Multiple Plane Wave Analysis of Acousto-Optic Diffraction of Gaussian Shaped Light Beams

by

John Horger

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electrical Engineering

APPROVED:

T.-C. Poon, Chairman

R. O. Claus

I. M. Besieris

August 1987

Blacksburg, Virginia

Multiple Plane Wave Analysis of Acousto-Optic Diffraction of Gaussian Shaped Light Beams

by

John Horger

Committee Chairman: Ting-Chung Poon
Electrical Engineering

(ABSTRACT)

A short history of acousto-optics research is presented along with a general description of how light and sound interact.

The Multiple Scattering model is derived and used with a Gaussian light beam to observe the distortion in light beam profile within the sound field. Numerical results are presented for comparison to previous studies using thick holograms and two orders of light. The results from using two light orders are compared to four light order results.

A Hamming sound amplitude distribution is introduced as a possible way to reduce the amount of light beam profile distortion.

Table of Contents

Chapter 1

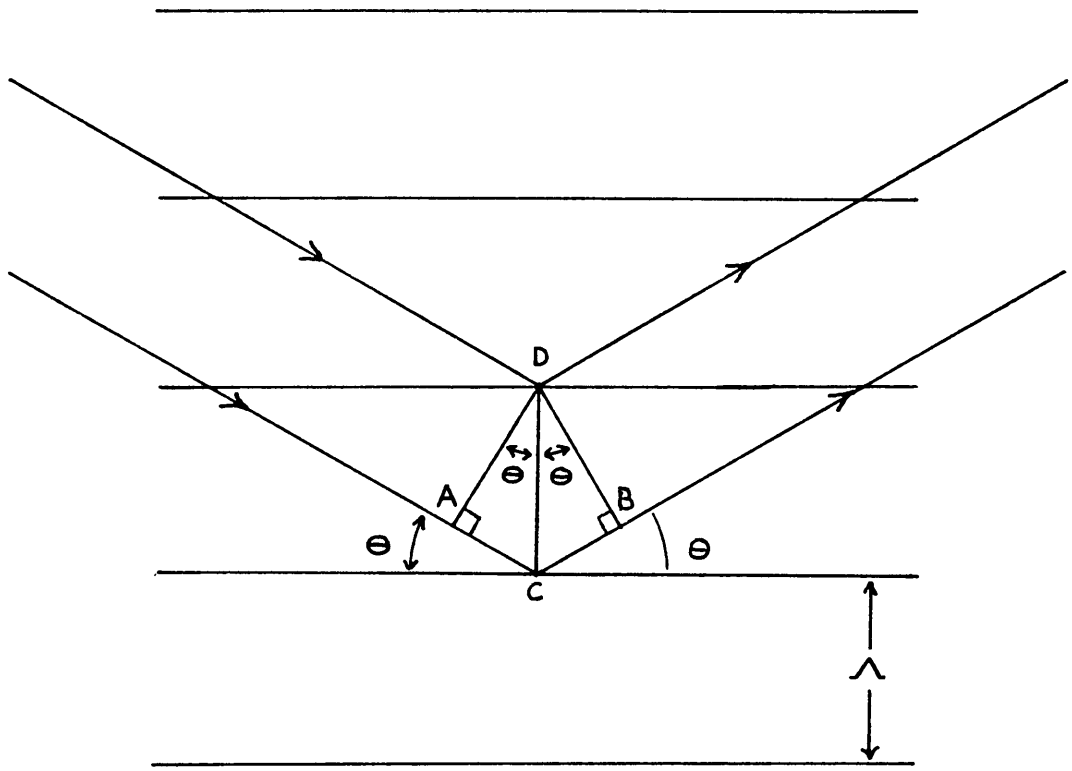
A Review of Acousto-Optics

In 1922 Leon Brillouin predicted that light could be scattered by sound waves under certain conditions (1). He reasoned that since a sound wave propagating through a fluid produces alternating regions of compression and rarefaction and a light wave moves fast enough compared to a sound wave to make those regions appear stationary, the light waves would reflect off of the planes of compression. If two parallel rays of light in phase with each other reflect off two adjacent layers of compression, then they would constructively interfere after reflection only if the difference in path length between the two rays is one wavelength long. This would only occur at a specific light angle of incidence. Referring to Figure 1.1, the length ABC would have to be one wavelength. Since AC and BC are equal, they each must be a half wavelength long. A simple relation for the incident angle can be derived from Figure 1.1:

$$\sin\Theta = BC / CD = \lambda / 2\Lambda, \qquad (1\text{-}1)$$

where λ is the wavelength of light, Λ is the wavelength of sound, and Θ is the angle of light incidence. This relation appears in the diffraction of X-rays by crystals and the angle is known as the Bragg angle ϕ_B.

Brillouin's prediction was experimentally proven in

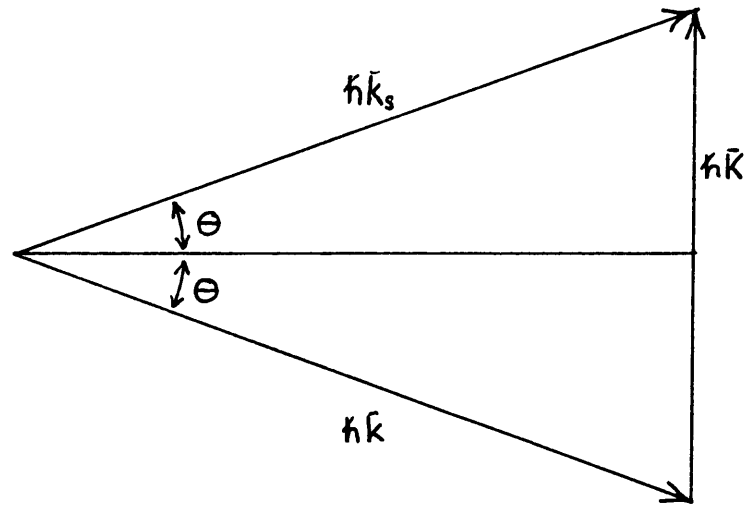


REFLECTION OFF REGIONS OF COMPRESSION

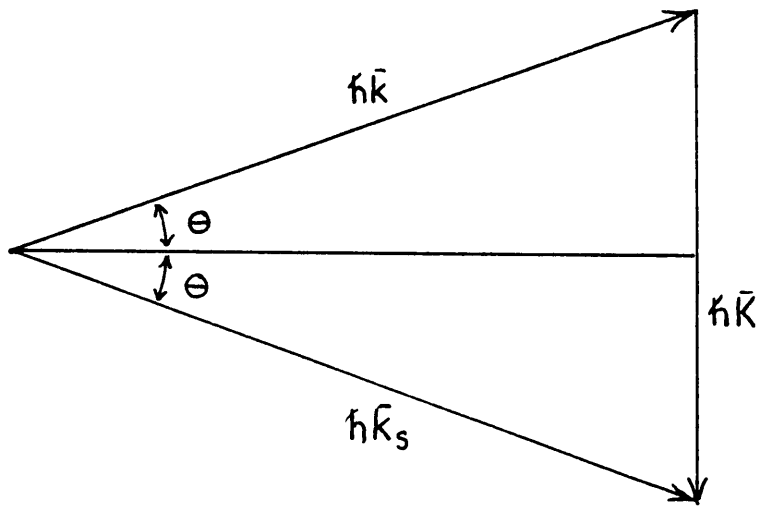
FIGURE 1.1

1932 by two independent groups: Lucas and Biquard (2) in France, and Debye and Sears (3) in the United States. Debye and Sears used a quartz crystal immersed in a water tank to produce sound waves in the MHz range and various lines of a mercury arc for a light source. The scattering of light by sound was achieved, but the Bragg angle was not distinct. Many orders of light were generated which were symmetric about the unscattered order only for light incidence parallel to the planes of compression, $\Theta = \emptyset$ in Figure 1.1. Also unexplained was the periodic intensity of the light orders as the angle was changed. Debye and Sears also predicted that the motion of the sound beam would doppler shift the light frequencies.

One of the reasons the results were not as Brillouin predicted is that, unlike atomic planes in a crystal which are discrete, compression in a sound beam is sinusoidal in nature. A more comprehensive way of explaining acousto-optic interaction is with phonon and photon collisions (4). A photon has momentum $\hbar k$ and a phonon has momentum $\hbar K$, where $\hbar = h / 2\pi$, h being Planck's constant, k is the wave number of light $2\pi/\lambda$, and K is the wave number of sound $2\pi/\lambda$. A phonon-photon collision can occur with two different geometries as shown in figure 1.2. Since the acoustic momentum is small relative to the light momentum, the scattered light momentum $\hbar k$ is nearly equivalent to the



a.



b.

LIGHT AND SOUND MOMENTUMS

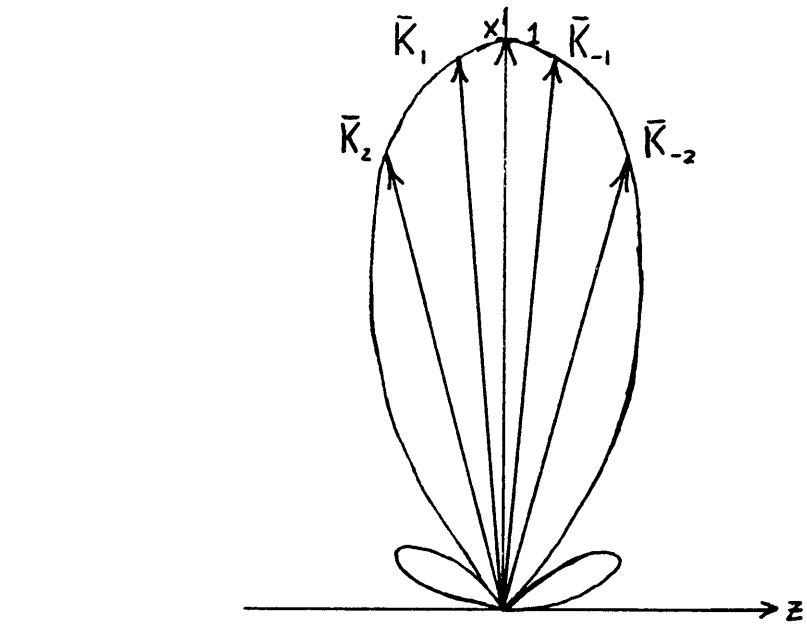
FIGURE 1.2

incident light momentum. We can derive a relation for the incident angle from the figure:

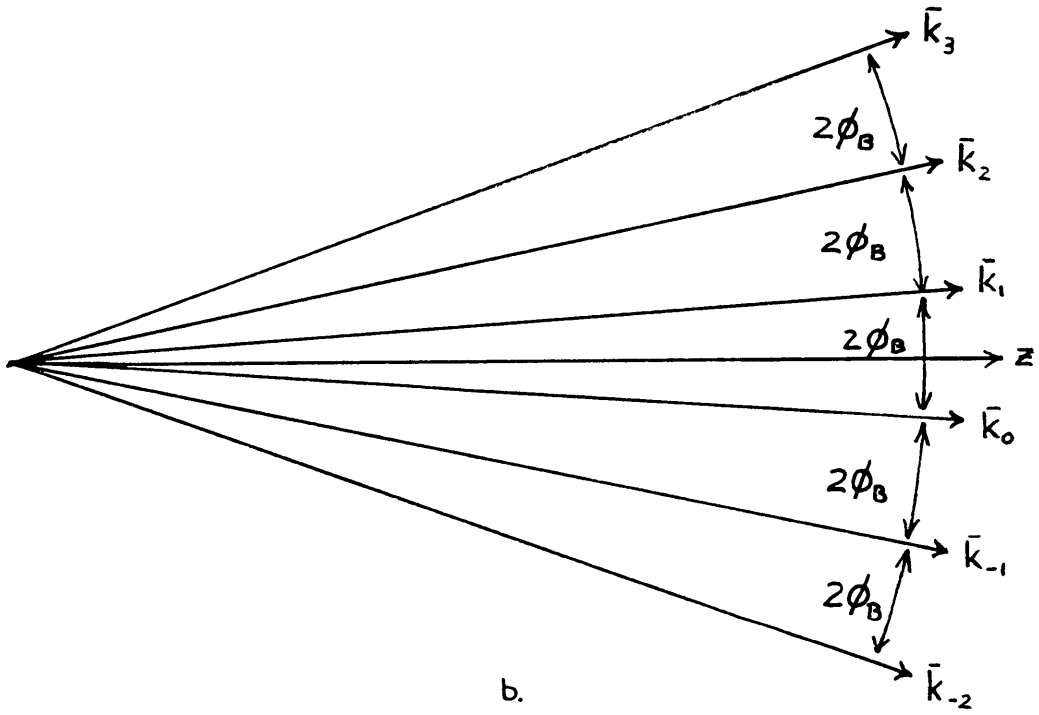
$$\sin \Theta = \hbar K / 2\hbar k = K / 2k = \lambda / 2\Lambda, \qquad (1\text{-}2)$$

which is the Bragg angle relation. The energies associated with light and sound are $\hbar\omega$ and $\hbar\Omega$, where ω and Ω are the radian frequencies of the light and sound respectively. Making no approximations, the scattered light energy would be $\hbar(\omega + \Omega)$ for the upshifting case in figure 1.2a, and $h(\omega - \Omega)$ for the downshifting case in figure 1.2b.

The theory so far predicts that an infinite plane wave of light interacting with an infinite plane wave of sound, characterized by a photon and phonon, will interact to produce one order of scattered light if the incident angle is $\pm \phi_B$. If the acoustic transducer is finite, the sound wave will no longer be planar. It will have a radiation pattern (angular or Fourier spectrum) as shown in figure 1.3a, giving rise to scattered light orders illustrated in figure 1.3b (4). This pattern is developed in the same manner as antenna radiation patterns. For the uniform transducer used in these examples, the antenna analog is the uniform line source (5). As the length of the transducer is reduced, the central lobe of the radiation pattern broadens, allowing more phonons propagating at wider angles. For example, a long transducer might only radiate the sound



a.



b.

ACOUSTIC ANGULAR SPECTRUM AND
ASSOCIATED LIGHT ORDERS

FIGURE 1.3

orders K_{+1} and K_{-1} in the central lobe. Phonons travelling in these directions would interact with the incident light, k_0, to produce the k_{+1} and k_{-1} orders of light. A shorter transducer could also radiate K_{+2} and K_{-2} phonons which would react with the k_{+1} and k_{-1} photons to produce k_{+2} and k_{-2} light orders, and so on. Rescattering can also happen such that a previously upshifted photon is downshifted and visa versa.

The angular width of the main lobe is proportional to Λ / L, L being the transducer length. The angular separation between the interacting phonons is twice the Bragg angle, λ / Λ. A ratio of these quantities is often used to define the interaction condition:

$$Q = 2\pi \lambda L / \Lambda^2 = K^2 L / k \ . \qquad (1\text{-}3)$$

The index of refraction of the interaction medium can be put in the denominator for referring the output light to another medium (6). For $Q >> 1$ the interaction region is long and the sound is nearly planar. This is called the Bragg region. For $Q < 1$ the interaction region is short; the sound is diffuse. This is the so called Debye-Sears or Raman-Nath region after early researchers in the field.

Since the amount that light can be scattered and rescattered by sound is dependent on the availability of phonons, the intensity of the light orders depends on the

intensity of the sound for a constant incident angle. In the Debye-Sears regime the light order amplitudes are proportional to the Bessel functions:

$$A_n = (-j)^n E_{INC} J_n(kC|S|L / 2) , \qquad (1-4)$$

where A_n is the nth order light amplitude, $j = \sqrt{-1}$, E_{INC} is the input light amplitude, C is a constant dependent on the interaction medium, J_n is the nth order Bessel function, and $|S|$ is the sound amplitude (4).

For Bragg diffraction, Q is large and $\Theta = \phi_B$, the scattered light amplitudes are given by:

$$\begin{aligned} E_o &= E_{INC} \cos(kC|S|L / 4) \\ E_{+1 \text{ OR } -1} &= -j\, E_{INC} \sin(kC|S|L / 4) , \end{aligned} \qquad (1-5)$$

where either the +1 or -1 order of light will exist dependent on upshifting or downshifting operation (4).

Applications for acousto-optics started shortly after its verification with an optical television in the mid-thirties (7,8). The development of the laser in the 1960s provided a more convenient light source and opened the way for coherent light processing. An extensive list of references for theory and applications appears in a report by Korpel (9). Some more recent applications include an interferometric method of measuring surface shape and vibration (10), spatial display of the correlation of two

electrical signals (11), two dimensional image processing (12,13,14), and Gaussian beam shaping (15).

The last topic, the effects of acousto-optic interaction on Gaussian light beams, is the topic of this study. The multiple plane wave scattering model of acousto-optic interaction developed by Korpel (9,16) is used to examine the distortion of a Gaussian profile light beam as it propagates through a sound beam. The distortion was shown experimentally by Forshaw (17), and a mathematical representation has been presented by Chu and Tamir (18). A mathematical development of the multiple scattering model is presented. Detailed graphical representations of the output light orders are presented and compared to related work by Moharam, Gaylord and Magnusson (19).

Chapter 2

The Multiple Plane Wave Scattering Model of Acousto-Optic Interaction

Since the prediction of acousto-optic interaction by Brillouin in 1922, many ways of mathematically modeling the scattering have been presented. Raman and Nath first treated the problem as light encountering a sinusoidal phase grating (20,21), then later using a wave equation approach (22,23). Bhatia and Noble used a model based on scattering by element volumes (24). These methods assumed that both the light and the sound fields were planar. Chu and Tamir developed a method for the bounded light beam case by using its angular plane wave spectrum, but the sound remains planar (18).

As mentioned in the preceding review, the sound field cannot be truthfully represented as a plane wave unless the sound source is very long compared to the sound wavelength. This implies large values of Q (eq. 1-3). The strong interaction model presented by Korpel (9,16) allows accurate representation of the light and sound fields by use of their respective angular spectra.

The following derivation shows the inclusion of the Fourier spectra of both the light and the sound beams and highlights the approximations that are made. It primarily follows the derivation published by Korpel (16) with some

differences in trigonometry relating to the directions of propagation of the light and sound beams, ϕ and Θ.

