the front focal length of the lens, Eq. (B.19) simplifies to

$$U_f(x_f, y_f) = \frac{A}{j \lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} t_o(x_o, y_o) \exp\left[-j \frac{2\pi}{\lambda f} (x_o x_f + y_o y_f) \right] dx_o dy_o \qquad \text{(B.20)}$$

which is an exact Fourier transform relation. So the single lens can be used to obtain the spectrum of light entering the lens. This is a very important property of a lens in that the spectrum is physically accessible and therefore can be manipulated simply by placing masks or optical filters in the Fourier transform plane.

B.3 The coherent-optical processor

Figure B.5 gives an example of the type of optical processor which can be set up to give similar results to those found in the acoustic case.

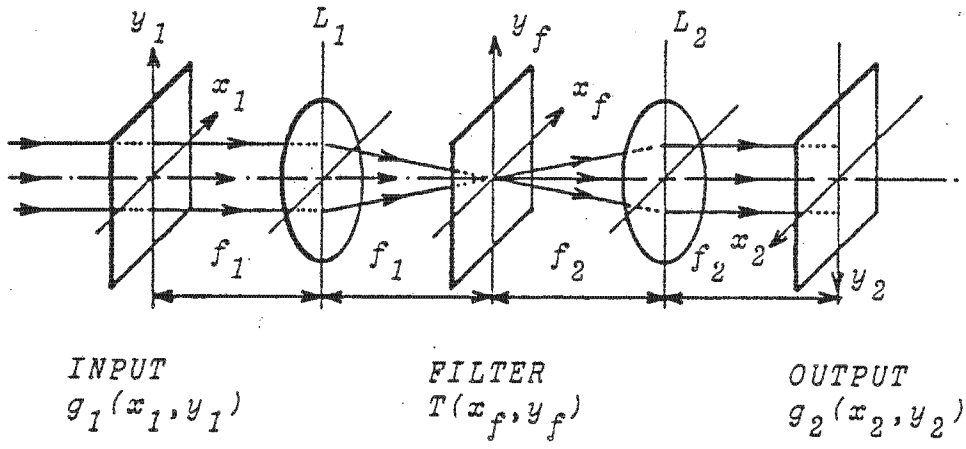


Figure B.5: Coherent-optical processor (after Almedia and Indebetouw (31)).

The block diagram for this processor is shown in Figure B.6.

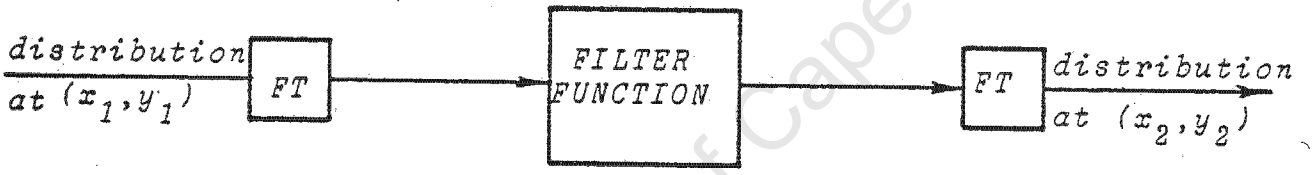


Figure B.6: Block diagram of optical-processor.

An input transparency of complex amplitude transmittance $g_1(x_1, y_1)$ is placed in the front focal plane of the lens L_1 and illuminated by a plane-parallel beam of uniform intensity and zero phase. The amplitude in the back focal plane P_f of the lens is represented by the Fourier transform $G_1(u,v)$ of the input, as given in Eq. (B.20).

Neglecting some proportionality factors

$$G_1(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_1(x_1,y_1)\, e^{-j2\pi (ux_1 + vy_1)} dx_1 dy_1 \qquad (B.21)$$

where λ is the wavelength

(u,v) are the rectangular spatial frequencies of the input. They are related to the coordinates (x_f, y_f) in the Fourier plane by

$$u = \frac{x_f}{\lambda f_1} \quad \text{and} \quad v = \frac{y_f}{\lambda f_1}$$

where f_1 is the focal length of lens L_1.

If a filter transparency of amplitude transmittance $T(x_f, y_f)$ is placed in the front focal plane P_f, the amplitude distribution just after the transparency is

$$U_f(u,v) = G_1(u,v) \cdot T(u,v) \tag{B.22}$$

The second lens, L_2, of the processor performs a second Fourier transform of $U_f(x_f,y_f)$, leading to an amplitude distribution in its back focal plane P_2 given by

$$g_2(x_2, y_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_1(u,v)\, T(u,v)\, e^{-j2\pi(ux_2 + vy_2)}\, du\, dv \tag{B.23}$$

This is a more general expression for the field at the plane (x_2, y_2) than developed in Eq. (B.13). In Eq. (B.13) it was assumed that the plane waves emanating from the origin could be found by Fourier transforming the field there, whereas in Eq. (B.23) one can experimentally find the plane waves by using a lens. If Eq. (B.23) is to be identical to Eq. (B.13), the filter at (x_f, y_f) must be removed, and the free space transfer function substituted for T (u,v).

Then Eq. (B.23) becomes

$$g_2(x_2, y_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_1(u,v) e^{-j\left[\frac{2\pi}{\lambda}(1-\lambda^2u^2-\lambda^2v^2)^{1/2}\right]z} e^{-j(k_1x_2+k_2y_2)}\, du\, dv \tag{B.24}$$

Now substituting $u = \frac{\alpha}{\lambda}$ and $v = \frac{\beta}{\lambda}$

and since $k_1 = \frac{2\pi\alpha}{\lambda}$ and $k_2 = \frac{2\pi\beta}{\lambda}$, Eq. (B.24) becomes

$$g_2(x_2, y_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_1(k_1, k_2)\, e^{-j(k^2-k_1^2-k_2^2)^{1/2}z}\, e^{-j(k_1x_2+k_2y_2)}\, du\, dv \tag{B.25}$$

In conclusion, it can be seen that the theory of optics is in good agreement with the theory of acoustics. In particular, it was found that the free space transfer function introduces a phase change. The lens can be used to physically obtain the two-dimensional spectrum from an incoming set of plane waves governed by the shape of an object placed at (x_1,y_1).