The general geometry for the interaction is shown in figure 2.1. It has been shown that the two dimensional interaction of sound and light fields with respective frequencies Ω and ω can be described by the following system of equations, provided that the sound velocity is far exceeded by the light velocity and the interaction medium is isotropic both optically and acoustically (25):

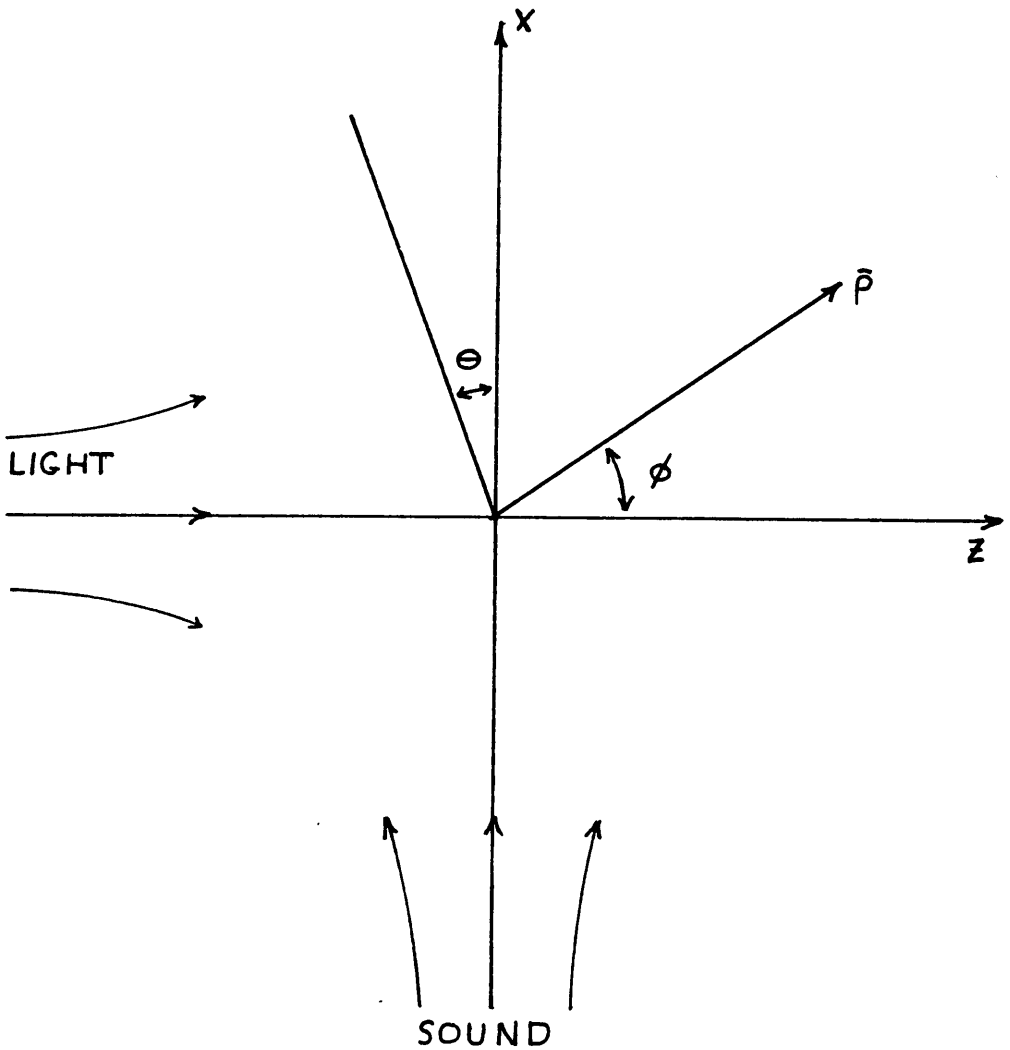
$$\nabla^2 E_n(\rho) + k^2 E_n(\rho) + 1/2\, k^2 CS(\rho)\, E_{n-1}(\rho) + 1/2\, k^2 CS^*(\rho)\, E_{n+1}(\rho) = 0, \qquad (2\text{-}1)$$

where ∇^2 is the LaPlacian operator $\partial^2/\partial x^2 + \partial^2/\partial z^2$, $E_n(\rho)$ is the light field at frequency $\omega + n\Omega$, k is the wave number of the incident light, $S(\rho)$ is the complex sound field, * denotes the complex conjugate, and C is a constant given by $-n^2 p$, where n is the refractive index of the unperturbed medium and p is the strain-optic constant given by:

$$\Delta(1/n_i^2) = p_{ij} S_j, \qquad (2\text{-}2)$$

S_j being the strain (26). The value of C is actually a tensor which reduces to a constant in isotropic materials.

The sound field can be written as its angular spectrum by way of its Fourier transform:



INTERACTION GEOMETRY

FIGURE 2.1

$$S(\rho) = \int \bar{S}(\Theta) \exp(-j\bar{K}\cdot\bar{\rho})\, d\Theta, \qquad (2\text{-}3)$$

where $\bar{K}$ is the acoustic propagation vector in the Θ direction. Similarly, the light fields, $E_n(\rho)$, can be written as their spectra:

$$E_n(\rho) = \int \bar{E}_n(z,\phi) \exp(-j\bar{k}\cdot\bar{\rho})\, d\phi, \qquad (2\text{-}4)$$

where k is the light propagation vector in the ϕ direction. The fields must contain a z dependence since they are constantly being modified as they cross the sound field. The incident light field is $\bar{E}_{INC}(z,\phi) = \bar{E}_0(-\infty,\phi)$. Putting (2-4) into the first two terms of (2-1) gives:

$$\nabla^2 E_n(\rho) + k^2 E_n(\rho) = \int \partial^2/\partial z^2\, \bar{E}_n(z,\phi) \exp(-j\bar{k}\cdot\bar{\rho})\, d\phi - 2j\int k_z\, \partial/\partial z\, \bar{E}_n(z,\phi) \exp(-j\bar{k}\cdot\bar{\rho})\, d\phi, \qquad (2\text{-}5)$$

where $k_z = k\cos\phi$. The first term on the right side of (2-5) is dropped with the assumption that $\partial^2/\partial z^2 \bar{E}_n$ is fairly small over distances on the order of λ, the light wavelength, compared to $k_z\, \partial/\partial z\, \bar{E}_n$. Putting (2-3), (2-4) and (2-5) into (2-1), we have:

$$k_z \int \partial/\partial z\, \bar{E}_n(z,\phi) \exp(-j\bar{k}\cdot\bar{\rho})\, d\phi = -1/4\, jk^2 C \left[\iint \bar{S}(\Theta)\, \bar{E}_{n-1}(z,\phi) \exp -j(\bar{K}\cdot\bar{\rho} + \bar{k}\cdot\rho)\, d\Theta d\phi + \iint \bar{S}^*(\Theta)\, \bar{E}_{n+1}(z,\phi) \exp -j(-\bar{K}\cdot\bar{\rho} + \bar{k}\cdot\bar{\rho})\, d\Theta d\phi\right]. \qquad (2\text{-}6)$$

Using the definition for the Dirac delta function:

$$2\pi\ (k - k_n) = \int_{-\infty}^{+\infty} \exp\ j(k_n x - kx)\ dx\ , \qquad (2\text{-}7)$$

equation (2-6) can be multiplied by $\exp(jk_{nx}\, x)$ and integrated over $-\infty < x < +\infty$ to produce for the left side:

$$k_z \iint \partial / \partial z\ \bar{E}_n(z,\phi)\ \exp -j(k_z z + k_x x - k_{nx} x)\ d\phi dx = 2\pi \int k_z\ \partial / \partial z\ \bar{E}_n(z,\phi)\ \exp(-jk_z z)\ \delta(k_x - k_{nx})\ d\phi\ . \qquad (2\text{-}8)$$

Substitution of $dk_x = d(k \sin\phi) = k \cos\phi\ d\phi = k_z\ d\phi$, $k \sin\phi = k_x$ and $k \sin\phi_n = k_{nx}$ into (2-8) gives:

$$2\pi \int \partial / \partial z\ \bar{E}_n(z,\phi)\ \exp(-jk_z z)\ \delta(k_x - k_{nx})\ d\phi = 2\pi \exp(-jkz \cos\phi_n)\ \partial / \partial z\ \bar{E}_n(z,\phi_n)\ . \qquad (2\text{-}9)$$

Similar treatment of the right-hand side first term in (2-6) gives:

$$-j/4\ k^2 C \iiint \bar{S}(\theta)\ \bar{E}_{n-1}(z,\phi)\ \exp\ j(-K_x x - K_z z - k_z z - k_x x + k_{nx} x)\ d\theta d\phi dx = -j/4\ k\ C\ 2 \iiint \bar{S}(\theta)\ \bar{E}_{n-1}(z,\phi)\ \exp\ j(-K_z z - k_z z)\ (k_x - k_{nx} + K_x)\ d\theta d\phi\ . \qquad (2\text{-}10)$$

Using $k_z = k \cos\phi$ and $k \sin\phi_{n-1} = k_{nx} - K_x$, (2-10) becomes:

$$-jkC/4\ 2\pi \int \bar{S}(\theta)\ \bar{E}_{n-1}(z,\phi_{n-1})/\cos\phi_{n-1}\ x\ \exp\ j(-K_z z - kz \cos\phi_{n-1})\ d\theta\ . \qquad (2\text{-}11)$$

The same process with the second term using $k \sin\phi_{n+1} = k_{nx} + K_x$ gives:

$$-jkC/4\ 2\pi \int \tilde{S}^{*}(\Theta)\ \tilde{E}_{n+1}(z,\phi_{n+1})/\cos\phi_{n+1} \quad x\ \exp j(K_z z - kz\cos\phi_{n+1})\ d\Theta\ . \tag{2-12}$$

Combining (2-9), (2-11), (2-12) and substituting $K_z = -K \sin\Theta$, we have:

$$\partial/\partial z\ \tilde{E}_n(z,\phi_n) = -jkC/4\ [\int \tilde{S}(\Theta)\ \tilde{E}_{n-1}(z,\phi_{n-1})/\cos\phi_{n-1}\ \exp j(Kz\sin\Theta - kz\cos\phi_{n-1} + kz\cos\phi_n)\ d\Theta + \int \tilde{S}^{*}(\Theta)\ \tilde{E}_{n+1}(z,\phi_{n+1})/\cos\phi_{n+1}\ \exp -j(Kz\sin\Theta + kz\cos\phi_{n+1} - kz\cos\phi_n)\ d\Theta]\ . \tag{2-13}$$

Employing trigonometric identity:

$$\cos x - \cos y = -2\sin 1/2(x+y)\sin 1/2(x-y), \tag{2-14}$$

the exponentials of (3-13) become:

$$\cos\phi_n - \cos\phi_{n-1} = -2\sin 1/2(\phi_n + \phi_{n-1})\sin 1/2(\phi_n - \phi_{n-1}) = -2\sin 1/2(\phi_n + \phi_{n-1})\sin 1/2\ \phi_1 \tag{2-15}$$

and:

$$\cos\phi_{n+1} - \cos\phi_n = -2\sin 1/2(\phi_{n+1} + \phi_n)\sin 1/2(\phi_{n+1} - \phi_n) = -2\sin 1/2(\phi_{n+1} + \phi_n)\sin 1/2\ \phi_1\ , \tag{2-16}$$

wnere ϕ_1 is the angular separation between the light orders. Factoring 2k out of the exponentials in (2-13) gives for the exponents:

term 1;

$$\exp j2kz\ (K/2k \sin\Theta - \sin 1/2(\phi_n + \phi_{n-1}) \sin 1/2\,\phi_1) \tag{2-17}$$

term 2;

$$\exp -j2kz\ (K/2k \sin\Theta - \sin 1/2(\phi_{n+1} + \phi_n) \sin 1/2\,\phi_1)\ .$$

For the strongest deflection of energy from the n+1 and n-1 orders into the nth order, the exponential terms should vanish. This happens when:

$$\Theta = 1/2\ (\phi_n + \phi_{n-1}) \text{ for the n-1 order}\ , \tag{2-18}$$

$$\Theta = 1/2\ (\phi_{n+1} + \phi_n) \text{ for the n+1 order} \tag{2-19}$$

and

$$\sin\ (1/2)\ \phi_1 = K\ /\ 2k\ . \tag{2-20}$$

Equation (2-20) gives the relation for the Bragg angle when K / 2k is small. For such small angles $\sin\Theta \cong \Theta$ and $\cos\Theta = 1$. Rewriting (2-13), we have:

$$\begin{aligned}\partial/\partial z\ \tilde{E}_n(z,\phi_n) = -jkC/4\ [&\int \tilde{S}(\Theta)\ \tilde{E}_{n-1}(z,\phi_{n-1})/\cos\phi_{n-1} \\ &\times \exp j2kz\,\phi_B(\Theta - 1/2(\phi_n + \phi_{n-1}))\ d\Theta \\ &+ \int \tilde{S}^*(\Theta)\ \tilde{E}_{n+1}(z,\phi_{n+1})/\cos\phi_{n+1} \\ &\times \exp -j2kz\,\phi_B(\Theta - 1/2(\phi_{n+1} + \phi_n))\ d\Theta]\ .\end{aligned} \tag{2-21}$$

Using $1/2(\phi_n + \phi_{n-1}) = (2n-1)\phi_B$, $1/2(\phi_{n+1} + \phi_n) = (2n+1)\phi_B$, and $\phi_B = K/2k$ in (2-21):

$$\partial / \partial z\ \tilde{E}_n(z,\phi_n) = -jkC/4\ [\int \tilde{S}(\Theta)\ \tilde{E}_{n-1}(z,\phi_{n-1})/\cos\phi_{n-1}$$
$$\exp jKz(\Theta - (2n-1)\phi_B)\ d\Theta \qquad (2\text{-}22)$$
$$+\int \tilde{S}^*(\Theta)\ \tilde{E}_{n+1}(z,\phi_{n+1})/\cos\phi_{n+1}\ \exp -jKz(\Theta - (2n+1)\phi_B)\ d\Theta] \ .$$

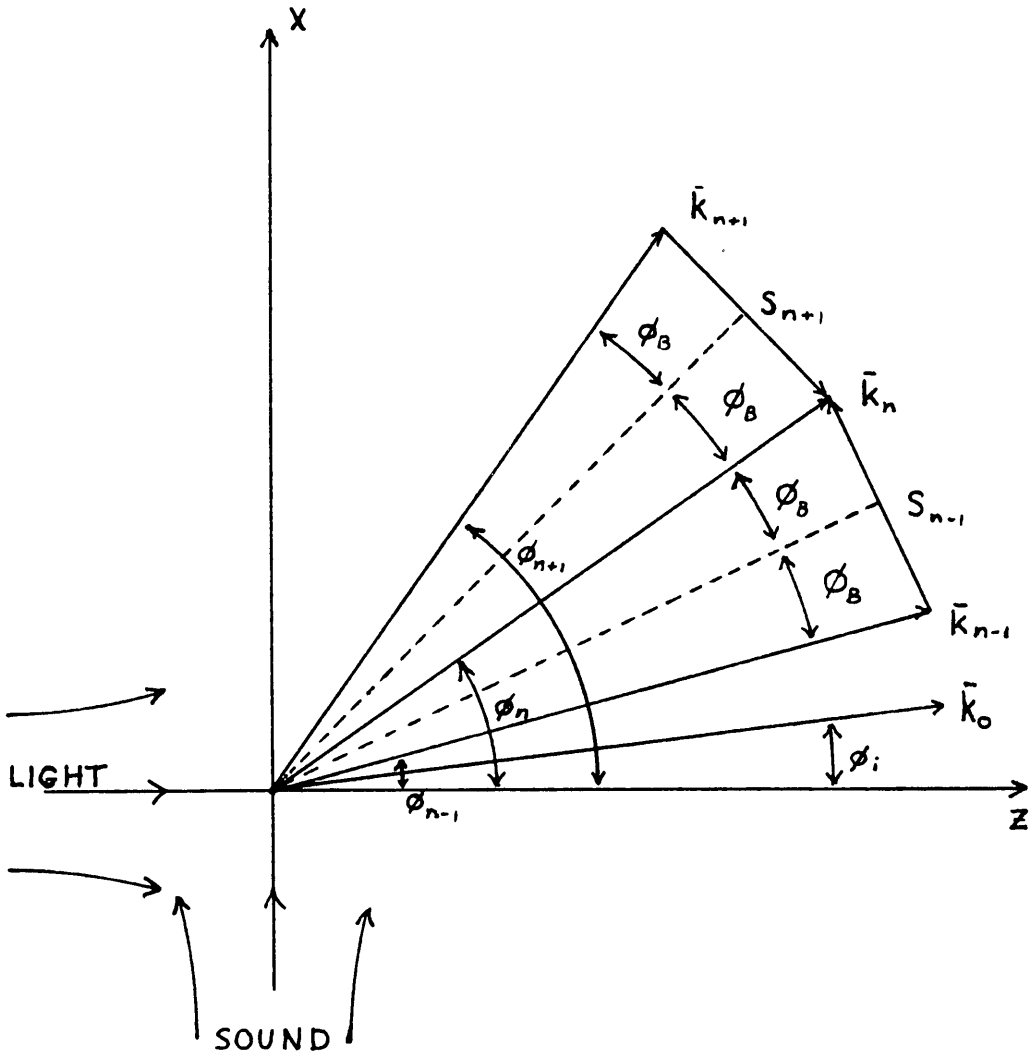
Using $\phi_{n-1} = \phi_n - 2\phi_B$, $\phi_{n+1} = \phi_n + 2\phi_B$ and defining coordinates:

$$x_{n-1} = z\phi_B(2n-1) \text{ and } x_{n+1} = z\phi_B(2n+1) \qquad (2\text{-}23)$$

in (2-22), the components take on the form of Fourier transforms and can be rewritten as:

$$\partial / \partial z\ \tilde{E}_n(z,\phi_n) =$$
$$-jkC/4\ [S(z,x_{n-1})\ \tilde{E}_{n-1}(z,\phi_n - 2\phi_B)/\cos(\phi_n - 2\phi_B)$$
$$+\ S^*(z,x_{n+1})\ \tilde{E}_{n+1}(z,\phi_n + 2\phi_B)/\cos(\phi_n + 2\phi_B)] \ . \qquad (2\text{-}24)$$

A graphical representation of (2-24) is shown in figure 2.2. The change in amplitude of the nth order of light comes from contributions from the n+1 and n-1 light orders. The n+1 light interacts with sound $S(x_{n+1})$ which, in figure 2.2, are phonons propagating perpendicular to the S_{n+1} line. The resultant photons are in the nth order. In the same fashion, the n-1 light interacts with sound $S(x_{n-1})$ which are phonons traveling perpendicular to the S_{n-1} line. Simultaneously, the n+2 and nth light orders are contributing to the n+1 order, the nth and n-2 light orders are contributing to the n-1 order and so on. Now, the bounded planar sound column mentioned above is given by:



DIFFRACTION INTO THE nTH ORDER

FIGURE 2.2

$$S_{n-1} = |S| \exp\ [-jKz(\phi_i + (2n-1)\phi_B)]$$

$$= |S| \exp\ [-jK^2z/2k\ (\phi_i / \phi_B + 2n-1)]$$

$$= |S| \exp\ [-jQ\xi / 2\ (\phi_i / \phi_B + 2n-1)]\ , \qquad (2\text{-}25)$$

where ξ = z/L is the normalized distance across the sound field, ϕ_i is the angle of incidence and Q is the Klein-Cook parameter, K^2L/k (6). Similarly:

$$S_{n+1} = |S| \exp\ [jQ\xi /2\ (\phi_i / \phi_B + 2n+1)]\ . \qquad (2\text{-}26)$$

This notation assumes a real sound field, that is $S = S^*$, for simplicity, and ϕ_B is positive for the downshifting case (27). This sound model is used in this study for purposes of comparison and because the light beam distortion investigated appears at high Q values where the planar sound approximation is valid. A rigorous derivation of some familiar equations has been done by Poon and Korpel (27). Also, treatment of the bounded case has been done using Feynman diagrams by Poon and Korpel (28,29). A study that successfully verified this model in the case of adjacent sound columns has recently been done by Poon, Chatterjee and Banerjee (30).