APPENDIX C

Program Listings

Listing C.1 gives the program for the HP-85 for data acquisition. This program requires the probe to be manually positioned at the top right hand corner of the grid of points to be scanned. When the CONT button is pressed, a grid of 32 x 32 points is measured. The program automatically positions the probe at the required position, so no manual intervention is required. The waveform at each of the 32 x 32 points is transferred from the Nicolete storage osilloscope to the HP-85.

Listings C.2 and C.3 give the integration program used to obtain two- and three-dimensional pressure patterns respectively. Listings C.4 and C.5 give the programs for the Fourier transform technique in two- and three-dimensions resepctively.

Listing C.6 gives the program for plotting the two-dimensional pressure or velocity patterns. The three-dimensional plots are produced using the THREED package on the VAX-VMS computer.

LISTING C.1

```
10 ! ************************************************************
20 ! Program to take measurements in tank over a square grid
30 ! by automatically stepping a probe
40 ! ************************************************************
50 !
60 OPTION BASE 1
70 !
80 ! --------------------
90 ! Define the variables
95 ! --------------------
96 !
130 DIM T$[61611] !                    Buffer
140 SHORT R1(32,32),I1(32,32) !    Real and imaginary parts
150 !
160 IOBUFFER T$ !                      Declare the buffer
170 !
180 ! ------------------------------------
190 ! Reset Nicolette and stepper motors
200 ! ------------------------------------
210 !
220 RESET 7 @ RESET 3
230 !
240 ! ----------------------
250 ! Set up stepper motors
260 ! ----------------------
270 !
280 CONTROL 3,5 ; 0@ CONTROL 3,4 ; 0@ CONTROL 3,3 ; 85
290 !
300 ! -------------------------
310 ! Set up Nicolette
320 ! =================
330 !
340 ! N0 - Standard numbers
350 ! E1 - CR
360 ! D0 - ASCII Auto Advance
370 ! H0 - Live; Hold next
380 ! -------------------------
390 !
400 OUTPUT 715 ;"N0"
410 CLEAR 7
420 CLEAR
430 !
440 ! ------------------------
450 ! Input normalizing set
460 ! A(2) : V-norm bit
470 ! A(3) : H-norm bit
480 ! A(4) : Memory fraction
490 ! A(5) : V-zero
500 ! A(6) : H-zero
510 ! A(7) : V-norm
520 ! A(8) : H-norm
530 ! ----------------------
540 !
550 ENTER 714 USING 560 ; A(2),A(3),A(4),A(5),A(6),A(7),A(8)
560 IMAGE D,D,D,X,DDDD,X,DDDD,D.De,D.De
570 DISP A(2);A(3);A(4);A(5);A(6);A(7);A(8)
580 OUTPUT 715 ;"E1D0"
590 OUTPUT 715 ;"H0"
```

```
600 !
610 ! ---------
620 ! Constants
630 ! ---------
640 !
650 F9=200 !                        Frequency in KHz
660 N=32 !                          Number of points on grid
670 S9=.5 !                         Sampling interval in fractions of a wavelength
680 !
690 DISP
700 DISP "Enter time for first sample"
720 DISP "Nicolette set for 1024 samples per waveform and"
730 DISP "triggers off -DC"
740 INPUT V5
750 V6=2*V5/(A(8)*1000000)*6
790 !
800 W7=1480/(F9*1000) !                      Wavelength in m
810 L1=W7*S9*1000 !                          Sampling interval in fractions
820 !                                        of a wavelength in mm
830 S=L1/.01 !                               Steps in 1 sample in mm
840 D1=L1*(N/2) !                            Distance of 1st point from centre of
850 !                                        transducer in mm
860 S1=D1/.01 !                              No of steps from 1st point to centre
870 !                                        of transducer in mm
880 DISP "Position the hydrophone"
890 DISP "so that both dimensions"
900 DISP "are at max positive"
901 DISP "Press CONT to continue" @ PAUSE
920 !
930 ! ------------------------------------------------------------------------------
940 ! Start the automatic data capture process
941 !
942 ! F2 is the dimension in which movement is to be made
943 !     = 1 : Move in X-dimension
944 !     = 4 : Move in Y-dimension
945 ! F1 is the direction of movement
946 !     = 2 : Move in positive X-direction or negative Y-direction
947 !     = 8 : Move in negative X-direction or positive Y-direction
948 ! F is the distance to be moved to the next position in the direction of
949 !   movement
950 ! F4 is the present position of the probe in the dimension other than the
951 !     direction of movement
960 !
961 ! ------------------------------------------------------------------------------
970 FOR I=1 TO N !                       Move in x-direction
980 FOR J=1 TO N-1 !                     Move in y-direction
990 J9=J
1000 IF FP (I/2)=0 THEN J9=N-J+1
1010 GOSUB 1990 !                    Measure
1020 F=(J+1)*S-S1 !                  Distance to next point in direction of
1021 !                               movement
1030 IF FP (I/2)<> 0 THEN F=S1-J*S
1040 F4=S1-(I-1)*S !                 Present position in x-direction
1050 F1=8 !                          Probe direction - positive in y-direction
1060 IF FP (I/2)<> 0 THEN F1=2 !     If I is odd, then probe direction is