Chapter 3

Diffraction of Gaussian Light Beams by Thick Sinusoidal Gratings

Since the phenomenon of light-sound interaction is most easily seen when coherent light is used to illuminate the sound beam, the laser is a logical light source for the purpose. Laser light is coherent, and highly directional. A problem with the laser as a light source for acousto-optics is the finite extent of the beam. Early models of acousto-optic interaction assumed infinite plane waves of light. It has been seen that the intensity profile of a laser beam, which is typically Gaussian in shape, distorts for thick sound beams with high sound intensities (17). An analytical approach to the beam distortion phenomenon in light diffraction by thick holograms has been done by Chu and Tamir (18), and Moharam, Gaylord and Magnusson (19). A thick hologram (i.e., a thick phase grating) is approximately equivalent to a bounded sound column, provided the sound velocity is small compared to the light velocity. The article by Moharam et al. presents calculated results in graphs, showing the evolution of a Gaussian light beam as the grating strength increases for several geometrical configurations. It is to these results that analyses of similar sound cell configurations, using the multiple scattering model, are compared.

Correlating Notation:

Moharam et al. use two variables in their analysis to illustrate the effect of bounded light beam diffraction by a thick hologram: a grating strength and a geometry factor. The grating strength is proportional to the variation in refractive index in the grating and is given by:

$$\gamma = n_1 \pi d / \lambda \cos \Theta \ , \tag{3-1}$$

where n_1 is the peak amplitude of the sinusoidal index variation, Θ is the light incident angle and d is the grating thickness. The grating thickness is specifically determined by the geometry factor:

$$g = d \sin \Theta / w \ , \tag{3-2}$$

where w is the $1/e^2$ Gaussian beam radius.

The multiple scattering model also uses two variables Q and $\hat{\alpha}$, which can be seen in the following formulation derived from (2-22), (2-25) and (2-26):

$$\begin{aligned} d\bar{E}_n / dz = -jkC|S|/4 \{ & \exp [-jQz/2L \ (\phi_i/\phi_B + (2n-1))] \bar{E}_{n-1} \\ & + \exp [jQz/2L \ (\phi_i/\phi_B + (2n+1))] \bar{E}_{n+1} \} \ . \end{aligned} \tag{3-3}$$

Using $\xi = z/L$, (3-3) becomes:

$$\begin{aligned} d\bar{E}_n / d\xi = -j\hat{\alpha}/2 \{ & \exp [-jQ\xi/2 \ (\phi_i/\phi_B + (2n-1))] \bar{E}_{n-1} \\ & + \exp [jQ\xi/2 \ (\phi_i/\phi_B + (2n+1))] \bar{E}_{n+1} \} \ , \end{aligned} \tag{3-4}$$

where $\hat{\alpha} = kCL|S|/2$, $Q = K^2L/k$ and $\cos \phi_n \cong 1$.

The parameter γ can be equated to $\hat{\alpha}$ in the following manner: strictly speaking k is the light wave number within the interaction medium so that $nk_o = k = 2\pi n / \lambda$. From the paper by Korpel (26), the peak index variation in the medium is:

$$\Delta n = 2n_1 = nC|S| \qquad (3\text{-}5)$$

For small incident angles, i.e., $\cos \phi_n \cong 1$, the values L and d are identical. Hence for (3-1):

$$\gamma = \pi n_1 d / \lambda \cos \Theta = knCL|S| / 2x2n = kCL|S|/4 = \hat{\alpha}/2. \qquad (3\text{-}6)$$

Throughout the study by Moharam et al., Θ was taken to be the Bragg angle so that $\sin \Theta = \sin \phi_B = K/2k$ for small values of K/2k. Using this in (3-2):

$$g = d \sin \phi_B / w \cong LK / 2kw = LK^2 / 2kKw = Q / 2Kw \; . \qquad (3\text{-}7)$$

Hence:

$$Q = 2gKw \; . \qquad (3\text{-}8)$$

Moharam et al. assumed that only the Øth and 1st orders of light were non-zero and they varied γ from Ø to 2π for g values 3, 1 and a value much less than 1. The equivalent equations using the multiple scattering model are:

$$d\bar{E}_0 / d\xi = -j\hat{\alpha}/2 \exp(jQ\xi/2\ (\phi_i/\phi_B + 1))\ \bar{E}_1 \qquad (3\text{-}9a)$$

$$d\bar{E}_1 / d\xi = -j\hat{\alpha}/2 \exp(-jQ\xi/2\ (\phi_i/\phi_B + 1))\ \bar{E}_0 \qquad (3\text{-}9b)$$

with $\hat{\alpha}$ varied from 0 to 4π. To match the values of g, some standard must be chosen for the Gaussian function and the sound wave. Choosing the Gaussian radius, w, to be .5 mm with a 40 MHz sound wave in flint glass, which has a longitudinal acoustic velocity of 4030 m/s (31), K is 62364 m^{-1}. With these values Q is 0, 62.4 and 187 for the respective g values of 0, 1 and 3.

Since the system of equations in (3-9) is written to describe plane waves of light, the Gaussian profile must be decomposed into superimposed plane waves, that is, its Fourier spectrum. The general form of a Gaussian and its Fourier transform are (32):

Gaussian	Fourier transform	
$g(x) = \exp(-ax^2)$	$G(\omega_x) = \sqrt{(\pi/a)}\ \exp(-\omega_x^2/4a)$,	(3-10)

where $\omega_x = 2\pi f$ and f is a spatial frequency $x/\lambda z$ where x and z are defined in the light ray diagram figure 3.1 (33). The length x is the spatial wavelength after a propagation distance of length z. Assuming that Θ is small $x/z = \tan\Theta \cong \Theta$, so $\omega_x = 2\pi\Theta/\lambda = k\Theta$. Hence the Fourier spectrum from (3-10) becomes:

$$G(\Theta) = \sqrt{(\pi/a)}\ \exp(-k^2\Theta^2/4a)\ . \qquad (3\text{-}11)$$

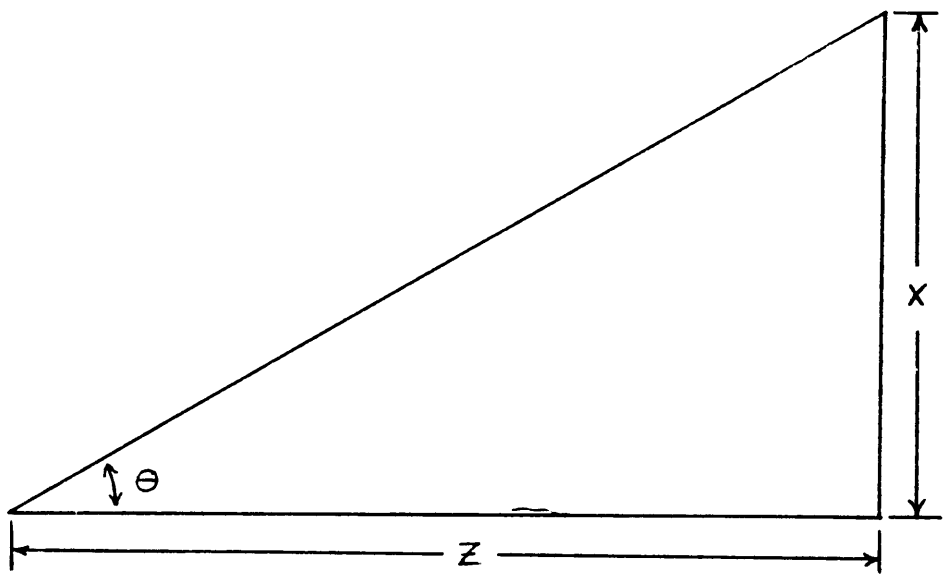


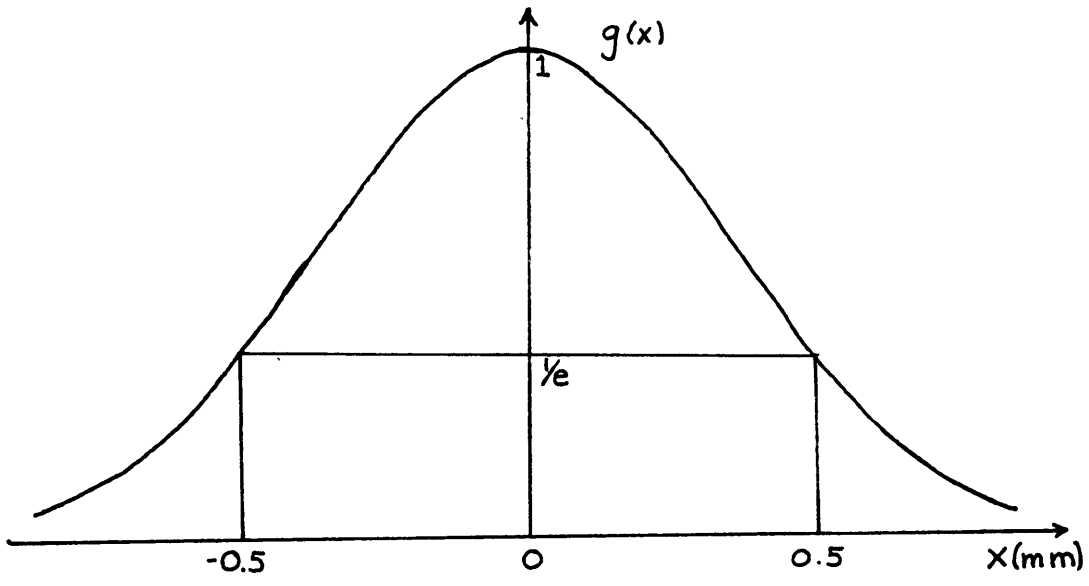
DIAGRAM FOR DEFINITION
OF SPATIAL FREQUENCIES

FIGURE 3.1

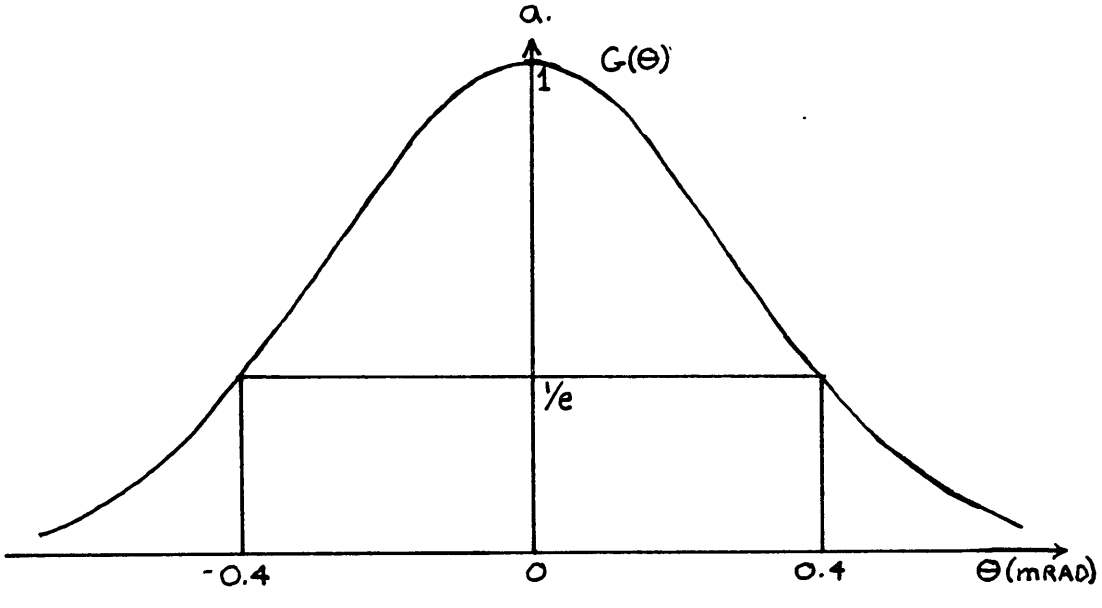
For a Gaussian with amplitude 1/e at x = 0.0005 m the constant a is 4×10^6. Using light from a He-Ne laser (λ = 6328 Å) the normalized spectrum is:

$$G(\Theta) = \exp(-6161789\,\Theta^2) \quad , \qquad (3\text{-}12)$$

which has a 1/e amplitude at Θ = 0.4 mrad so the small angle approximation is valid. Figure 3.2 shows the input Gaussian and its angular spectrum. The initial condition for the $\bar{E}_0$ light amplitude is the amplitude of the angular spectrum at the appropriate incident angle. The center of the Gaussian is at the upshifting Bragg angle, $-\phi_B$. The $\bar{E}_1$ light amplitude is 0 before entering the sound. Equation (3-9) is solved numerically for several input angles around the Bragg angle to give the output angular spectra of both the 0th and 1st order light. An inverse Fourier transform is then done to give the output intensity profiles of the two light orders. Figures 3.3, 3.4 and 3.5 show the output intensity profiles for cases corresponding to those presented by Moharam et al. in figures 2 and 3 of ref. 19. The S and R beams in ref. 19 are the E_0 and E_1 beams presented here. An explanation of the numerical analysis methods and programming used to generate these graphs appears in the appendix. The darkened line indicates the starting location of the center of each order and the spike at $\hat{\alpha}$ =0 shows the height of the input Gaussian.



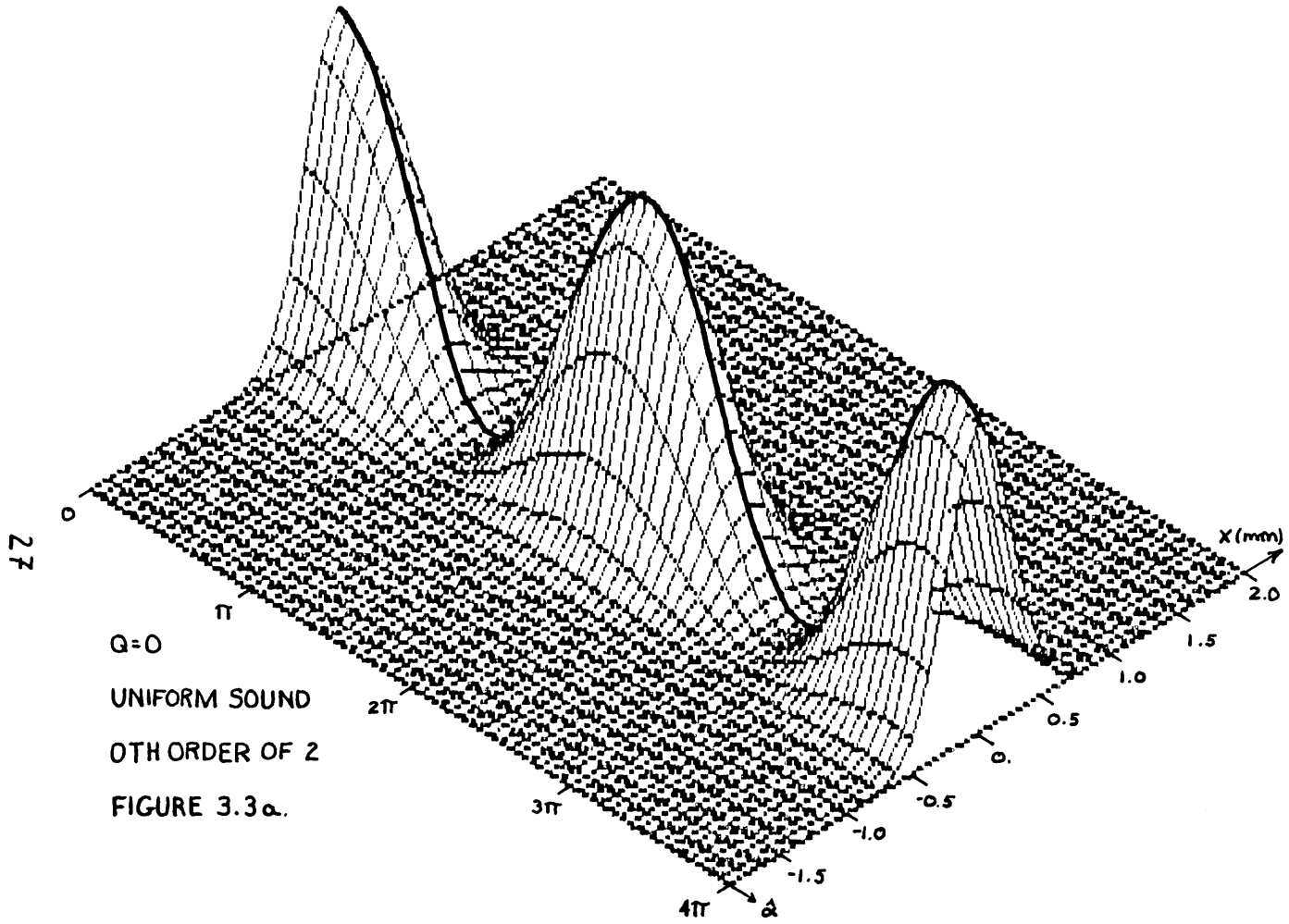
a.



b.

A GAUSSIAN PROFILE AND
ITS ANGULAR SPECTRUM

FIGURE 3.2

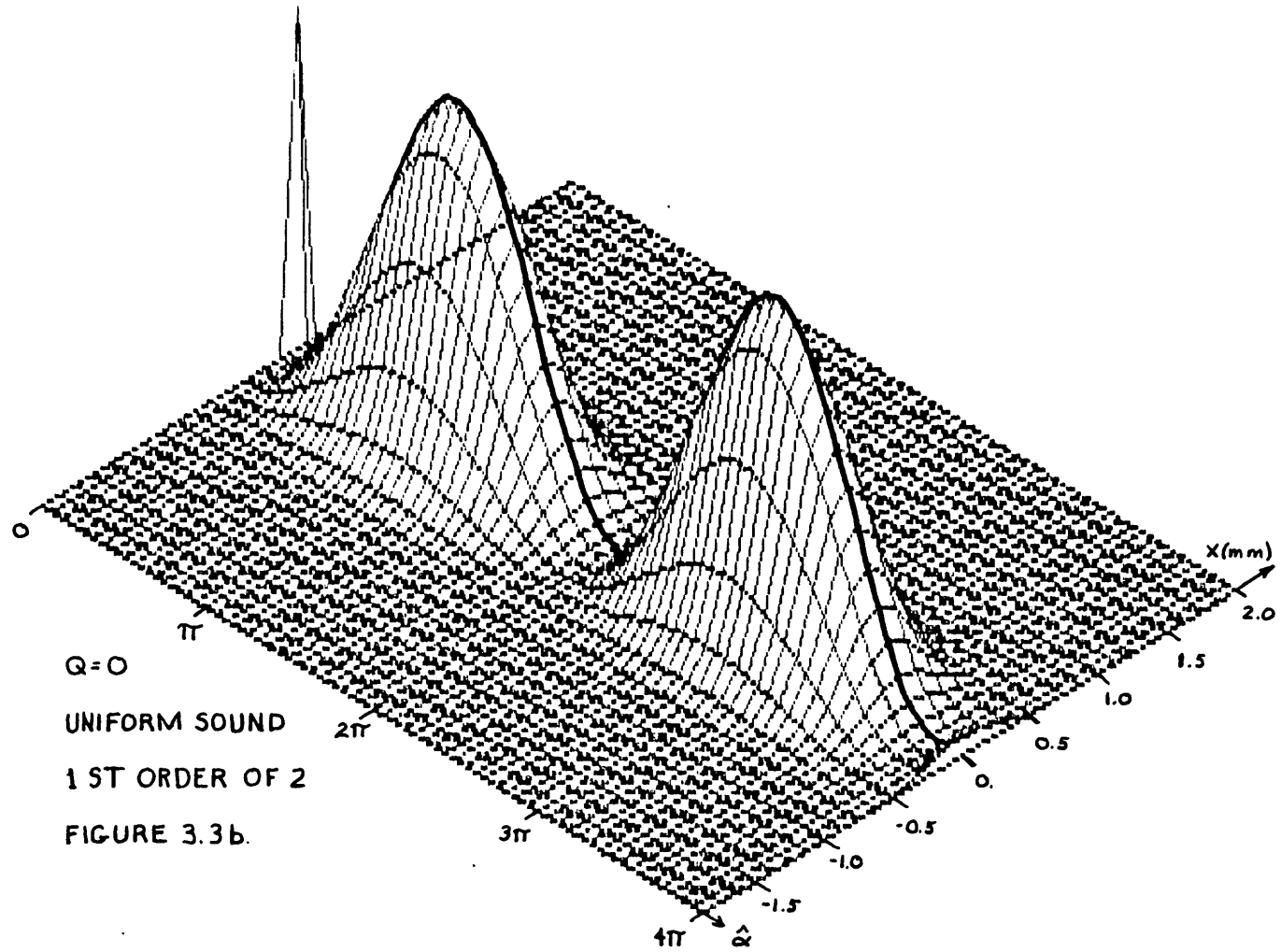


Q=0

UNIFORM SOUND

0TH ORDER OF 2

FIGURE 3.3a.

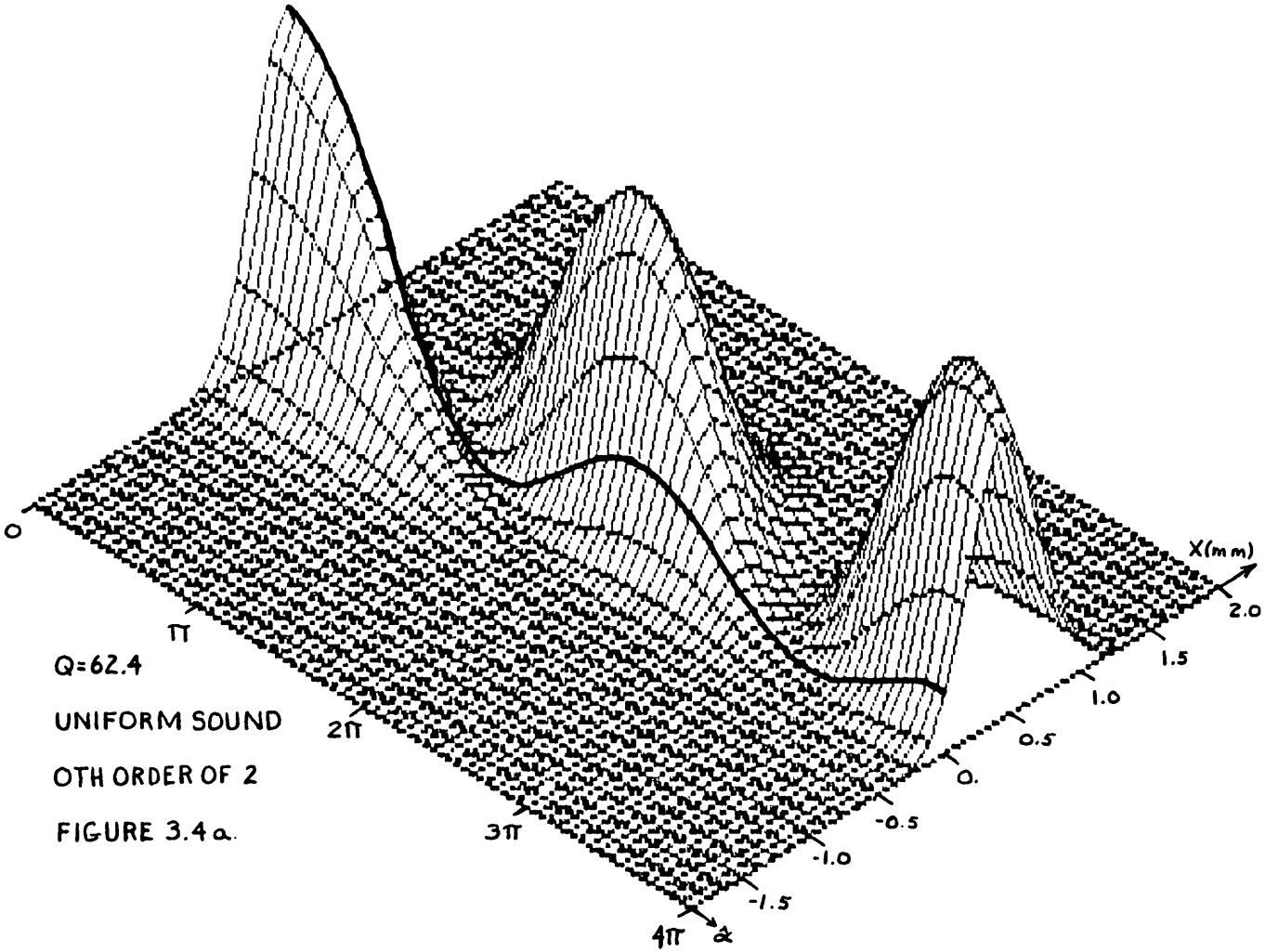


Q=O

UNIFORM SOUND

1 ST ORDER OF 2

FIGURE 3.3b.

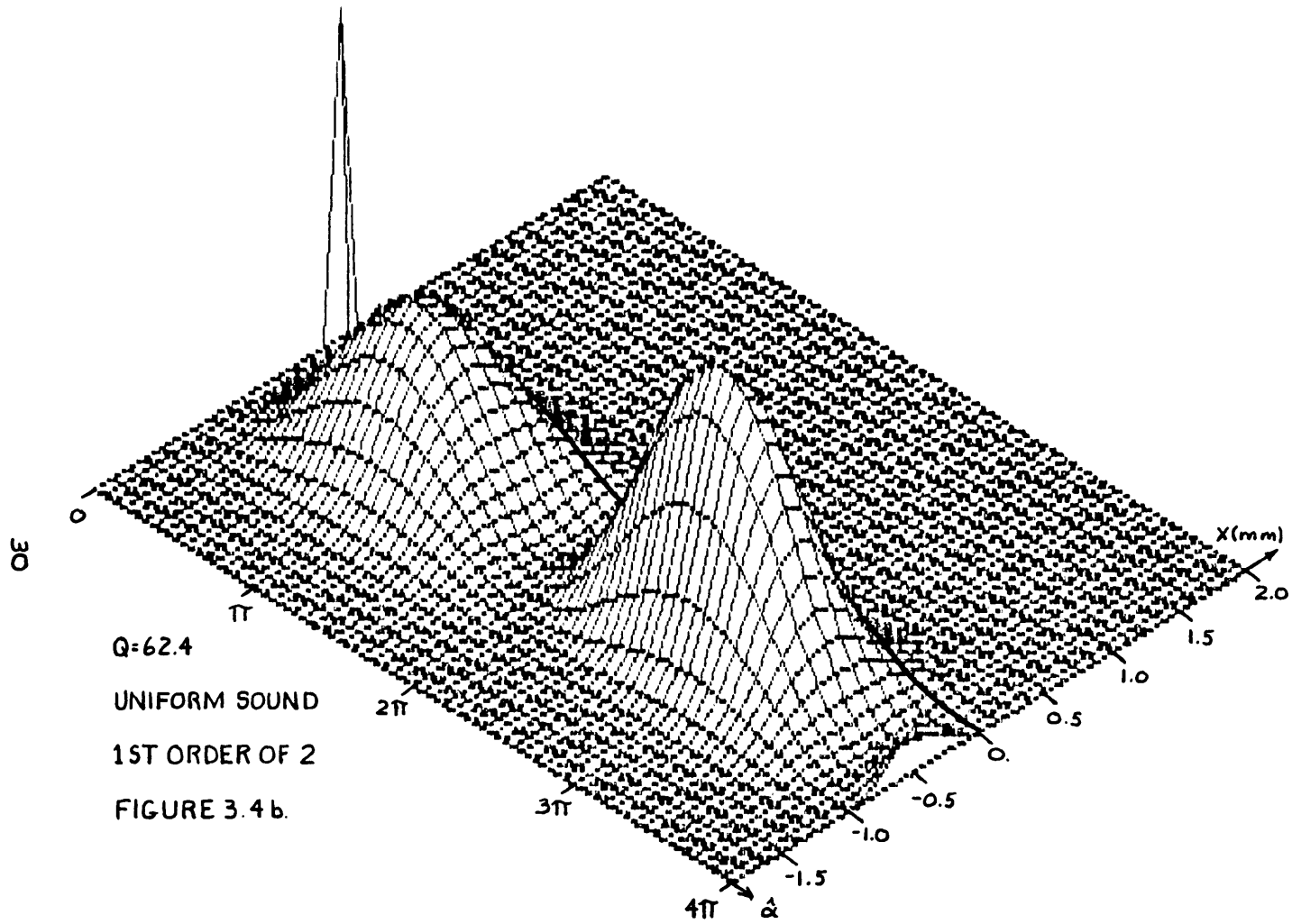


Q=62.4

UNIFORM SOUND

OTH ORDER OF 2

FIGURE 3.4a.

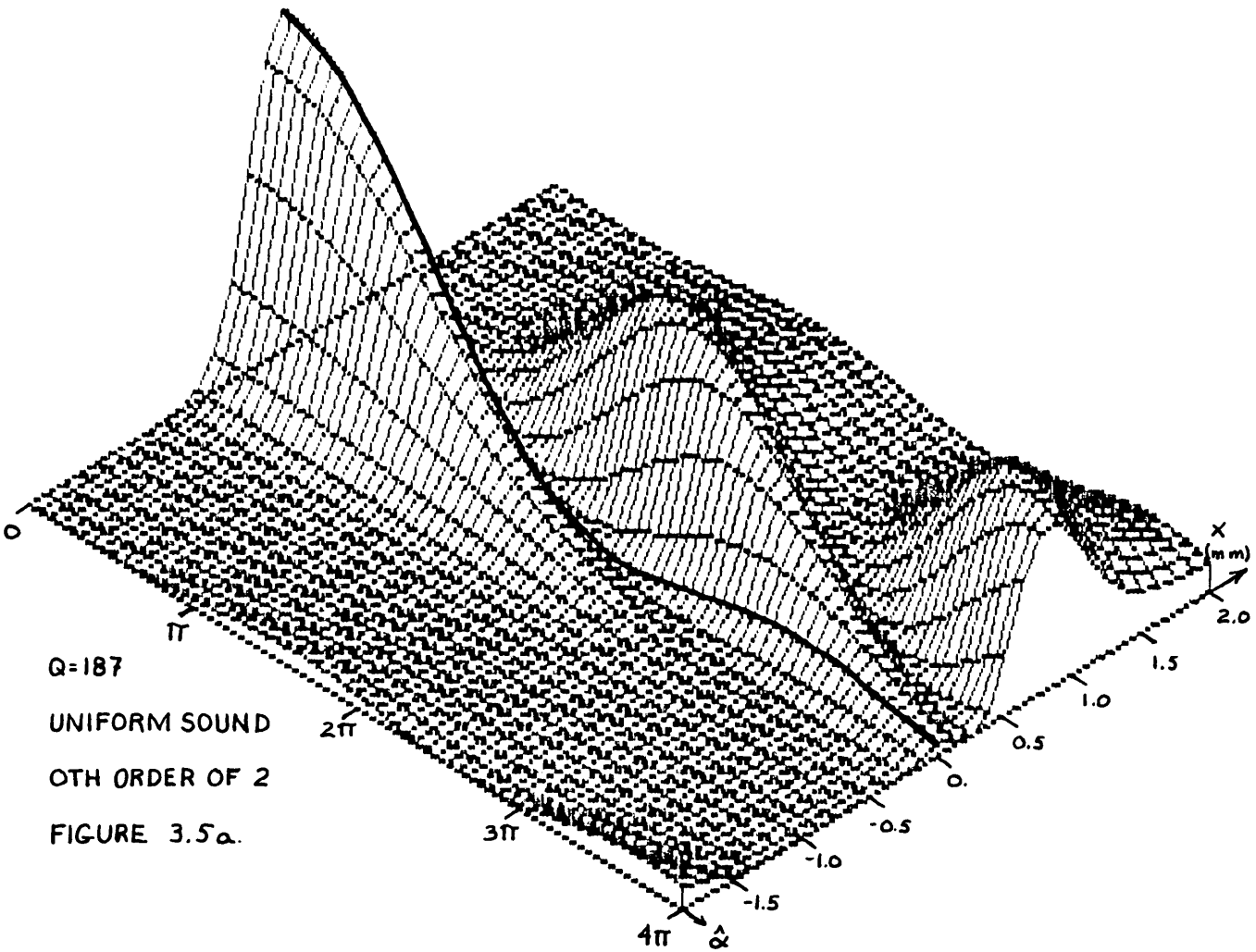


Q=62.4

UNIFORM SOUND

1ST ORDER OF 2

FIGURE 3.4 b.

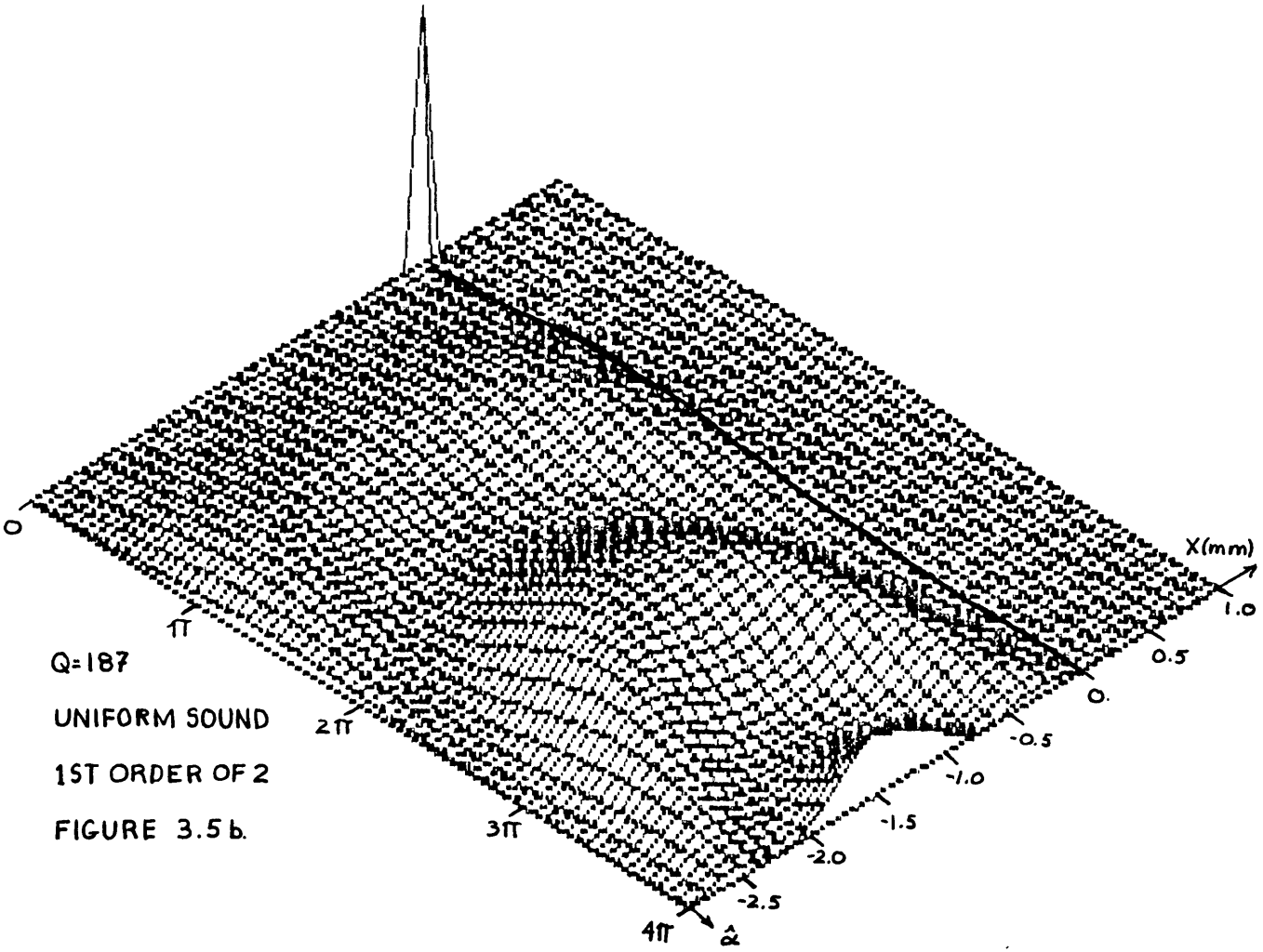


Q=187

UNIFORM SOUND

OTH ORDER OF 2

FIGURE 3.5a.



Q=187

UNIFORM SOUND

1ST ORDER OF 2

FIGURE 3.5 b.

The results agree well with Moharam et al. for both the graph shapes and the movement of the axis of symmetry for each order away from its original center. The graphs also show the gradual motion of the two orders towards each other observed by Forshaw (17). The very high Q values used here at first seem ridiculous; for the 40 MHz sound frequency the sound source length for Q = 187 would almost 1/2 meter. However, the sound frequency can be in the GHz range. The transducer length at a sound frequency of 1 GHz and Q = 187 would only be 0.7 mm. At such frequencies much higher Q values would be reasonable. The graphs presented here are valid for any sound frequency since the system is normalized. What would change is the relationship between g and Q; for g = 3 at 1 GHz, Q = 4677. What is unreasonable is the assumption that only two orders of light will exist for the low values of Q. As mentioned in Chapter 1, low Q values represent Debye-Sears region operation where the output intensities are proportional to Bessel functions.