1061 !                               negative
1070 F2=4 !                          Choose to move in y-direction
1080 GOSUB 1320 !                    Move probe to next position
1090 NEXT J
```

```
1100 J9=N
1110 IF FP (I/2)=0 THEN J9=1
1120 GOSUB 1990 !                          Measure
1130 F4=F !                                Present position in y-direction
1140 F=S1-I*S !                            Distance to next point
1150 F1=8 !                                Probe direction - negative in x-direction
1160 F2=1 !                                Choose to move in x-direction
1170 GOSUB 1320 !                          Move probe to next position
1180 NEXT I
1190 !
1200 ! ------------------------------------------------------
1210 ! Create data file, then output the readings to it
1220 ! ------------------------------------------------------
1230 !
1240 CREATE "DAT22",2,8448
1250 ASSIGN# 1 TO  "DAT22"
1260 PRINT# 1,1 ; R1()
1270 PRINT# 1,2 ; I1()
1280 ASSIGN# 1 TO  *
1290 STOP
1300 !
1310 ! --------------------------
1320 ! Move the probe by S steps
1330 ! --------------------------
1340 !
1350 FOR I3=1 TO S
1360 ASSERT 3;F1 @ ASSERT 3;F2
1370 WAIT 1
1380 NEXT I3
1390 ENTER 3 ; X,Y !          Enter coordinates of probe position
1400 W=Y
1410 IF F2=1 THEN W=X
1420 !
1430 ! ------------------------------------------------------------------------
1440 ! Move the probe to the required position by calculating the difference
1450 ! between the current position and the required position
1460 ! ------------------------------------------------------------------------
1470 !
1480 IF ABS (W)>ABS (F) THEN F1=10-F1 ! If probe past required position,
1490 !                                      then reverse
1500 L9=ABS (ABS (W)-ABS (F)) !            Difference between current position
1510 !                                      and required position
1520 FOR I3=1 TO L9 !                      Move the probe by L9 steps
1530 ASSERT 3;F1 @ ASSERT 3;F2
1540 WAIT 1
1550 NEXT I3
1560 ENTER 3 ; X,Y !          Enter coordinates of probe position
1570 W=Y !
1580 IF F2=1 THEN W=X !
1590 IF W=F THEN 1710 !       Correct position?
1600 !
1610 IF W>F AND F1=2 AND F2=1 THEN F1=8 ! \
1620 IF W<F AND F1=8 AND F2=1 THEN F1=2 !  \
1630 IF W>F AND F1=8 AND F2=4 THEN F1=2 !    Check the direction of movement
1640 IF W<F AND F1=2 AND F2=4 THEN F1=8 !  /
1650 ASSERT 3;F1 @ ASSERT 3;F2 !       Step the probe in fine increments until
1660 !                                  it is at the correct position
```

```
1690 !
1700 ! ------------------------------------------------------
1710 ! Position the probe correctly in the other direction
1720 ! ------------------------------------------------------
1730 !
1740 ENTER 3 ; X,Y !          Enter coordinates of probe position
1750 W=Y
1760 IF F2=4 THEN W=X
1770 IF W=F4 THEN 1930 !      Correct position?
1780 D=8
1790 IF W>F4 AND F2=1 THEN D=10-D
1800 IF W<F4 AND F2=4 THEN D=10-D
1810 ASSERT 3;D @ ASSERT 3;(5-F2)
1820 ENTER 3 ; X,Y
1830 W=Y
1840 IF F2=4 THEN W=X
1850 IF W=F4 THEN 1930
1860 IF W<F4 AND D=2 AND F2=1 THEN D=8 ! \
1870 IF W>F4 AND D=8 AND F2=1 THEN D=2 !  \
1880 IF W<F4 AND D=8 AND F2=4 THEN D=2 !   Check the direction of movement
1890 IF W>F4 AND D=2 AND F2=4 THEN D=8 !  /
1900 ASSERT 3;D @ ASSERT 3;(5-F2) !   Step the probe in fine increments until
1910 !                                it is at the correct position
1920 GOTO 1820
1930 RETURN
1940 !
1950 ! ------------------------------------------------
1960 ! Measure the magnitude and phase of the pressure
1970 ! ------------------------------------------------
1980 !
1990 OUTPUT 715 ;"HO" !          Nicolette : Live, hold next
2000 SEND 7 ; LISTEN 14 !        Nicolette : The listener
2010 TRANSFER 714 TO  T$ FHS ! Transfer waveform to buffer
2020 M5=0
2030 BEEP !                      Signal end of transfer
2040 !
2050 OUTPUT 715 ;"LO" !          Nicolette : Live
2060 CLEAR 7
2070 S7=0
2080 FOR J4=V6+1 TO V6+601 STEP 12 ! \
2090 G=VAL (T$[J4,J4+4])*A(7) !       \
2100 S7=S7+G !                          Sum of 51 real values
2110 NEXT J4 !                        /
2120 S8=0
2130 FOR J4=V6+7 TO V6+607 STEP 12 ! \
2140 G=VAL (T$[J4,J4+4])*A(7) !       \
2150 S8=S8+G !                          Sum of 51 imaginary values
2160 NEXT J4 !                        /
2170 S7=S7/51 @ S8=S8/51 !              Average of real and imaginary parts
2180 ENTER 3 ; X,Y !                    Read the coordinates of the probe
2190 R1(J9,I-01)=S7 @ I1(J9,I-01)=S8 !  Store the real and imaginary parts
```

```
2200 DEG
2210 M6=SQR (S7^2+S8^2) @ P6=ATN (S8/S7) ! Calculate the magnitude and phase
2220 RAD
2230 IF S7<0 AND S8<0 THEN P6=P6-180
2240 IF S7<0 AND S8>0 THEN P6=P6+180
2250 !
2260 ! Print the probe position and the magnitude and phase of the pressure
2270 !
2280 PRINT USING 2290 ; .01*X/(W7*1000),.01*Y/(W7*1000),M6,P6
2290 IMAGE 2(DDD.DD,1X),2(DDDD.DD,2X)
2300 !
2310 ! Clear buffer
2320 !
2330 CONTROL T$,0 ; 1,0 !          Clear the buffer
2340 RETURN
2350 END
```