Chapter 4

Bragg Diffraction of Gaussian Beams: A Four Order Approximation

The equations describing acousto-optic interaction developed in chapter 2 suggest that the amount of light in any diffracted light order is dependent on the amount of light in the orders adjacent to it. Indirectly, every order is affected by every other light order on both sides of it, therefore the infinite set of coupled equations. Since an infinite set of complex coupled differential equations is mathematically cumbersome, some approximations are usually made. In this case the higher orders of light, both positive and negative, are assumed to be zero. Since there is seldom enough sound energy at broad angles to transfer light energy from the incident light order into the higher orders, ignoring those orders is reasonable. The question then arises. How many orders does one assume non-zero? Moharam et al. assumes all but the central two orders to be zero for the Bragg diffraction case (19). For very thick gratings this is a reasonable assumption. The angular spectrum for a thick grating is virtually a plane wave so most light is in the first two orders. However, for thin gratings the two light order approximation is insufficient. For the case of the bounded light beam many light plane waves are considered for each light order. This would

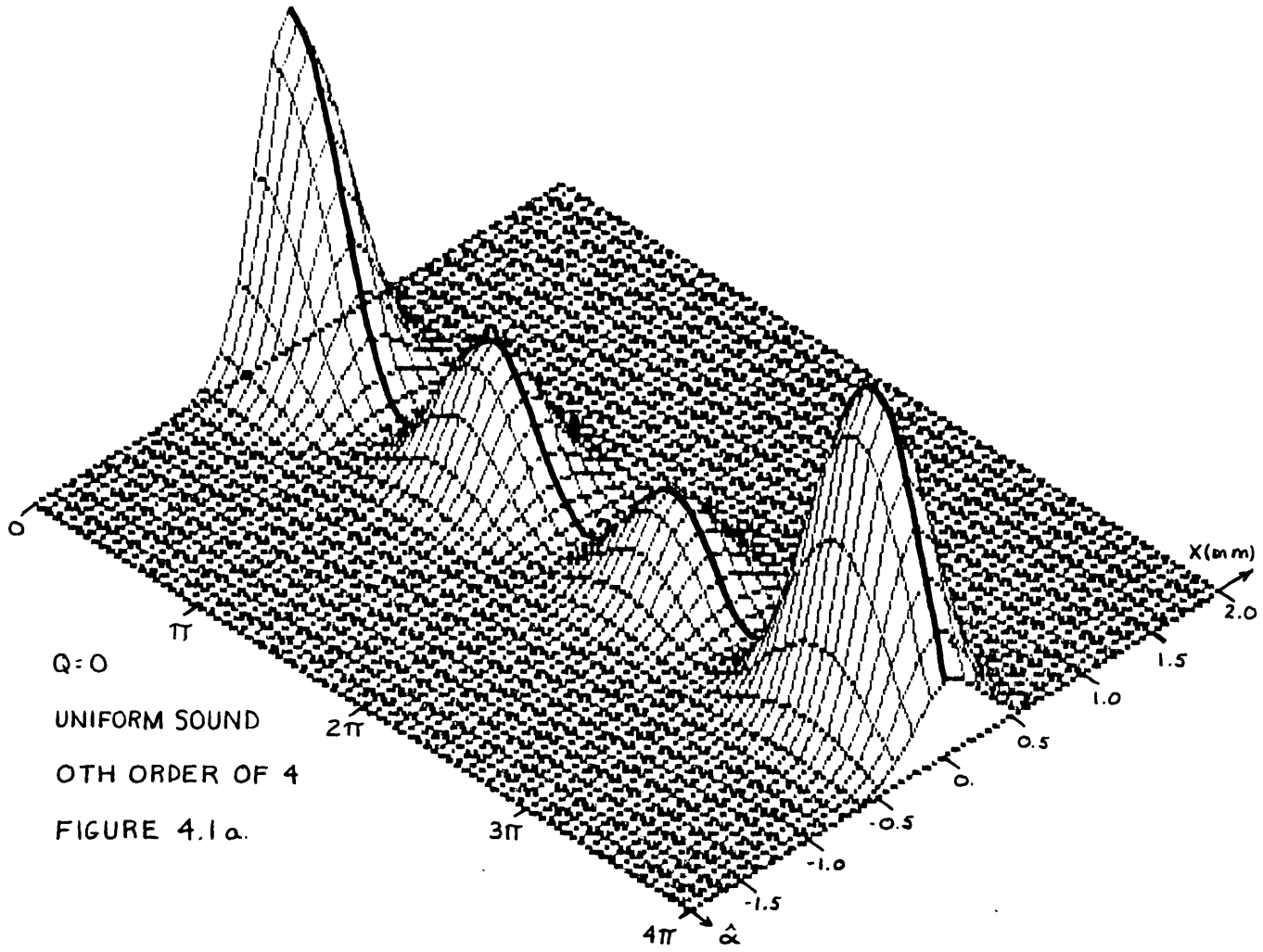
suggest the use of more light orders, even for thick gratings, to avoid compounding error.

Moving up in order of approximation from two would require adding the same number of orders to either side of the original two. For a four order approximation with a light upshifting incident angle, the non-zero light orders would be -1, 0, 1 and 2. The four differential equations describing the case would be, from (3-4):

$$
\begin{aligned}
d\bar{E}_{-1}/d\xi &= -j\hat{\alpha}/2 \{\exp[jQ\xi/2\,(\phi_i/\phi_B - 1)]\,\bar{E}_0\} \\
d\bar{E}_0/d\xi &= -j\hat{\alpha}/2 \{\exp[-jQ\xi/2\,(\phi_i/\phi_B - 1)]\,\bar{E}_{-1} + \\
&\qquad \exp[jQ\xi/2\,(\phi_i/\phi_B + 1)]\,\bar{E}_1\} \\
d\bar{E}_1/d\xi &= -j\hat{\alpha}/2 \{\exp[-jQ\xi/2\,(\phi_i/\phi_B + 1)]\,\bar{E}_0 + \\
&\qquad \exp[jQ\xi/2\,(\phi_i/\phi_B + 3)]\,\bar{E}_2\} \\
d\bar{E}_2/d\xi &= -j\hat{\alpha}/2 \{\exp[-jQ\xi/2\,(\phi_i/\phi_B + 3)]\,\bar{E}_1\}.
\end{aligned}
\qquad (4\text{-}1)
$$

The initial conditions are zero for all light orders except E_0. The field E_0 is initially a plane wave with amplitude given by (3-12), where $\Theta = 0$, the center of the Gaussian, corresponds to the upshifting Bragg angle, $\phi_i = \phi_B$, just as in the cases given in chapter 3. The graphs of numerical solutions to (4-1) for the 0th and 1st light orders with Q values of 0, 62.4 and 187 are given in figures 4.1, 4.2 and 4.3. These cases correspond to the two order approximation graphs in figures 3.3, 3.4 and 3.5.

The differences between the two and four order

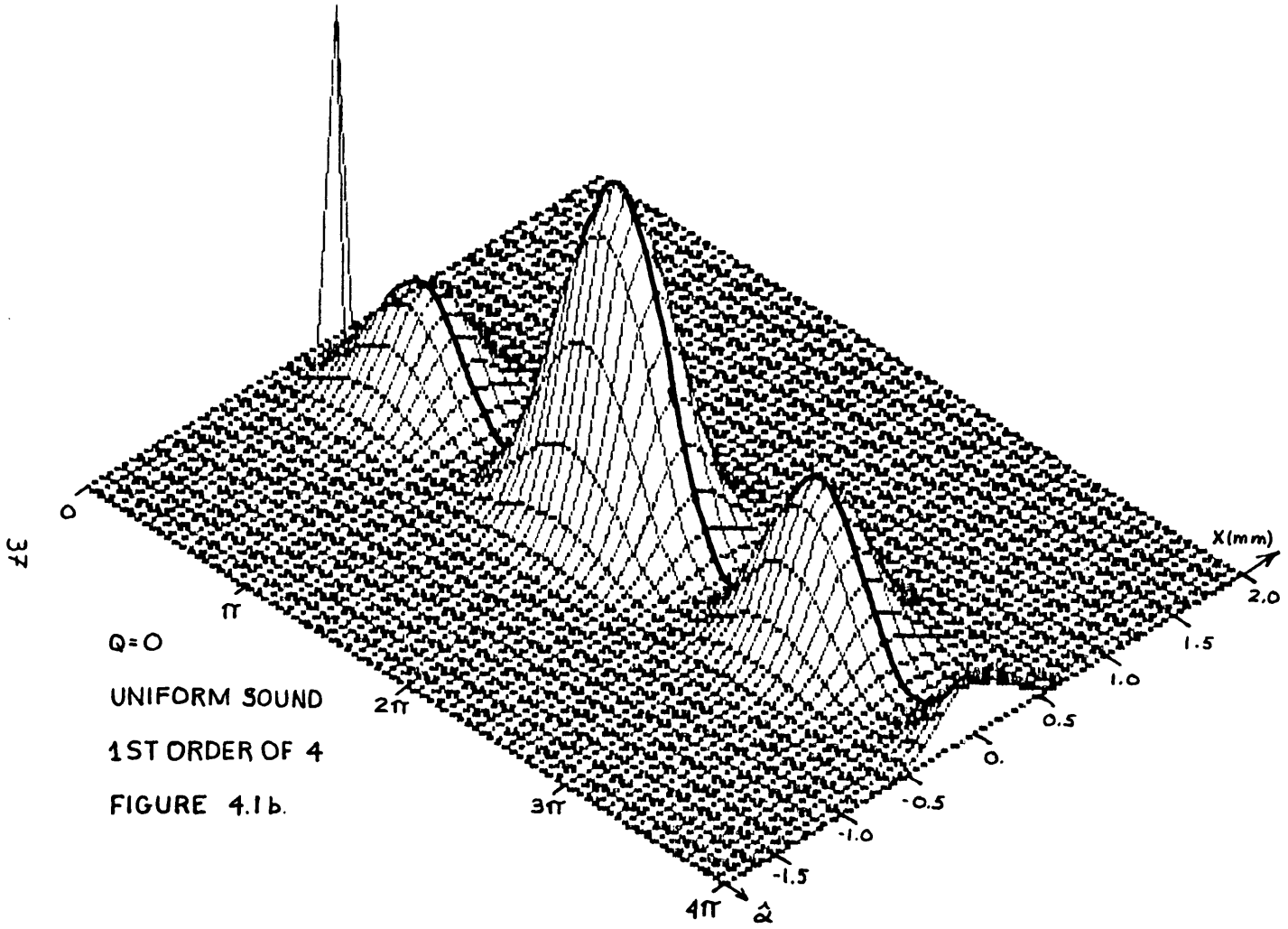


Q=0

UNIFORM SOUND

OTH ORDER OF 4

FIGURE 4.1a.

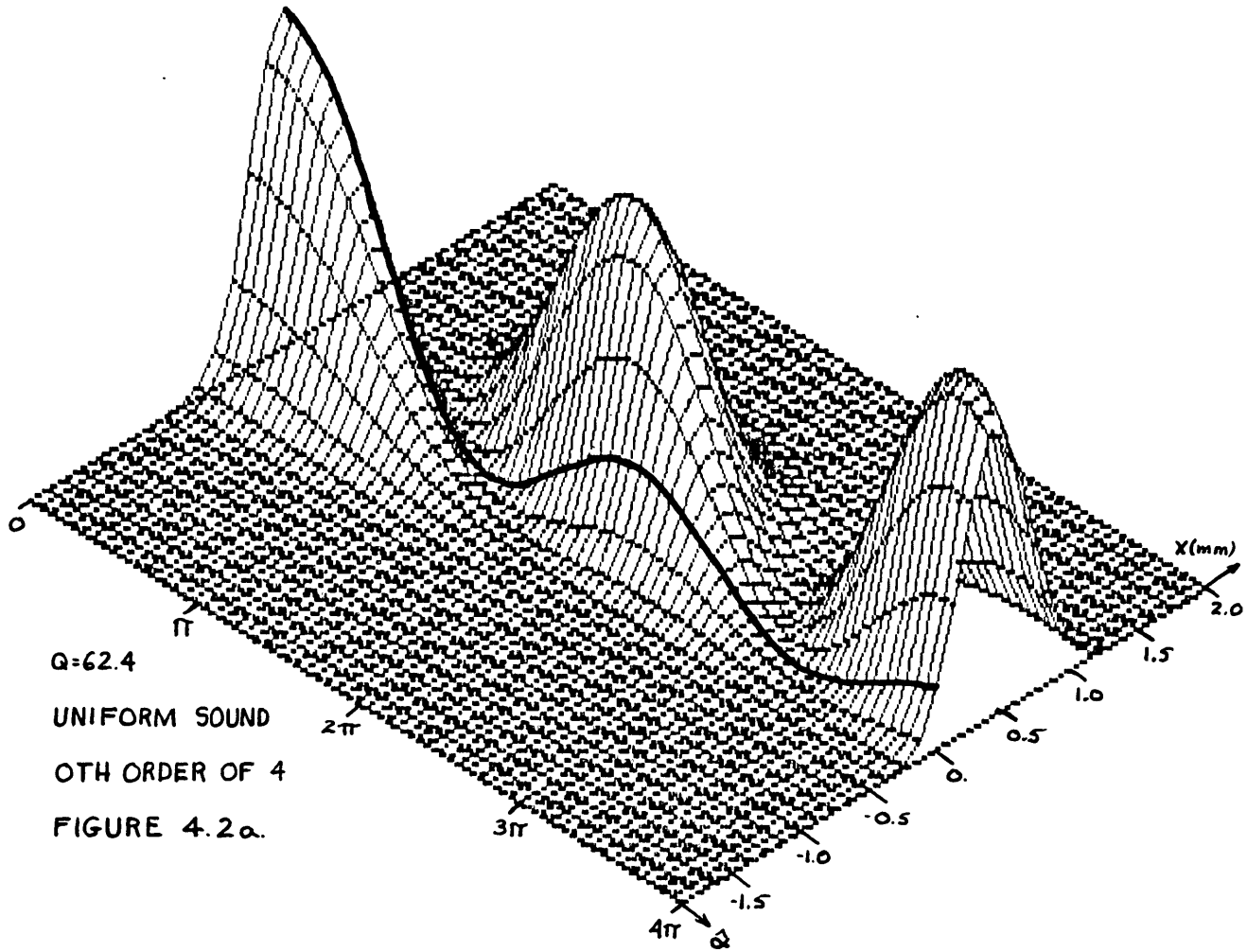


Q=0

UNIFORM SOUND

1ST ORDER OF 4

FIGURE 4.1b.

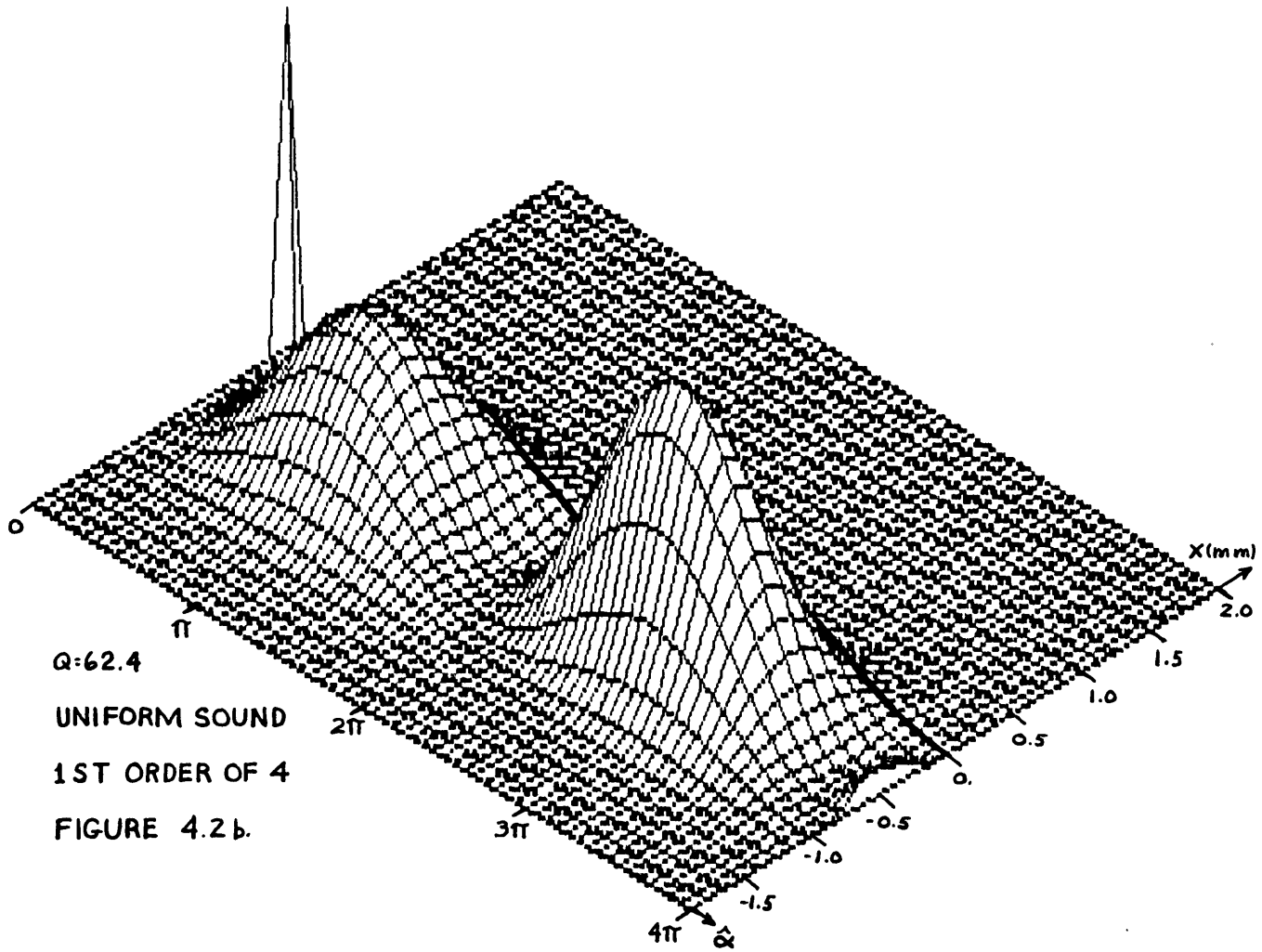


Q=62.4

UNIFORM SOUND

OTH ORDER OF 4

FIGURE 4.2a.

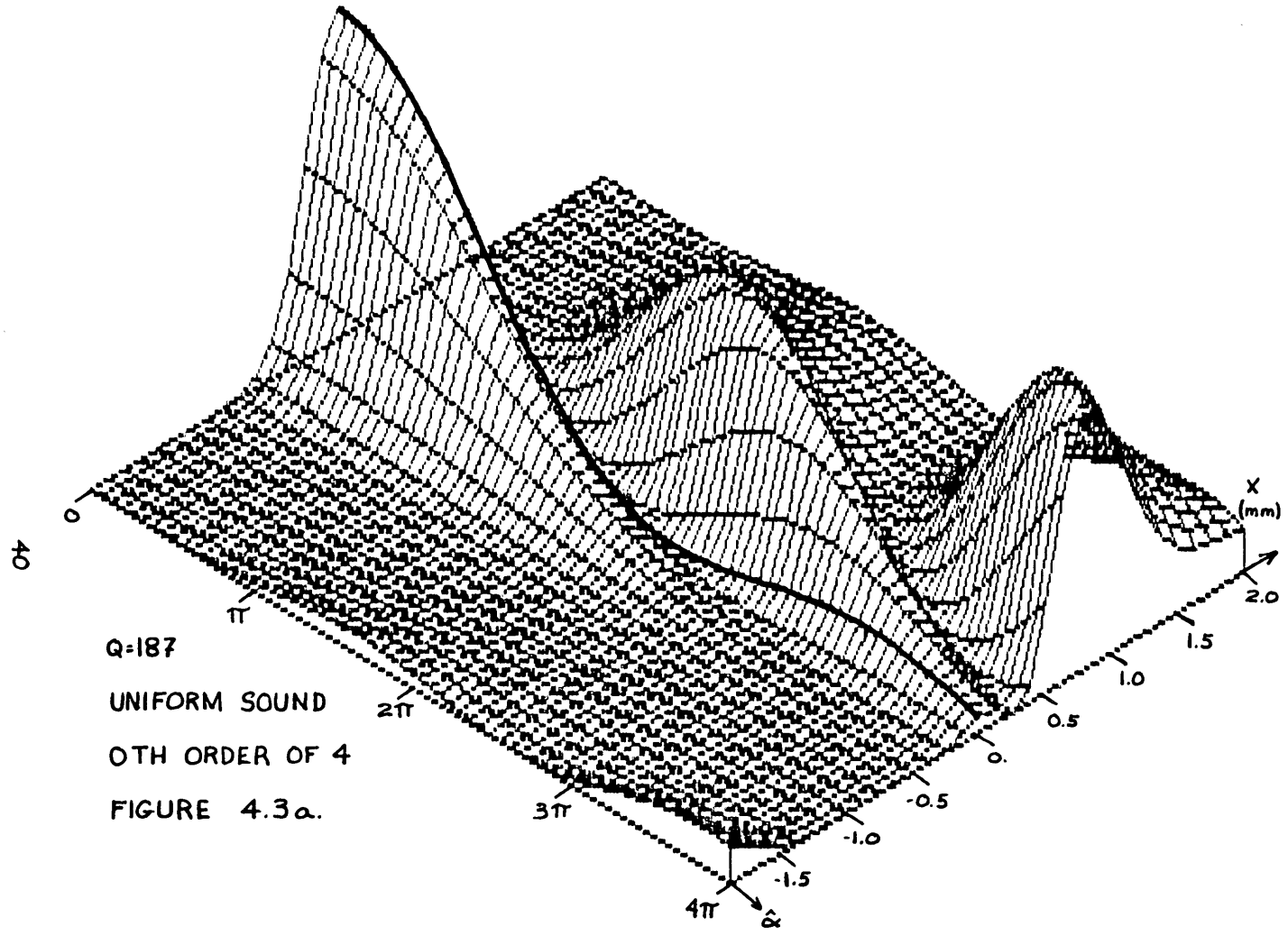


Q:62.4

UNIFORM SOUND

1ST ORDER OF 4

FIGURE 4.2b.

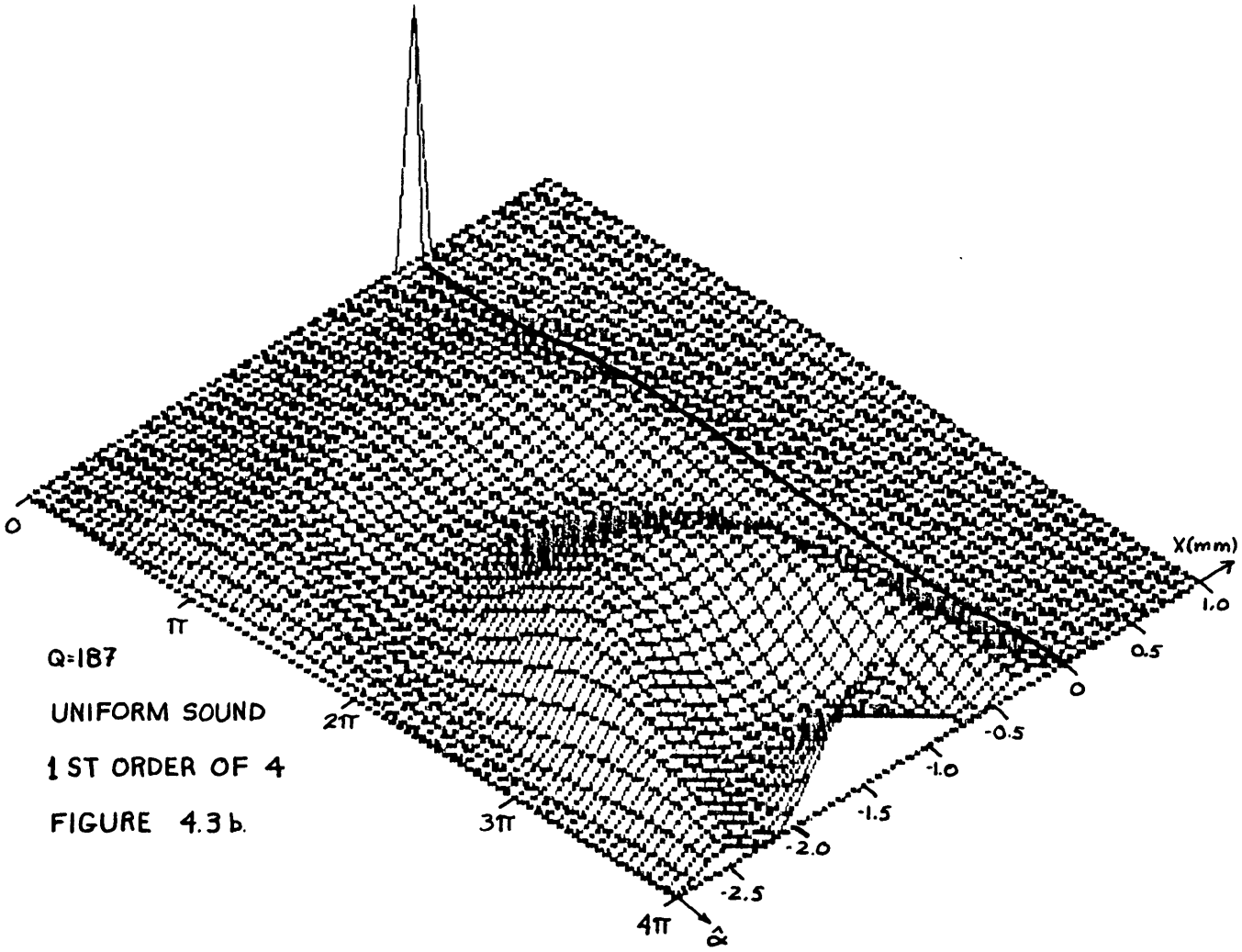


Q=187

UNIFORM SOUND

OTH ORDER OF 4

FIGURE 4.3a.



Q=187

UNIFORM SOUND

1 ST ORDER OF 4

FIGURE 4.3 b.

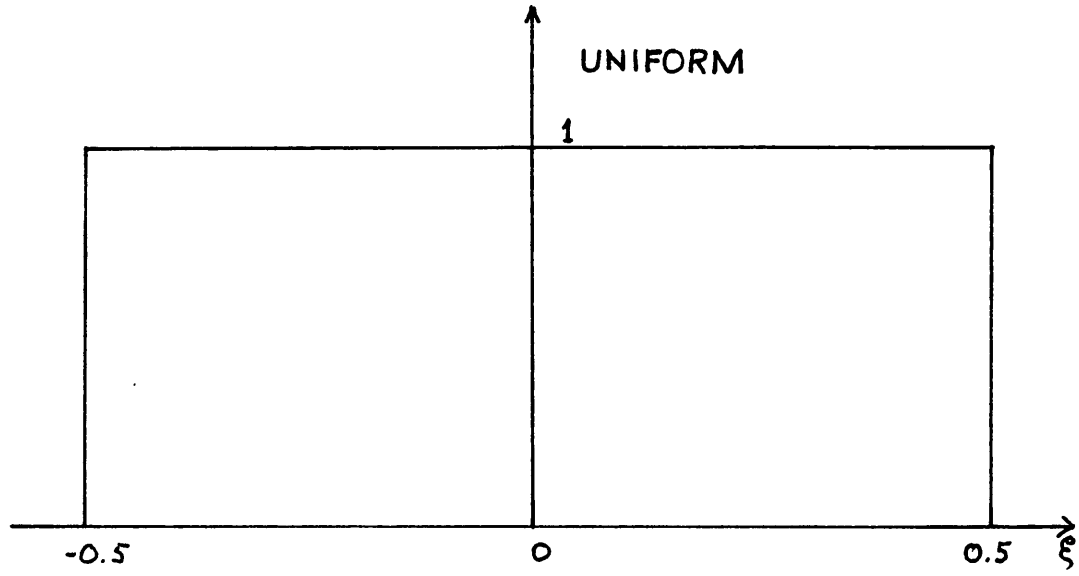
approximations are easily seen when comparing figures 3.3 and 4.1. For the Q = Ø case, the graphs look very little like each other except for the initial peak of the Øth order, which is the input Gaussian. Provided the input beam shape stays the same, the Øth order should always be the same at $\hat{\alpha}$ = Ø since the light travels straight through the unperturbed medium. The low peaks, beyond the first, in figures 4.1a and 4.1b, indicate a substantial amount of light being transferred into the higher orders, orders -1 and 2. If the sound were allowed to increase beyond the level corresponding to $\hat{\alpha}$ = 4π a six order approximation would probably be worth investigating. For the Q = 62.4 and 187 cases the graphs have shapes similar to the two order graphs. Close comparison of figures 4.2a and 3.4a, which show the Øth order at Q = 62.4, shows that the four order case has slightly lower levels throughout the second and third peaks. Slightly lower peaks also appear on the 1st order graph, figure 4.2b, when compared to figure 3.4b. Both Øth and 1st order graphs being lower in intensity shows that there are still small but noticeable light energies in orders -1 and 2. In the Q = 187 case, figures 3.5 and 4.3, the four order graphs have different light distributions from the two order graphs. The third peak in the Øth order is actually higher in the four order graph than the two order graph. At such a high Q level the light quantity in

the outer orders is very small, but they cannot be neglected because they affect the way the light moves in the inner orders. It is also interesting to note that the extra light orders in the approximation have no visible affect on how far the light beam centers move from their original positions as the sound level increases. The lateral motion of the light beam is evidently a function of Q and $\hat{\alpha}$, but not a function of the number of orders used in the approximation.

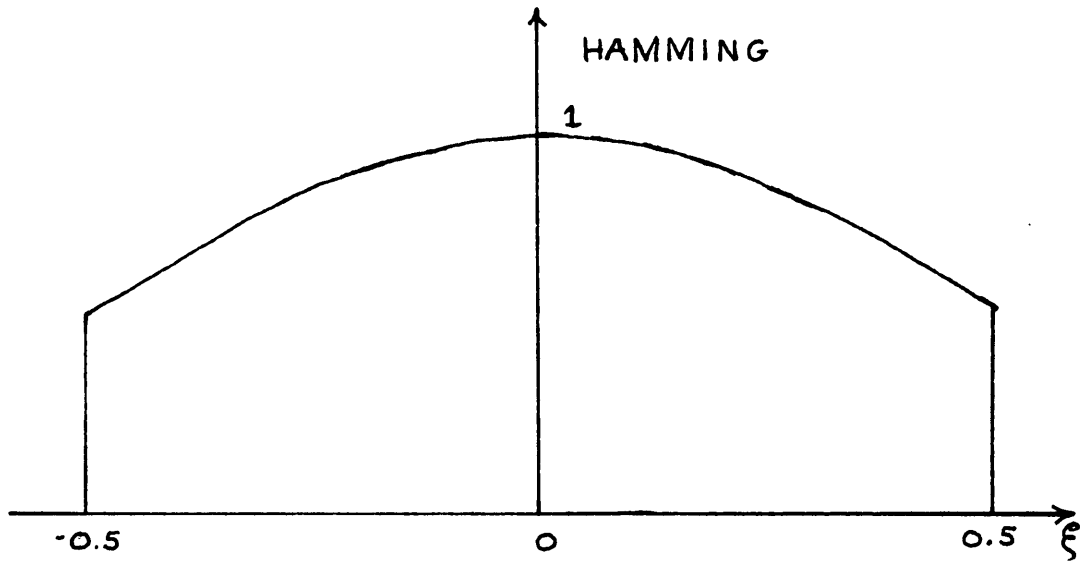
Changing the length of the transducer is not the only way of changing the shape of the sound radiation pattern. Changing the amplitude distribution across the transducer will also change the radiation pattern. So far in this study the sound distribution has been taken to be uniform, i.e., the sound amplitude has been unity across the whole transducer as shown in figure 4.4a. An alternate distribution, called the Hamming function, is of interest because the size of the radiation pattern side-lobes is diminished. This puts more sound energy into the central lobe, thereby reducing the amount of sound available to move light into the higher light orders. The Hamming function is graphed in figure 4.4b and is expressed mathematically as:

$$\text{Hamming}(x) = 0.54 + 0.46 \cos(\pi x) \qquad (4\text{-}2)$$

Since the Hamming function modifies the sound amplitude $|S|$,



a.



b.

UNIFORM AND HAMMING DISTRIBUTIONS

FIGURE 4.4

it is inserted into the interaction formulae (4-1) by multiplying it with $\hat{\alpha}$. For the sound cell running from $\xi = 0$ to 1, (4-1) becomes:

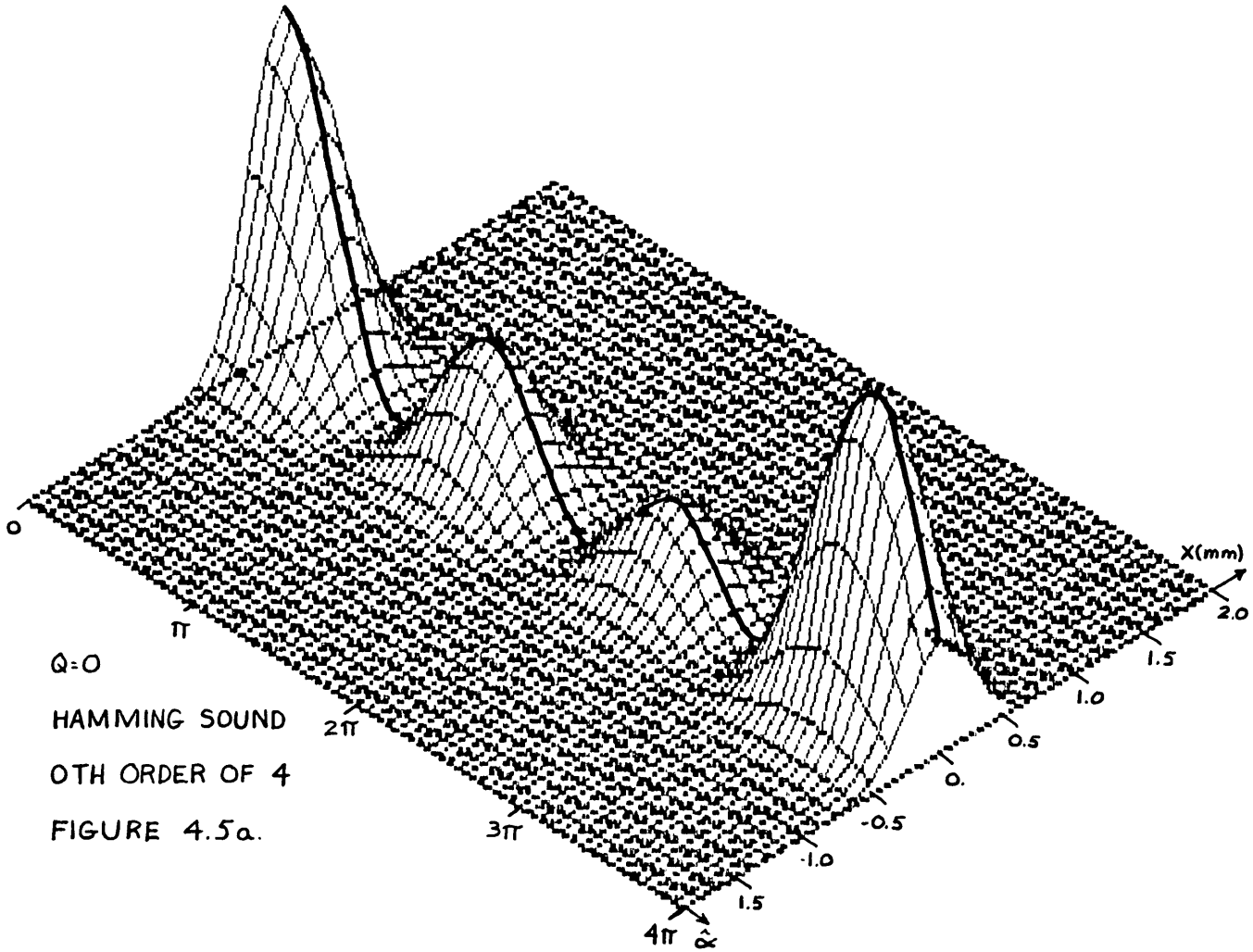
$$\begin{aligned}
d\bar{E}_{-1}/d\xi &= -j\hat{\alpha}\,ham/2\ \{\exp[jQ\xi/2\ (\phi_i/\phi_B - 1)]\ \bar{E}_0\} \\
d\bar{E}_0/d\xi &= -j\hat{\alpha}\,ham/2\ \{\exp[-jQ\xi/2\ (\phi_i/\phi_B - 1)]\ \bar{E}_{-1} + \\
&\qquad \exp[jQ\xi/2\ (\phi_i/\phi_B + 1)]\ \bar{E}_1\} \\
d\bar{E}_1/d\xi &= -j\hat{\alpha}\,ham/2\ \{\exp[-jQ\xi/2\ (\phi_i/\phi_B + 1)]\ \bar{E}_0 + \\
&\qquad \exp[jQ\xi/2\ (\phi_i/\phi_B + 3)]\ \bar{E}_2\} \\
d\bar{E}_2/d\xi &= -j\hat{\alpha}\,ham/2\ \{\exp[-jQ\xi/2\ (\phi_i/\phi_B + 3)]\ \bar{E}_1\}\ ,
\end{aligned} \tag{4-3}$$

where:

$$ham = 0.54 + 0.46\sin(\pi\xi)\ . \tag{4-4}$$

Plots for the cases shown in figures 4.1, 4.2 and 4.3 using four light orders and the Hamming sound distribution are in figures 4.5, 4.6 and 4.7.

Comparison of figures 4.1 and 4.5 show no visible differences. This makes sense since the Q = 0 case implies an infinitely thin sound beam, where no real amplitude distribution can exist. Any difference between these graphs would be artifacts of the computer programming. The graphs for the higher Q values show more interesting differences. The graphs for both the 0th and 1st orders, in both the Q = 62.4 and Q = 187 cases, are slightly higher. Less light is escaping into the higher light orders. This shows that



Q=0

HAMMING SOUND

OTH ORDER OF 4

FIGURE 4.5a.