LISTING C.2

```
C ***************************************************************************
C *                                                                         *
C * Program to calculate the pressure distribution in two-                  *
C * dimensions using the integration technique.                             *
C *                                                                         *
C * Date      : 5/11/84                                                     *
C *                                                                         *
C * Author    : E.Yudelman                                                  *
C *                                                                         *
C * Inputs    : Wave - distance from source plane to required plane         *
C *                                                                         *
C * Outputs   : File INTDAT.DAT - contains the data to be plotted           *
C *             using the PLOT program in listing 8.4                       *
C *             This file can also be used if the integration data          *
C *             is to be used by the Fourier technique                      *
C *                                                                         *
C ***************************************************************************

C ---------------------------------------------------------------------------
C                      Define the variables
C ---------------------------------------------------------------------------

      Character*6 port
      Real zmag(128),phas(128)
      Complex*16 xr(128)
      N=128

      Type *,'enter distance away'
      Accept *,wave

C ---------------------------------------------------------------------------
C    Calculate the sum of the left hand half of the plane
C    using equation 6.4.
C    Summations from line arrays are calculated from -40 wavelengths
C    to 40 wavelengths in the transverse direction in steps
C    of .1 wavelength.
C
C    Since the results are symmetrical, the data in the right half
C    plane is found from the data calculated in the left half plane
C
C    K1 is a counter
C ---------------------------------------------------------------------------

      K1=1
      Do 100 zno=-floatj(n/4),0.,.5
         Re1=0.
         Zim1=0.
         Do 50 dst=-40.,40.,.1
            Do 20 src=-5.,5.,.5
               Dist=sqrt(wave*wave+(zno-src)*(zno-src)+dst*dst)
               Phase=dist*360.
               Zmagnitude=1.
               If (src.eq.-5..or.src.eq.5.) zmagnitude=.5
               Re1=re1-zmagnitude*sind(-phase)/dist
               Zim1=zim1+zmagnitude*cosd(-phase)/dist
20          Continue
50       Continue
         Xr(k1)=dcmplx(re1,zim1)
         K1=k1+1
100   Continue
```

```
C ------------------------------------------------------------------
C     Open file for 2-d plotting and write to it
C ------------------------------------------------------------------

      Open (unit=20,status='new',file='intdat.dat')
      Write (20,*) xr
      Close (unit=20)

      Stop
      End
```

```
                    If (xl.eq.-5..or.xl.eq.5..or.yl.eq.-5..or.
     *               yl.eq.5.) zmagnitude=.5
                    Real=real-zmagnitude*sind(-phase)/dist
                    Zimag=zimag+zmagnitude*cosd(-phase)/dist
40                Continue
60              Continue
                Zmag(i,j)=sqrt(real*real+zimag*zimag)
                Xr(i,j)=dcmplx(real,zimag)
                I=i+1
80            Continue

C ---------------------------------------------------------------------
C               Copy from bottom left hand sector to top left hand sector
C ---------------------------------------------------------------------

            Do 303 kj=(npoints/2)-2,npoints
               Zmag(kj,j)=zmag(npoints+2-kj,j)
               Xr(kj,j)=xr(npoints+2-kj,j)
303         Continue
            J=j+1
100       Continue

C ---------------------------------------------------------------------
C               Copy from left hand plane to right hand plane
C ---------------------------------------------------------------------

          Do 709 ij=1,npoints
             Do 708 kj=(npoints/2)-2,npoints
                Zmag(ij,kj)=zmag(ij,npoints+2-kj)
                Xr(ij,kj)=xr(ij,npoints+2-kj)
708          Continue
709       Continue

C ---------------------------------------------------------------------
C               Open file for 3-d plotting and write to it
C ---------------------------------------------------------------------

          Open (unit=20,status='new',file='intdat3d.dat')
          Write(20,990)'a',1
990       Format(a1,i4)
          Write(20,991)'b',1,1,0,0,npoints,npoints,0,0,0
991       Format(a1,i4,8(i5))
          Write(20,992)'c','(8(F8.2,2x))'
992       Format(a1,a12)
          Write (20,999) ((zmag(ij,kj),ij=1,npoints),kj=1,npoints)
999       Format(8(f8.2,2x))
          Write(20,993)'f',0.,0.,-10.,10.,10.,15.,5.,21.,0.5
993       Format(a1,f9.0,6(f10.0),2(f5.0))
          Write(20,994)'g',1
994       Format(a1,i4)
          Write(20,995)'h',1,30.,60.,3.
995       Format(a1,i9,3(f10.0))
          Write(20,996)'i',title
996       Format(a1,a50)
          Close (unit=20)

C ---------------------------------------------------------------------
C          Inquire if data to be stored for later use
C ---------------------------------------------------------------------

          Type *,'Store data for later use ? (1=yes,0=no)'
```

```
Accept *,ilat
If (ilat.eq.1) then
   Open (unit=20,status='new',file='dat.dat')
   Write (20,*) xr
   Close (unit=20)
End if

Stop
End
```

LISTING C.4

```
C ***************************************************************
C *                                                             *
C * Program to calculate pressure or velocity distributions     *
C * in two-dimensions using the Fourier technique.              *
C *                                                             *
C *                                                             *
C * Date     : 5/11/84                                          *
C *                                                             *
C * Author   : E.Yudelman                                       *
C *                                                             *
C * Inputs   : Wave - distance between source plane and         *
C *                   required planes                           *
C *            Ifile - true if external file to be read         *
C *            Filename - external file if ifile is true        *
C *            Ipresvel - true if pressure to velocity          *
C *                       conversion is required                *
C *            Ivelpres - true if velocity to pressure          *
C *                       conversion is required                *
C *                                                             *
C * Outputs  : File FOURDAT.DAT - contains the data to be       *
C *            plotted via PLOT.FOR                             *
C *                                                             *
C ***************************************************************

C -------------------------------------------------------------
C               Define the variables
C -------------------------------------------------------------

      Complex*16 xr(128),cwk(128),temp,va,t2
      Real rwk(918),v1
      Real*8 counti,countj
      Character*15 outfile
      Integer n,iwk(918),iques
      Common xr

      N=128                             ! Number of points
      L1=10                             ! Size of source
      Type *,'enter distance away'
      Accept *,wave
      S=.5                              ! Sampling interval

C -------------------------------------------------------------
C      Form the input data by reading a data file,
C                        or generating the data
C -------------------------------------------------------------

      Type *,'read a file (1=yes,0=no)'
      Accept *,ifile
      If (ifile.eq.1) then


C ----- Input filename, then read in data from file -----

         Type *,'Input file name to be read'
         Read (*,405) filename
405      Format (a15)
         Open (unit=20,status='old',file=filename)
         Read (20,*) xr
         Close (unit=20)
      Else
```

```
C ----- Generate data -----

            Do 10 j=1,n
               Istart=int((i-int(i1/2))/s)
               iend=int((i+int(i1/2))/s)+1
               If (j.le.istart.or.j.gt.iend) then
                  Xr(j)=(0.,0.)
               Else if (j.eq.istart+1.or.j.eq.iend) then
                  Xr(j)=(0.5,0.)
               Else
                  Xr(j)=(1.0,0.0)
               End if
10          Continue
         End if

C ---------------------------------------------------------------
C          Swop the data, then perform FFT
C ---------------------------------------------------------------

         Call swop(n)
         Call FFT3D(xr,n,1,n,1,1,1,iwk,rwk,cwk)

C ----- Normalize the data returned from the FFT -----

         Do 670 j=1,n
            xr(j)=cmplx(xr(j)/floatj(n))
670      Continue


*        Type *,'Do you want pressure to velocity conversion? (1=yes,0=no) '
         Accept *,ipresvel
         If (ipresvel.eq.1) then

C ---------------------------------------------------------------
C          Convert pressure data to velocity data
C ---------------------------------------------------------------

            Type *,'          -- pressure to velocity conversion -- '
            Do 229 j=0,(n/2)-1
               Countj=floatj(J)/(floatj(n/2)-1.)
               If (floatj(J).gt.floatj(n/2)-1.5) then
                  Countj=(floatj(n/2)-1.5)/(floatj(N/2)-1.)
               End if
               V1v2=countj**2.
               Va=cdsqrt(dcmplx(1.-v1v2))
               Xr(j+1)=xr(j+1)*va
               If (j.ne.0) then
                  Xr(n-j+1)=xr(n-j+1)*va
               End if
229         Continue
         End if

         Type *,'Do you want velocity to pressure conversion? (1=yes,0=no) '
         Accept *,ivelpres
         If (ivelpres.eq.1) then
```

```
C ------------------------------------------------------------------
C         Convert velocity data to pressure data
C ------------------------------------------------------------------

            Type *,'        -- velocity to pressure conversion -- '
            Do 239 j=0,(n/2)-1
               Countj=floatj(J)/(floatj(n/2)-1.)
               If (floatj(J).gt.floatj(n/2)-1.5) then
                  Countj=(floatj(n/2)-1.5)/(floatj(N/2)-1.)
               End if
               V1v2=countj**2.
               Va=cdsqrt(dcmplx(1.-v1v2))
               Xr(j+1)=xr(j+1)/va
               If (j.ne.0) then
                  Xr(n-j+1)=xr(n-j+1)/va
               End if
239         Continue
         End if

C ------------------------------------------------------------------
C       Modify by the phase factor
C ------------------------------------------------------------------

         V1=wave*2.*3.1415
         Do 304 j=0,(n/2)-1
            Countj=floatj(J)/(floatj(n/2)-1.)
            V1v2=countj**2.
            Va=cmplx(V1)*cdsqrt(dcmplx(1.-v1v2))
            Temp=dcmplx(cos(dreal(-va)),sin(dreal(-va)))
            Xr(j+1)=xr(j+1)*temp
            If (j.ne.0) then
               Xr(n-j+1)=xr(n-j+1)*temp
            End if
304      Continue

C ------------------------------------------------------------------
C         Perform inverse FFT followed by swop to restore order
C ------------------------------------------------------------------

         Do 105 j=1,n
            Xr(j)=dconjg(xr(j))
105      Continue
         Call FFT3D(xr,n,1,n,1,1,1,iwk,rwk,cwk)
         Do 106 j=1,n
            Xr(j)=dconjg(xr(j))
106      Continue
         Call swop(N)

C ----- Normalize the data returned from the FFT -----

         Do 305 j=1,n
            Xr(J)=cmplx(xr(j)/float(n))
305      Continue
         Open (unit=20,status='new',file='fourdat.dat')
         Write (20,*) xr
         Close (unit=20)
         Stop
         End
```

```
C ***************************************************************
C *                                                             *
C *       Subroutine to swop the data                           *
C *                                                             *
C *       Date    : 5/11/84                                     *
C *                                                             *
C *       Author  : E.Yudelman                                  *
C *                                                             *
C *       Inputs  : One dimensional array to be swopped.        *
C *                 Size is 128                                 *
C *                                                             *
C *       Output  : Same array as input                         *
C *                                                             *
C ***************************************************************

      Subroutine swop(n)
      Complex*16 xr(128)
      Common xr
      Complex temp

C ---------------------------------------------------------------
C               Save the old value, then swop the data
C ---------------------------------------------------------------

      Do 20 j=1,n/2
         Temp=cmplx(xr(j))
         Xr(J)=xr(int(n/2+j))
         Xr(int(n/2+j))=temp
20    Continue
      Return
      End
```