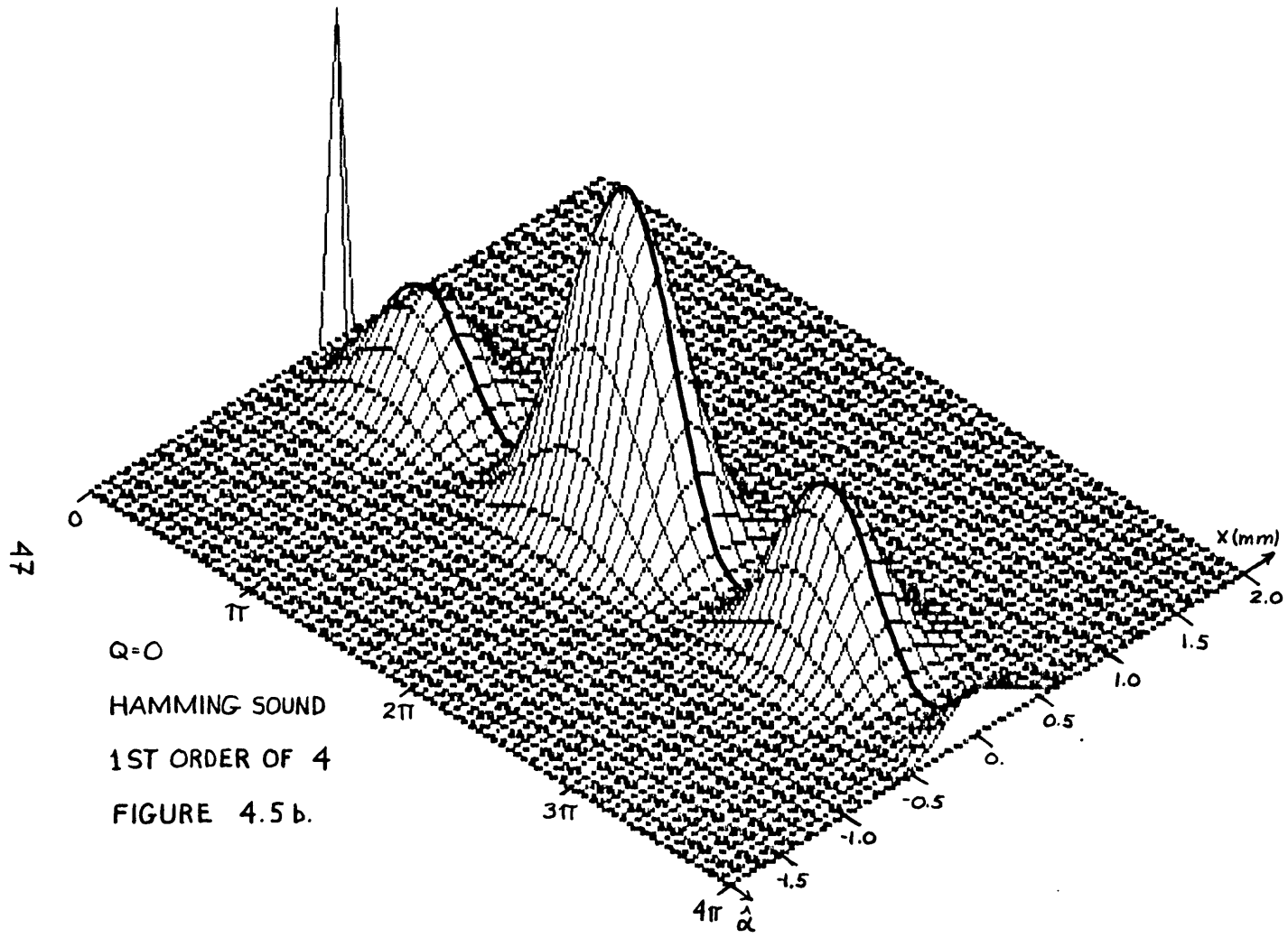
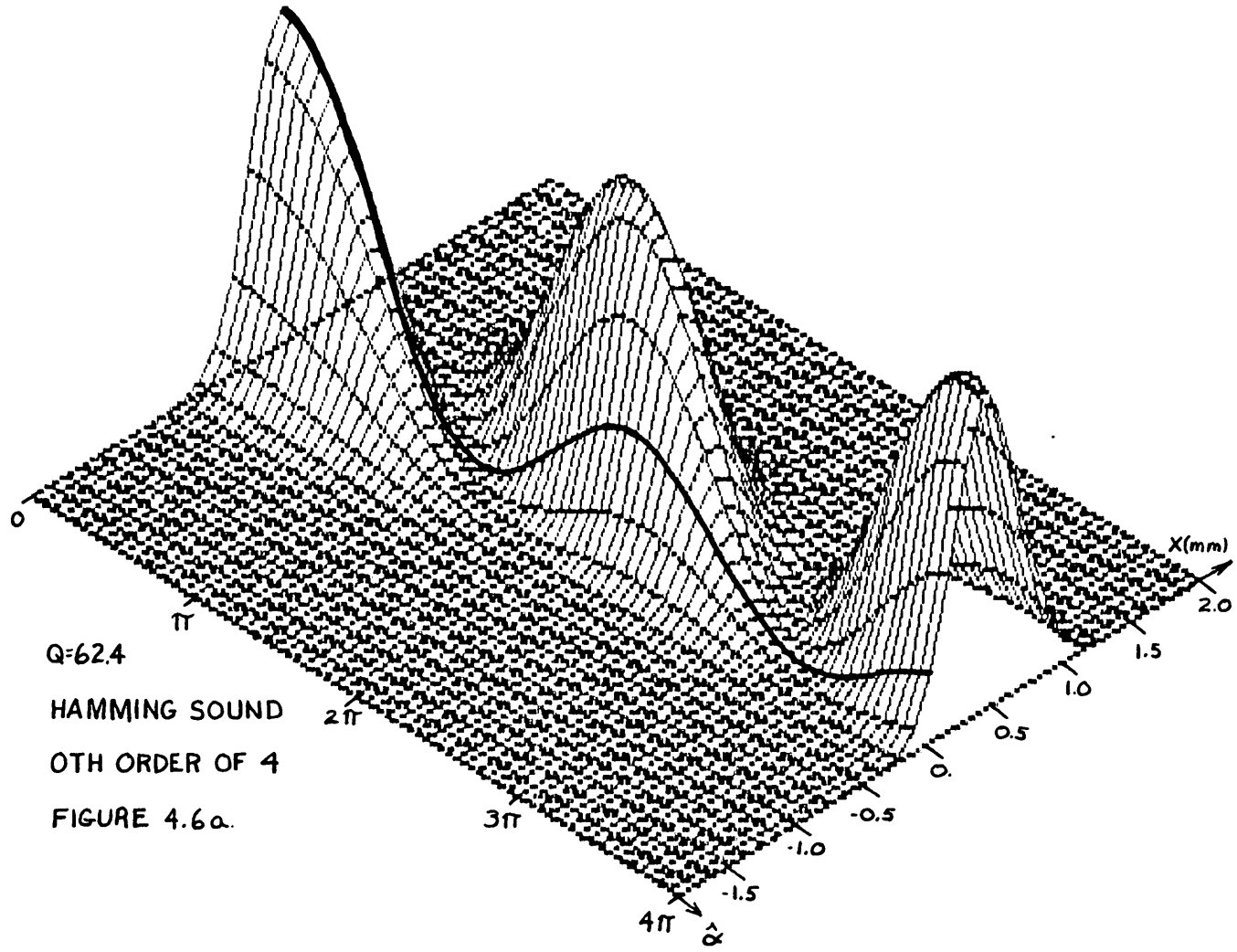

47
0
π
2π
3π
4π
$\hat{\alpha}$
X(mm)
2.0
1.5
1.0
0.5
0.
-0.5
-1.0
-1.5
Q=0
HAMMING SOUND
1ST ORDER OF 4

FIGURE 4.5 b.

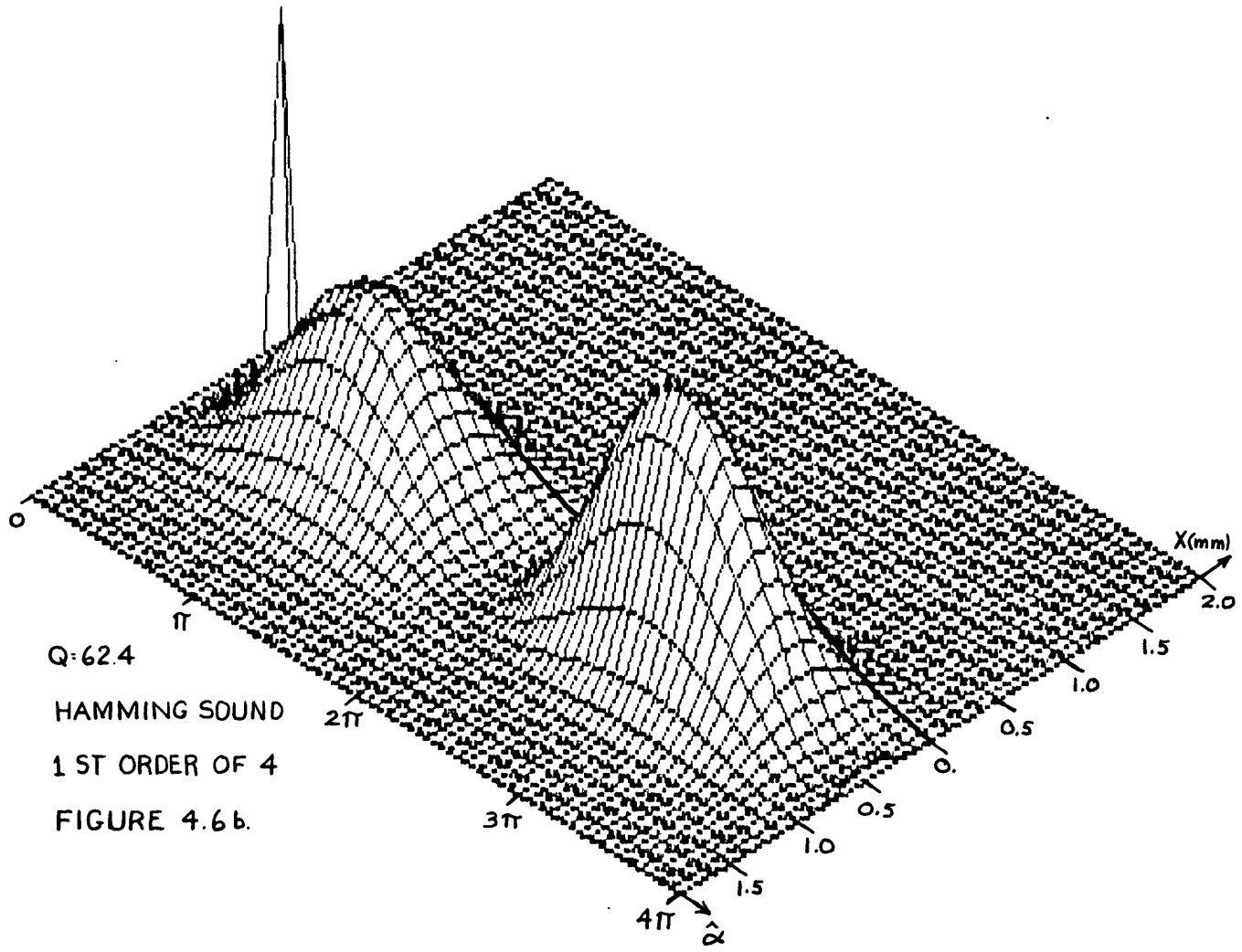


Q=62.4

HAMMING SOUND

OTH ORDER OF 4

FIGURE 4.6a.

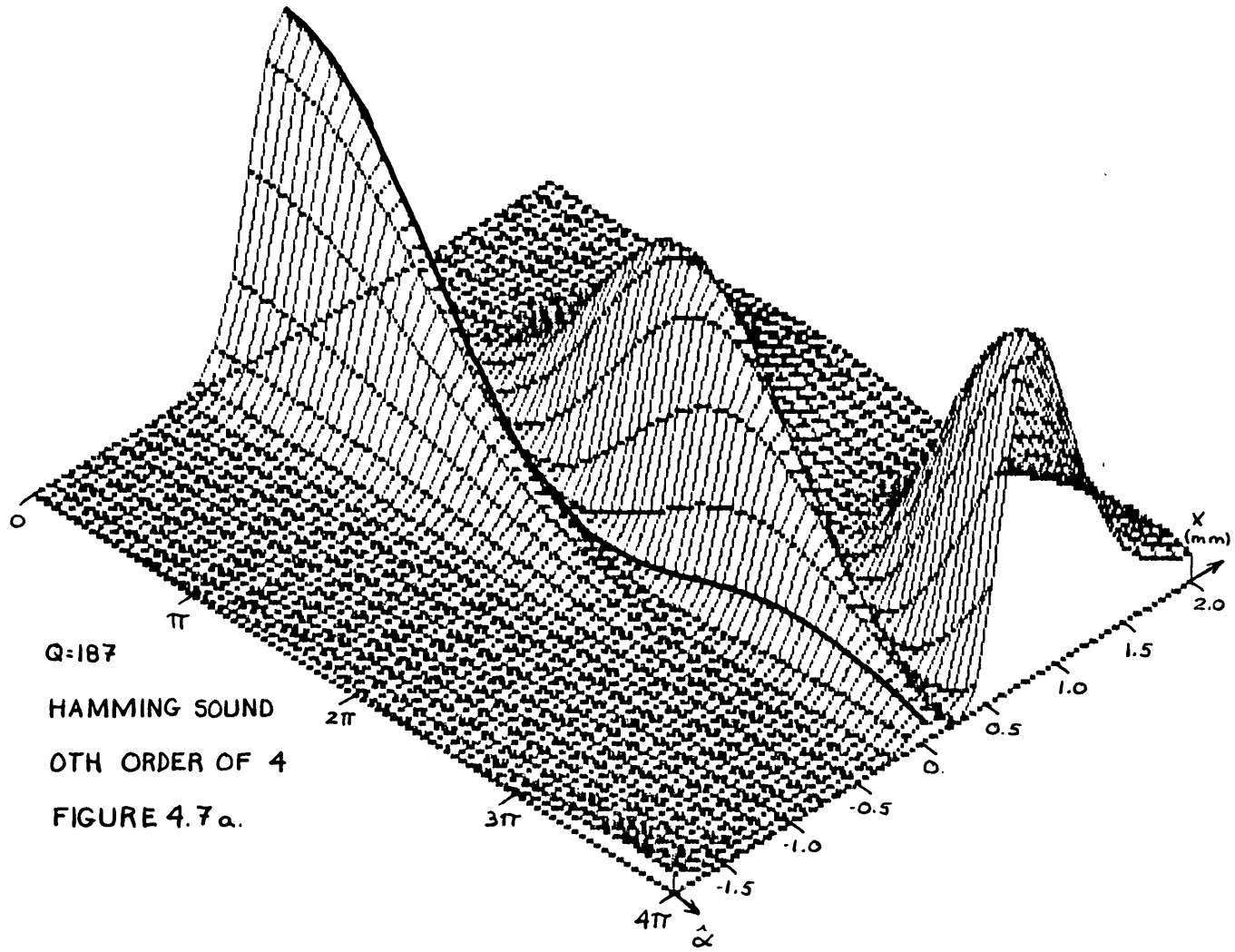


Q=62.4

HAMMING SOUND

1 ST ORDER OF 4

FIGURE 4.6 b.



Q=187

HAMMING SOUND

OTH ORDER OF 4

FIGURE 4.7a.

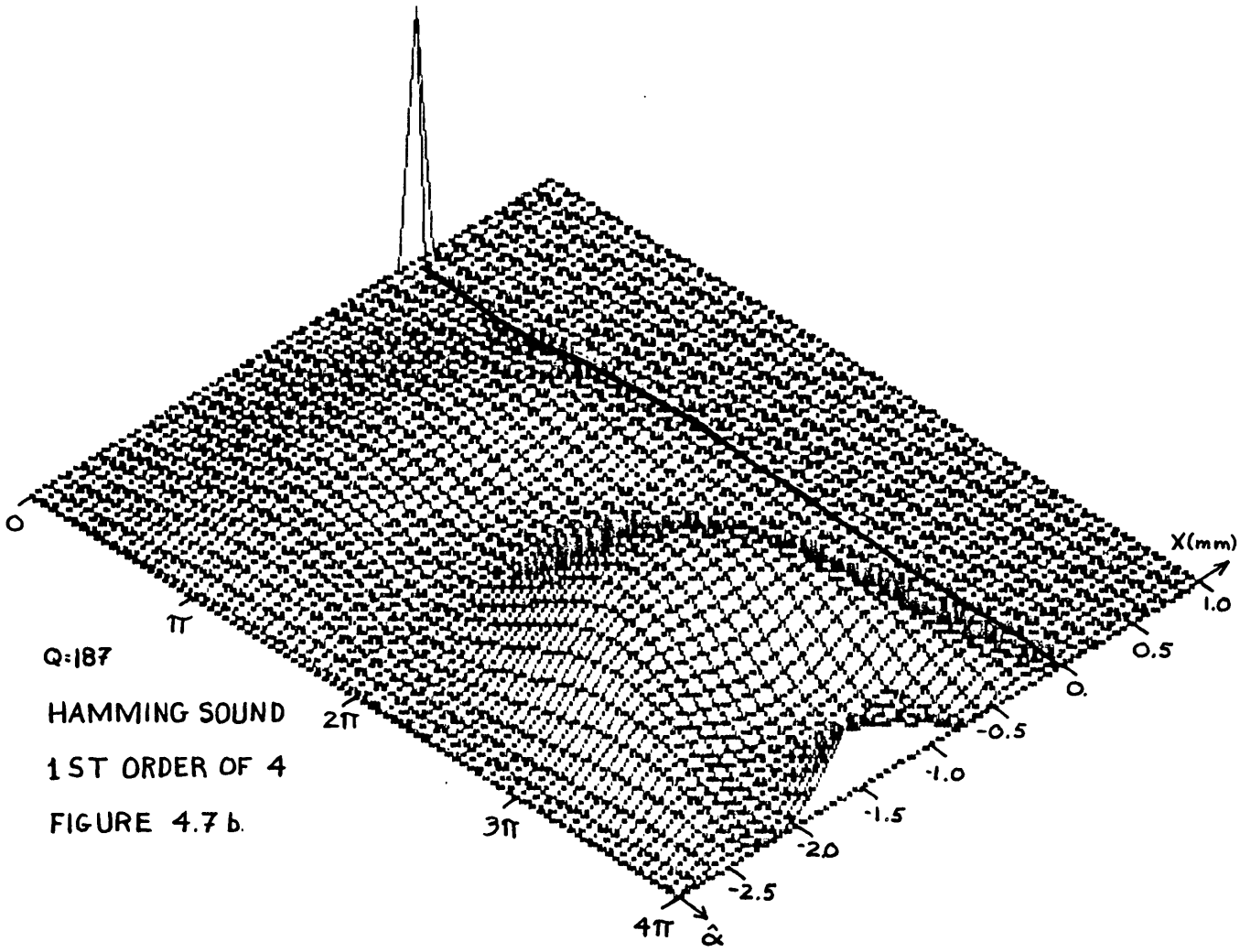

0
π
2π
3π
4π
α̂
X(mm)
1.0
0.5
0.
-0.5
-1.0
-1.5
-2.0
-2.5
Q:187
HAMMING SOUND
1 ST ORDER OF 4

FIGURE 4.7 b.

the Hamming distribution does collimate the sound better than the uniform distribution. The increases are not dramatic, but at these high Q levels not much light is in the -1st and 2nd orders even for the uniform sound source. Furthermore, the Hamming distribution does have an effect on the lateral motion of the light beam centers. The peaks have moved less with the Hamming source than with the uniform source, noting specifically the Øth order at Q = 187, in figures 4.3a and 4.7a. The peak for the uniform sound source has moved 8 lines to the right compared to a 7 line shift in the Hamming case, a difference of about 1/8 mm in a 1 mm diameter light beam. The lateral shift of the light orders can evidently be changed by either change in the sound source length or amplitude distribution.

Chapter 5

Summary and Conclusions

In this study the multiple plane wave scattering model was used to solve the problem of acousto-optic interaction of a Gaussian input light beam. Using a uniform sound source and two orders of light only, this model produced results very close to those found by Moharam, Gaylord and Magnusson (19). Assuming they were correct, Gaussian light beams are within the bounds of acousto-optic problems solvable using the multiple scattering model.

Extension of the bounded light beam problem to four orders of light showed how ineffective the two light order approximation is for thin sound beams. The two light order case is simply too idealistic for thin gratings. The four light order results for thicker gratings showed some differences in light beam shape and intensity from the two order results, but the differences were small compared to the amount of distortion the beams show as the sound strength increases. Judging by the way the beam distortion develops with rising Q (K^2L/k) and sound level, its effects can be mostly avoided by operating at sound levels corresponding with the light intensity peaks. At the peaks the distortion is small, provided that the Q value is not too high. Lower Q values, unfortunately, while having less distortion, also have higher light energies in the higher

orders. A trade-off exists between the amount of light beam distortion, and how much light is lost to the higher orders.

An alternative to the trade-off is to find another way of controlling the system. The Hamming distribution is introduced, as the sound source amplitude profile, to lower the amount of light leaking into the higher orders. The graphs in chapter 4 show clearly that although the central light orders are a little brighter, the Hamming function is not a good way to control distortion of the beams. The change is just not enough. The fact that there is a change suggests that other sound distributions might be more effective. Also, the plots only show results for two Q values, the Hamming function might do better for different Q values.