LISTING C.5

```
C ***************************************************************
C *                                                             *
C * Program to calculate pressure or velocity distributions     *
C * in three-dimensions using the Fourier technique.            *
C *                                                             *
C *                                                             *
C * Date    : 5/11/84                                           *
C *                                                             *
C * Author  : E.Yudelman                                        *
C *                                                             *
C * Inputs  : Ititle - 50 character title of 3-d graph          *
C *           Ifile  - true if external file to be read         *
C *           Filename - external file if ifile is true         *
C *           Nplot - number of points to be plotted in both    *
C *                   x and y directions.                       *
C *           Wave - distance from source plane to required     *
C *                  plane                                      *
C *           Ipresvel - true if pressure to velocity           *
C *                      conversion is required                 *
C *           Ivelpres - true if velocity to pressure           *
C *                      conversion is required                 *
C *                                                             *
C * Outputs : File DATPLOT.DAT - contains the data to be        *
C *           plotted via the THREED graphics package           *
C *                                                             *
C ***************************************************************

C ---------------------------------------------------------------
C                  Define the variables
C ---------------------------------------------------------------

      Complex*16 xr(128,128),cwk(128),temp,va
      Complex tem(128,128)
      Real rwk(918),phase(128),v1,zmag(128)
      Real re1(32,32),zim1(32,32)
      Real*8 counti,countj
      Integer n,iwk(918)
      Common xr
      Character*6 port
      Character*50 title
      Character*15 filename

      Type *,'Input title of graph'
      Read (*,923) title
923   Format(a50)
      Port='_ttc4:'

C ---------------------------------------------------------------
C       If file to be read, input the filename
C ---------------------------------------------------------------

      Type *,'file to be read ? (1=yes,0=no)'
      Accept *,ifile
      If (ifile.eq.1) then
         Type *,'Input file to be read'
         Read (*,925) filename
925      Format(a15)
      End if

      Type *,'Enter number of points to be plotted'
      Accept *,nplot
```

```
      N=128                                ! Number of points to be used
      L1=10                                ! Length of source
      Type *,'Enter distance away'
      Accept *,wave
      S=.5                                 ! Sampling interval
      L=(n/2)*S

C -------------------------------------------------------------
C Form the input data - if simulation then generate the data
C                       if measured data then reshuffle data
C -------------------------------------------------------------


      If (Ifile.eq.1) then

C -----  Measured data so read it in then reshuffle it to allow -----
C -----                   for augmenting zeros                   -----

         Read (20,*) tem
         Open (unit=20,status='old',file=filename)
         Close (unit=20)
         Do 333 i=1,32
            Do 223 j=1,32
               Xr(i,j)=dcmplx(tem(i,j))
223         Continue
333      Continue
         Do 122 j=1,128
            Do 123 I=1,128
               If (i.lt.49.or.i.gt.80
     *           .or.j.lt.49.or.j.gt.80) then
                  Xr(I,j)=dmplx(0.0,0.0)
               Else
                  Xr(I,j)=dmplx(rel(I-48,j-48),zim1(I-48,j-48))
               End If
123         Continue
122      Continue
      Else

C   ----- Simulation, so generate data -----

         Do 15 I=1,n
            Do 10 j=1,n
               Istart=int((I-int(l1/2))/s)
               Iend=int((I+int(l1/2))/s)+1
               If (j.le.istart.or.j.gt.iend.or.i.le.istart
     *              .or.i.gt.iend) then
                  Xr(i,j)=(0.,0.)
               Else if (j.eq.istart+1.or.j.eq.iend.or.i.eq
     *              .istart+1.or.i.eq.iend) then
                  Xr(i,j)=(.5,0.)
               Else
                  Xr(i,j)=(1.0,0.0)
               End If
10          Continue
15       Continue
      End If
```

```
C ------------------------------------------------------------
C     Swop the data, then perform FFT
C ------------------------------------------------------------


        Call swop(n)
        Call FFT3D(xr,128,128,128,128,1,1,iwk,rwk,cwk)
        Type *,'     *** First FFT completed ***     '

C  ----- Normalize the data returned from the FFT -----

        Do 660 j=1,n
           Do 670 i=1,n
              Xr(i,j)= cmplx(xr(i,j)/floatj(n))
 670       Continue
 660    Continue
        Type *,'Do you want pressure to velocity(1=yes,0=no)'
        Accept *,ipresvel
        If (ipresvel.eq.1) then

C ------------------------------------------------------------
C     Convert pressure data to velocity data
C ------------------------------------------------------------


        Type *,'     -- pressure to velocity conversion -- '
        Do 229 j=0,63
           Do 20 i=0,63
              Counti=floatj(i)/63.
              Countj=floatj(j)/63.
              If (floatj(i).gt.62.5) then
                 Counti=62.5/63.
              End if
              If (floatj(j).gt.62.5) then
                  Countj=62.5/63.
              End if
              V1v2=counti**2.+countj**2.
              Va=cdsqrt(dcmplx(1.-v1V2))
              Xr(j+1,i+1)=xr(j+1,i+1)*va
              If (i.ne.0) then
                Xr(j+1,n-i+1)=xr(j+1,n-i+1)*va
              End if
              If (j.ne.0) then
                Xr(n-j+1,i+1)=xr(n-j+1,i+1)*va
              End if
              If (i.ne.0.and.j.ne.0) then
                Xr(n-j+1,n-i+1)=xr(n-j+1,n-i+1)*va
              End if
 20        Continue
 229    Continue
        End if

        Type *,'Velocity to pressure ?(1=yes,0=no)'
        Accept *,ivelpres
        If (ivelpres.eq.1) then
```

```
C ------------------------------------------------------------------
C     Convert velocity data to pressure data
C ------------------------------------------------------------------

      Type *,'     -- velocity to pressure conversion -- '
      Do 556 j=0,63
         Do 76 i=0,63
            Counti=floatj(i)/63.
            Countj=floatj(j)/63.
            If (floatj(i).gt.62.5) then
                Counti=62.5/63.
            End if
            If (floatj(j).gt.62.5) then
                 Countj=62.5/63.
            End if
            V1v2=counti**2.+countj**2.
            Va=cdsqrt(dcmplx(1.-v1V2))
            Xr(j+1,i+1)=xr(j+1,i+1)/va
           *If (i.ne.0) then
              Xr(j+1,n-i+1)=xr(j+1,n-i+1)/va
            End if
            If (j.ne.0) then
              Xr(n-j+1,i+1)=xr(n-j+1,i+1)/va
            End if
            If (i.ne.0.anD.j.ne.0) then
              Xr(n-j+1,n-i+1)=xr(n-j+1,n-i+1)/va
            End if
76       Continue
556   Continue
      End if

C ------------------------------------------------------------------
C     Modify by the phase factor
C ------------------------------------------------------------------

212   V1=wave*2.*3.1415
      Do 304 j=0,63
         Do 30 i=0,63
            Counti=floatj(i)/63.
            Countj=floatj(j)/63.
            V1v2=counti**2.+countj**2.
            Va=cmplx(v1)*cdsqrt(dcmplx(1.-v1V2))
            If (V1v2.lT.1.) then
               Temp=dcmplx(cos(dreal(-va)),sin(dreal(-va)))
            Else
               Temp=dcmplx(exp(-abs(dimag(Va))))
            End if
            Xr(i+1,j+1)=xr(i+1,j+1)*temp
            If (i.ne.0) then
              Xr(i+1,n-j+1)=xr(i+1,n-j+1)*temp
            End if
            If (j.ne.0) then
              Xr(n-i+1,j+1)=xr(n-i+1,j+1)*temp
            End if
            If (i.ne.0.and.j.ne.0) then
              Xr(n-i+1,n-j+1)=xr(n-i+1,n-j+1)*temp
            End if
30       Continue
304   Continue
```

```
C ------------------------------------------------------------
C       Perform inverse FFT followed by swop to restore order
C ------------------------------------------------------------

         Do 225 j=1,128
            Do 226 i=1,128
               Xr(i,j)=dconjg(xr(i,j))
226         Continue
225      Continue
         Call FFT3D(xr,128,128,128,128,1,1,iwk,rwk,cwk)
         Type *,'   *** Second ffT completed ***     '
         Do 222 j=1,128
            Do 255 i=1,128
               Xr(i,j)=dconjg(xr(i,j))
255         Continue
222      Continue
1090     Call swop(n)

C ------------------------------------------------------------
C    Open file for 3-d plotting and write to it
C ------------------------------------------------------------

         Open (unit=21,status='new',file='plotdat.dat')
         Write(21,990)'a',1
990      Format(a1,i4)
         Write(21,991)'b',1,1,0,0,nplot,nplot,0,0,0
991      Format(a1,i4,8(i5))
         Write(21,992)'c','(8(f8.2,2X))'
992      Format(a1,a12)
         Do 400 j=65-nplot/2,64+nplot/2
            Do 300 i=65-nplot/2,64+nplot/2
               Xr(i,j)=xr(i,j)/(float(n))
               Re=dreal(xr(i,j))
               Zim=dimag(xr(i,j))
               Zmag(i)=sqrt(re**2.+zim**2.)
300         Continue
            Write(21,999)(zmag(ij),ij=65-nplot/2,64+nplot/2)
999         Format(8(f8.2,2X))
400      Continue
         Write(21,993)'f',0.,0.,-10.,10.,10.,15.,5.,21.,0.5
993      Format(a1,f9.0,6(f10.0),2(f5.0))
         Write(21,994)'g',1
994      Format(a1,i4)
         Write(21,995)'h',1,30.,60.,3.
995      Format(a1,i9,6(f10.0),i10)
         Write(21,996)'i',title
996      Format(a1,a50)
         Close (unit=21)
         Stop
         End
```

```
C ***************************************************************
C *                                                             *
C *    Subroutine to swop the data                              *
C *                                                             *
C *    Date     : 5/11/84                                       *
C *                                                             *
C *    Author   : E.Yudelman                                    *
C *                                                             *
C *    Inputs   : Two dimensional array to be swopped.          *
C *               Size is 128 X 128                             *
C *                                                             *
C *    Output   : Same array as input                           *
C *                                                             *
C ***************************************************************

          Subroutine swop(n)
          Complex*16 xr(128,128)
          Common xr
          Complex temp

C ---------------------------------------------------------------
C     Save the old value, then swop the data for the rows
C ---------------------------------------------------------------

          Do 20 j=1,n
             Do 10 i=1,n/2
                Temp=cmplx(xr(j,i))
                Xr(j,i)=xr(j,int(n/2+i))
                Xr(j,int(n/2+i))=temp
10           Continue
20        Continue

C ---------------------------------------------------------------
C     Save the old value, then swop the data for the columns
C ---------------------------------------------------------------

          Do 40 j=1,n
             Do 30 i=1,n/2
                Temp=cmplx(xr(i,j))
                Xr(i,j)=cmplx(xr(int(n/2+i),j))
                Xr(int(n/2+i),j)=cmplx(temp)
30           Continue
40        Continue
          Return
          End
```

LISTING C.6

```
C ***************************************************************
C *                                                             *
C * Program to plot the two-dimensional data.                   *
C * Compares the results from the integration and Fourier       *
C * techniques.                                                 *
C *                                                             *
C * Date    : 5/11/84                                           *
C *                                                             *
C * Author  : E.Yudelman                                        *
C *                                                             *
C * Inputs  : Integration data filename - intfile               *
C *           Fourier data filename     - fourfile              *
C *                                                             *
C * Outputs : Graphs comparing the two techniques               *
C *                                                             *
C ***************************************************************

C -------------------------------------------------------------
C       Define the variables
C -------------------------------------------------------------

        Real zmag(128),phas(128)
        Complex*16 xr(128)
        Character*6 port
        Character*15 intfile
        Character*15 fourfile

C -------------------------------------------------------------
C       Input the integration and Fourier filenames
C -------------------------------------------------------------

        Type *,'Input integration file name'
        Read (*,877) intfile
        Type *,'Input Fourier file name'
        Read (*,877) fourfile
877     Format(a15)
        N=128                       ! Number of points

C -------------------------------------------------------------
C       Set port to device to where plotting will take place
C -------------------------------------------------------------

        Port='_ttc4:'

C -------------------------------------------------------------
C       Initialise graphics
C -------------------------------------------------------------

        Call grafic(2,port)

C -------------------------------------------------------------
C       Set up left lower screen for phase of Fourier data
C -------------------------------------------------------------

        Call posn(1,40,450,40,300)
        Call drwaxes(0.,-32.,32.,4.,0.,-180.,180.,45.,1)
        Call alpsiz(5.0)
        Call axnum(1,2,0,1,1,0,1)
        Call tcurs(150,340)
        Call alpsiz(3.)
        Call carwrt('GRAPH OF PHASE VS. TRANSVERSE DISTANCE')
```

```
      Call poslab(1,7)
      Call alpsiz(2.5)
      Call carwrt('FOURIER')
      Call tcurs(22,10)
      Call alpsiz(2.5)
      Call carwrt('TRANSVERSE DISTANCE (WAVELENGTHS)')
      Call rrotat(90.)
      Call tcurs(25,90)
      Call alpsiz(2.5)
      Call carwrt('PHASE IN DEGREES')
      Call rrotat(0.)

C -------------------------------------------------------------
C       Set up right lower screen for phase of integration data
C -------------------------------------------------------------

      Call posn(2,500,910,40,300)
      Call drwaxes(0.,-32.,32.,4.,0.,-180.,180.,45.,2)
      Call alpsiz(10.0)
      Call axnum(2,2,0,1,1,0,1)
      Call poslab(2,11)
      Call alpsiz(2.5)
      Call carwrt('INTEGRATION')
      Call tcurs(490,10)
      Call alpsiz(2.5)
      Call carwrt('TRANSVERSE DISTANCE (WAVELENGTHS)')
      Call rrotat(90.)
      Call tcurs(490,90)
      Call carwrt('PHASE IN DEGREES')
      Call rrotat(0.)

C -------------------------------------------------------------
C       Set up upper half of screen for magnitude
C -------------------------------------------------------------

      Call posn(3,40,750,400,690)
      Call drwaxes(0.,-32.,32.,4.,0.,0.,1.,.2,3)
      Call axnum(3,2,0,1,1,1,1)
      Call tcurs(50,700)
      Call alpsiz(3.)
      Call carwrt('GRAPH OF MAGNITUDE VS. TRANSVERSE DISTANCE')
      Call tcurs(160,380)
      Call alpsiz(2.5)
      Call carwrt('TRANSVERSE DISTANCE (WAVELENGTHS)')
      Call rrotat(90.)
      Call tcurs(25,450)
      Call carwrt('NORMALIZED MAGNITUDE')
      Call rrotat(0.)

C -------------------------------------------------------------
C       Data on frequency, size of source and distance away
C -------------------------------------------------------------

      Call alpsiz(2.0)
      Call tcurs(780,690)
      Call carwrt('FREQ : 200 KHZ')
      Call tcurs(780,670)
      Call carwrt('SIZE : 10 WAVELENGTHS')
      Call tcurs(780,650)
      Call carwrt('MEAS : 5 WAVELENGTHS')
      Call tcurs(780,550)
```

```
      Call newpen(1)
      Call alpsiz(1.0)
      Call carwrt(' + + + + + + + +')
      Call tcurs(880,550)
      Call alpsiz(2.0)
      Call carwrt('FOURIER')
      Call tcurs(780,530)
      Call newpen(2)
      Call tdraw(860,530)
      Call tcurs(880,530)
      Call carwrt('INTEGRATION')

C -----------------------------------------------------------------
C       Read fourier data from file
C -----------------------------------------------------------------

      Open (UNIT=20,STATUS='OLD',FORM='FORMATTED',FILE=fourfile)
      Read (20,*) xr
      Close (UNIT=20)

C -----------------------------------------------------------------
C       Find the phase, and the maximum magnitude
C -----------------------------------------------------------------

      Zmax=0.
      Do 3 j=1,n
         Re=dreal(xr(j))
         Zim=dimag(xr(j))
         Zmag(j)=sqrt(re**2.+zim**2.)
         If (re.ne.0.) then
            Phas(j)=atand(zim/re)
         Else
            Phas(j)=90.
         End if
         If (re.lt.0..and.zim.lt.0.) phas(j)=phas(j)-180.
         If (re.lt.0..and.zim.gt.0.) phas(j)=phas(j)+180.
         If (zmag(j).gt.zmax) zmax=zmag(j)
3     Continue

C -----------------------------------------------------------------
C       Normalize the magnitude
C -----------------------------------------------------------------

      Do 4 j=1,n
         Zmag(j)=zmag(j)/zmax
         If (zmag(j).lt.0.1) phas(j)=0.
4     Continue


C -----------------------------------------------------------------
C       Plot Fourier data
C -----------------------------------------------------------------

      Call doplot(zmag,phas,1,1,n)
      Call doplot(zmag,phas,2,1,n)

C -----------------------------------------------------------------
C       Read integration data from file
C -----------------------------------------------------------------

      Open (UNIT=20,STATUS='OLD',FORM='FORMATTED',FILE=intfile)
```

```
      Read (20,*) xr
      Close (UNIT=20)

C -------------------------------------------------------------
C     Find the phase, and the maximum magnitude
C -------------------------------------------------------------

      Do 6 j=1,n
         Re=dreal(xr(j))
         Zim=dimag(xr(j))
         Zmag(j)=sqrt(re**2.+zim**2.)
         Phas(j)=atand(zim/re)
         If (re.lt.0..and.zim.lt.0.) phas(j)=phas(j)-180.
         If (re.lt.0..and.zim.gt.0.) phas(j)=phas(j)+180.
         If (zmag(j).gt.zmax) zmax=zmag(j)
6     Continue

C -------------------------------------------------------------
C     Normalize the magnitude
C -------------------------------------------------------------

      Do 8 j=1,n
         Zmag(j)=zmag(j)/zmax
         If (zmag(j).lt.0.1) phas(j)=0.
8     Continue

C -------------------------------------------------------------
C     Plot Integration data
C -------------------------------------------------------------

      Call doplot(zmag,phas,1,2,n)
      Call doplot(zmag,phas,2,2,n)

C -------------------------------------------------------------
C     End of plot
C -------------------------------------------------------------

      Call finitt(0,0)
      Stop
      End

C -------------------------------------------------------------
C     Subroutine to plot the magnitude or phase
C     depending on the value of iflag
C     iflag=1 - plot phase
C     iflag=2 - plot magnitude
C -------------------------------------------------------------

      Subroutine doplot(zmag,phas,iflag,ifour,n)
      Real zmag(128),phas(128)
      Xpl=-floatj(n/4)                  ! Starting point for plotting

C -------------------------------------------------------------
C     Plot the points
C -------------------------------------------------------------

      Do 600 j=1,n

C ----- Find the point to be plotted -----

         If (iflag.eq.2) then
```

```
            Plt=zmag(j)
         Else
            Plt=phas(j)
         End if

C ------------------------------------------------------------
C        Plot the phase as discrete lines
C        Plot the magnitude continuously
C ------------------------------------------------------------

         If (iflag.eq.1) then                  ! Phase to be plotted
            Call dcurs(ifour,xpl,0.)
            Call ddraw(ifour,xpl,plt)
         Else
            If (j.eq.1) then                   ! Magnitude to be plotted
               Call dcurs(3,-32.,0.)
            Else
               If (ifour.eq.2) then            ! Integration data plotted
                  Call ddraw(3,xpl,plt)        ! as a solid line
               Else
                  Call cross(xpl,plt,2,3)      ! Fourier data plotted as
               End if                          ! crosses
            End if
         End if
         Xpl=xpl+.5
600      Continue
         Return
         End
```