Lastly, the Q values used in this study for thick gratings are very high (though they could be much higher for higher sound frequencies). At such high Q values the assumption that the sound beam is a uniform column is reasonable. As the Q value lowers, i.e., the length of the sound source gets smaller compared to the sound wavelength, the sound field that the light beams will encounter will look more and more like the spectrum shown in figure 1.3. The methods shown here are very effective for high Q values, sound fields like figure 1.3 should be used for analysis of the lower Q values.

References

1. L. Brillouin, "Diffusion de la lumiere et des rayons X par un corps transparent homogene," Ann. Phys. (Paris), vol. 17, pp. 88-122, 1922.

2. R. Lucas and P. Biquard, "Proprietes optiques des milieux solides et liquides soumis aux vibration elastques ultra sonores," J. Phys. Rad., vol. 3, pp. 464-477, 1932.

3. P. Debye and Sears, "On the scattering of light by supersonic waves," Proc. Nat. Acad. Sci. (U.S.), vol. 18, pp. 409-414, June 1932.

4. A. Korpel, "Acousto-Optics -- A Review of Fundementals," Proc. IEEE, vol. 69, pp. 48-53, Jan. 1981.

5. "Antenna Theory and Design," W. Stutzman and G. Thiele, John Wiley & Sons, Inc., 1981, chap. 1, pp. 26-29.

6. W. Klein and B. Cook, "Unified Approach to Ultrasonic Light Diffraction," IEEE Trans. Sonics Ultrason., vol. SU-14, pp. 123-134, July 1967.

7. D. Robinson, "The Supersonic Light Control and its Application to Television with Special Reference to the Scarphony Television Receiver," Proc. IRE, vol.27, pp. 483-486, Aug. 1939.

8. F. Okolicsanyi, "The wave-slot, an optical television system," Wireless Engineer, vol. 14, pp. 527-536, Oct. 1937.

9. A. Korpel, "Acousto-Optics," in "Applied Solid State Science," vol. 3, R. Wolfe Ed., Academic Press, New York and London, 1972, chap. 2, pp. 72-180.

10. S. Nakadate, "Shearing heterodyne interferometry using acoustooptic light modulators," Appl. Opt., vol. 24, pp. 3079-3087, 15 Sept. 1985.

11. A. VanderLugt, "Crossed Bragg Cell Processors," Appl. Opt., vol. 23, pp. 2275-2281, 15 July 1984.

12. G. Indebetouw and T.-C. Poon, "Incoherent spatial filtering with a scanning heterodyne system," Appl. Opt., vol. 23, pp. 4571-4574, 15 Dec. 1984.

13. T.-C. Poon, "Scanning holography and two-dimensional image processing by acousto-optic two-pupil synthesis," J. Opt. Soc. Am., vol. 2, pp. 521-527, Apr. 1985.

14. T.-C. Poon, "Method of two-dimensional bipolar incoherent image processing by acoust-optic two-pupil synthesis," Opt. Let., vol. 10, pp 197-199, May 1985.

15. Y. Ohtsuka, Y. Arima and Y. Imai, "Acoustooptic 2-D profile shaping of a Gaussian laser beam," Appl. Opt., vol. 24, pp. 2813-2819, 1 Sept. 1985.

16. A. Korpel, "Two dimensional plane wave theory of strong acousto-optic interaction in isotropic media," J. Opt. Soc. Am., vol. 69, pp. 678-683, May 1979.

17. M. Forshaw, "Diffraction of a narrow laser beam by a thick hologram: Experimental results," Opt. Comm., vol. 12, pp. 279-281, Nov. 1974.

18. R.-S. Cnu and T. Tamir, "Bragg diffraction of Gaussian beams by periodically modulated media," J. Opt. Soc. Am., vol. 66, pp. 220-226, Mar. 1976.

19. M. Moharam, T. Gaylord and R. Magnusson, "Bragg diffraction of finite beams by thick gratings," J. Opt. Soc. Am., vol. 70, pp. 300-304, Mar. 1980.

20. C. Raman and N. Nath, "The diffraction of light by high frequency sound waves," Proc. Ind. Acad. Sci., vol. 2, pp. 406-420, 1935.

21. C. Raman and N. Nath, "The diffraction of light by high frequency sound waves," Proc. Ind. Acad. Sci., vol. 3, pp. 75-84, 1936.

22. C. Raman and N. Nath, "The diffraction of light by high frequency sound waves," Proc. Ind. Acad. Sci., vol. 3, pp. 119-125, 1936.

23. C. Raman and N. Nath, "The diffraction of light by high frequency sound waves," Proc. Ind. Acad. Sci., vol. 3, pp. 459-465, 1936.

24. A. Bhatia and W. Noble, "Diffraction of light by ultrasonic waves," Proc. Roy. Soc., Sec. A, vol. 220, pp. 356-368, 1953.

25. A. Korpel, "Acousto-Optics," in "Applied Solid State Science," vol. 3, R. Wolfe Ed., Academic Press, New York and London, 1972, chap. 2, parts II and III, pp. 82-87.

26. A. Korpel, "Acousto-Optics," in "Applied Solid State Science," vol. 3, R. Wolfe Ed., Academic Press, New York and London, 1972, chap. 2, part V, pp. 108-115.

27. T.-C. Poon and A. Korpel, "Use of LaPlace transforms in acousto-optic multiple scattering," Opt. Let., vol. 6, pp. 546-548, Nov. 1981.

28. A. Korpel and T.-C. Poon, "Explicit formalism for acousto-optic multiple plane-wave scattering," J. Opt. Soc. Am., vol. 70, pp. 817-820, July 1980.

29. T.-C. Poon and A. Korpel, "Feynman diagram approach to acousto-optic scattering in the near Bragg region," J. Opt. Soc. Am., vol. 71, pp. 1202-1208, Oct. 1981.

30. T.-C. Poon, M. Chatterjee and P. Banerjee, "Multiple plane-wave analysis of acousto-optic diffraction by adjacent ultrasonic beams of frequency ratio 1:m," J. Opt. Soc. Am., pp. 1402-1406, Sept. 1986.

31. "Principles of Acoustic Devices," V. Ristic, John Wiley & Sons, Inc., 1983, chap. 1, p. 7.

32. "Modern Digital and Analog Communication Systems," B. Lathi, CBS College Publishing, 1983, chap. 2, p. 41.

33. "Introduction to Fourier Optics," J. Goodman, McGraw-Hill, Inc., 1968, chap. 3, pp. 48-53.

34. "Numerical Analysis," L. Johnson & R. Reiss, Addison-Wesly Publishing Co., Reading, Mass., 1982, pp. 378-379.

35. "Numerical Analysis," L. Johnson & R. Reiss, Addison-Wesly Publishing Co., Reading, Mass., 1982, Sect. 7.2.3, pp. 366-371.

36. "Numerical Analysis," L. Johnson & R. Reiss, Addison-Wesly Publishing Co., Reading, Mass., 1982, Sect. 6.6, pp. 341-349.

37. "Optical Physics," S. Lipson & H. Lipson, Cambridge Univ. Press, 1981, App. V, Part 2, pp. 424-426.

Appendix

Computer Programming

The physical problem that this study presented was a bounded sound column causing diffraction of a Gaussian shaped light beam over a range of sound amplitudes. The mathematical problem arising from using the Multiple Plane Wave Scattering method to model this physical problem was a system of first order, complex, coupled differential equations to be solved over the interval 0 to 1. The initial values of the light orders were 0 for all orders used but the 0th order which had unit amplitude. The formulation used by the model assumes that the input light is a plane wave, so any non-planar light must first be broken into plane waves at the input and put back into profile form at the output. These solutions must be found repetitively for different sound amplitudes so that the development of the light beam profiles as the sound changes can be seen. The following algorithm was used to solve this problem.

1. Convert the input gaussian to its angular spectrum representation by means of a Fourier transform.

2. Sample the spectrum a sufficient number of times, and at close enough intervals, to allow reconstruction of the shape at the output of the system.

3. Solve the system of equations for each plane wave sample and store the results for each output light order.

4. Find the inverse Fourier transform of the output spectrum to get an output light profile.

5. Square the light profiles to get intensity profiles and store the results.

6. Repeat steps 2 through 5 for each sound amplitude.

7. Plot the intensity profiles against the sound amplitude.

The Fourier transforming of the input Gaussian was performed by hand, the resultant function appears in (3-12) and figure 3.2. The spectrum sampling, differential system solving, inverse transforming, squaring and storing were all done in one computer program, which appears in figure A.1. The sampling of the light beam profile function was done 32 times at 0.15 mrad intervals centered around the upshifting Bragg angle which was the nominal angle of incidence.

The differential equation solving method used was the fifth order Runge-Kutta method (34):

$$y_{i+1} = y_i + h(16K_1/135 + 6656K_3/12825 + 28561K_4/56430 - 9K_5/50 + 2K_6/55) \quad \text{(A-1a)}$$

$$K_1 = f(x_i, y_i) \quad \text{(A-1b)}$$

$$K_2 = f(x_i + h/4,\ y_i + h/4\ K_1) \quad \text{(A-1c)}$$

$$K_3 = f(x_i + 3h/8,\ y_i + h(3K_1/32 + 9K_2/32)) \quad \text{(A-1d)}$$

$$K_4 = f(x_i + 12h/13,\ y_i + h(1932K_1/2197 - 7200K_2/2197 + 7296K_3/2197)) \quad \text{(A-1e)}$$

$$K_5 = f(x_i + h,\ y_i + h(439K_1/216 - 8K_2 + 3680K_3/513 - 845K_4/4104)) \quad \text{(A-1f)}$$

$$K_6 = f(x_i + h/2,\ y_i + h(-8K_1/27 + 2K_2 - 3544K_3/2565 + 1859K_4/4104 - 11K_5/40)) \qquad \text{(A-1g)}$$

where y_{i+1} is the approximate solution at the end of a step, y_i is the approximate solution, or initial value, at the beginning of a step, h is the step length and f(x,y) is dy/dx, i.e., the system of first order differential equations to be solved.

The Runga-Kutta methods are derived by assuming the solution at the end of an interval is the solution at the interval beginning plus a weighted sum of derivative evaluations taken across the interval. Such is the form of (A-1a). The evaluation locations and weights are chosen by expanding the general form of (A-1a), with the weights and locations as arbitray constants, into a Taylor series and comparing terms to the Taylor expansion of a general two dimensional function. The resultant set of equations to be solved for the arbitrary constants is under specified, so there are always possible solutions. The order of the method is based on the number of terms in the Taylor expansion that are used. The order of the possible error made on each step of approximation is proportional to the first term in the Taylor expansion that is dropped. The above method is fifth order, so the local error is on the order of the sixth term in the Taylor expansion, which is proportional to h^6. The h used in the following

program was .Ø4 so the local error was on the order of $4 \times 1Ø^{-9}$ (35). When testing the results from the program for conservation of energy, the ratio of the total light energy input to the total light energy output, the error was never more than Ø.1%.

The inverse Fourier transforming was done by means of a Fast Fourier Transform (FFT) subprogram written by Jim Cooley, a graduate of the Virginia Tech Electrical Engineering Department. The only change made to his routine was to allow for double precision complex numbers. For more information on the FFT see references 36 and 37.

Squaring of the profile was done by squaring each of the 32 values returned from the FFT routine. The final results were printed to output data files. The arrays had to be rearranged in the printing since the FFT routine returns them with the center of the profile in the first position. The incrementing of the sound was done in a large loop containing all the above computer executed steps. The execution of the program was done on the IBM 3Ø9Ø mainframe computer available at Virginia Tech. The data files were downloaded to an IBM-PC for graphing.

The program to do the graphics appears in figure A.2, along with the directly and indirectly called library subroutines that do the actual three dimensional plotting. The subroutines were written by Ken Becker, a Master's

graduate of Virginia Tech Electrical Engineering and were obtained through the Engineering College PC Lab. The plots were done on an IBM-PC equipped with an 8087 math co-processor (a great convenience but not a necessity), a Color Graphics Adapter, an IBM-PC Color Display, and a graphics printer. The compiling and linking of the fortran and assembler routines was done using the Microsoft Fortran package, version 3.31, August, 1985.

```
C          This program solves a system of ordinary differential equa-
C     tions.  It is tailored specifically to the solution of the system
C     representing a bounded light beam encountering an ultra-sonic
C     sound beam.  The solving routine, subroutine RK, uses a 5th order
C     Runge-Kutta method to approximate the solution (see "Numerical
C     Analysis," L. Johnson and R. Riess, Addison-Wesley, 1982,p. 378).
C     The call for RK is nested in a loop so that it may be used for
C     many input angles to give an output angular spectrum of light.
C     These spectra are sent to a fast Fourier transform routine, FFT,
C     to produce an amplitude profile in rectangular coordinates.  The
C     light intensities are then found and printed.  Both the RK and FFT
C     calls are nested in a large loop that increments the amplitude of
C     the sound.  The initial conditions are placed in the array E in
C     subroutine RK.
C
C     VARIABLES:
C
C     MAIN PROGRAM:
C     A1,A3,A4,A5,A6,B21,B31,B32,B33,B41,B42,B43,B44,B52,B53,B54,B55,B61
C     B62,B63,B64,B65,B66: Constants used by the solving routine, they
C         also appear in RK.
C
C     N: The number of plane waves used in the angular spectrum of light
C        it must satisfy N = 2**P, where P is a positive integer.
C
C     M: The number of light orders in the problem, ie. th number of
C        equations in the system to be solved.
C
C     INV: a parameter for the FFT routine, 0 for the Fourier transform
C          1 for the inverse transform.
C
C     PI = 4.0 * ATAN(1.0), the computer generated version of pi.
C
C     DELALP: The increment value of alpha.
C
```

Figure A.1

```
C     ALPHA(I): A variable proprtional to the sound amplitude.
C
C     AL: A real number to hold the present value of ALPHA(I).
C
C     SUM: A real variable to hold the total intensity of all the light
C          waves in all the light orders at the output for a constant
C          value of ALPHA(I).  It is a check for energy conservation
C          that can be removed.
C
C     ANG: The light angle of incidence in radians.
C
C     BRAGG: The light angle of incidence normalized to the Bragg angle.
C
C     E1,E2,E3,E4: Complex arrays to hold the light amplitude spectra.
C
C     POWN2,POWN1,POW0,POW1,POW2: Real arrays to hold the intensity
C          spectra for each of the output orders.
C
C     SUBPROGRAM RK:
C     GAUSS: The Gaussian formula holding the shape of the input beam
C            of light.
C
C     E(I): A complex array holding the initial light amplitude
C           conditions and where the where the final values are passed
C           back to the main program.
C
C     H: The step length in the x-direction (across the sound beam)
C        used by the solving method.
C
C     X: The location in the sound beam of the beginning of a step.
C
C     XT: The intermittent locations within a step.
C
C     ET(I): A complex array for the temporary storage of light
C            amplitudes at the various values of XT.
```

Figure A.1 (cont')

```
C
C     K1,K2,K3,K4,K5,K6: Complex arrays for the values of dE/dx found
C           in subroutines EP1 and EP2, at the various XT values.
C
C     SUBROUTINE EP1 (and EP2 if included):
C     EJ: j, SQRT(-1)
C
C     Q: A parameter in the system to be solved.
C
C     HAM: The local value of the sound amplitude (optional)
C
C     ECAL: -j*ALPHA(I)/2
C
C     ECEX: j*Q*XT/2
C
C     EPR(I): A complex array holding dE/dx.
C
C     ET(I): A complex array holding the current value of E(I).
C
C
C
C     Set variable types and the constants for the differential equation
C         solver.
C
C
      IMPLICIT COMPLEX*16 (E)
      DIMENSION E(5),ALPHA(53),POWN2(32),POWN1(32),POW0(32)
      DIMENSION POW1(32),POW2(32)
      DIMENSION E1(32),E2(32),E3(32),E4(32),E5(32)
      COMMON /A/A1,A3,A4,A5,A6,B21,B31,B32,B33,B41,B42,B43,B44,B52,B53,
     $B54,B55,B61,B62,B63,B64,B65,B66
      A1=16./135.
      A3=6656./12825.
      A4=28561./56430.
      A5=-9./50.
```

Figure A.1 (cont')

```
      A6=2./55.
      B21=1./4.
      B31=3./8.
      B32=3./32.
      B33=9./32.
      B41=12./13.
      B42=1932./2197.
      B43=-7200./2197.
      B44=7296./2197.
      B52=439./216.
      B53=-8.
      B54=3680./513.
      B55=-845./4104.
      B61=1./2.
      B62=-8./27.
      B63=2.
      B64=-3544./2565.
      B65=1859./4104.
      B66=-11./40.
C
C     Input the number of waves in the specta and the number of light
C      orders to be used in the problem.  Set INV for inverse transform,
C      establish pi and set the sound increment.
C
      N= 32
      M= 4
      INV=1
      PI = 4.0*ATAN(1.0)
      DELALP= .0625*PI
C
C     The primary loop, which increments the sound pressure, begins.
C
      DO 100 I=1,65
        ALPHA(I)= (I-1)*DELALP
        AL= ALPHA(I)
```

Figure A.1 (cont')

```
C       SUM= 0.
C
C     Secondary loop, which increments the input angle and calls the
C       differential equation solver, begins.
C
          DO 90 J1=1,N
          ANG= (J1-1)*.1500E-3 -2.325E-3
          BRAGG= (ANG -3.25E-3)/3.25E-3
          CALL RK(E,AL,ANG,BRAGG,M)
C
C     Amplitude values for each light wave transfered to an array to
C       hold the entire spectrum for an order.
C
            E1(J1)= E(1)
            E2(J1)= E(2)
            E3(J1)= E(3)
            E4(J1)= E(4)
C
C     Light intensities summed for consevation of energy testing.
C
C         POWN1(J1)=(CDABS(E(1)))**2
C         POW0(J1) =(CDABS(E(2)))**2
C         POW1(J1) =(CDABS(E(3)))**2
C         POW2(J1) =(CDABS(E(4)))**2
C         SUM= SUM+POWN1(J1)+POW0(J1)+POW1(J1)
C    $        +POW2(J1)
   90     CONTINUE
C
C     Inverse Fourier transforms found.
C
          CALL FFT(E1,N,INV)
          CALL FFT(E2,N,INV)
          CALL FFT(E3,N,INV)
          CALL FFT(E4,N,INV)
C
```

Figure A.1 (cont')

```
C     Light intensity profiles found.
C
          DO 95 J1=1,N
          POWN1(J1)=(CDABS(E1(J1)))**2
          POW0(J1) =(CDABS(E2(J1)))**2
          POW1(J1) =(CDABS(E3(J1)))**2
   95     POW2(J1) =(CDABS(E4(J1)))**2
C
C     Output desired order(s).
C
        WRITE(6,20) (POW1(J1),J1=25,32),(POW1(J1),J1=1,24)
   20   FORMAT(1X,8F10.7)
  100 CONTINUE
      STOP
      END
C
C
C
C
C     Subroutine to solve a system of ordinary differential equations.
C
      SUBROUTINE RK(E,AL,ANG,BRAGG,M)
      IMPLICIT COMPLEX*16 (E,K)
      DOUBLE PRECISION GAUSS
      DIMENSION E(5),K1(5),K2(5),K3(5),K4(5),K5(5),K6(5),ET(5)
      COMMON /A/A1,A3,A4,A5,A6,B21,B31,B32,B33,B41,B42,B43,B44,B52,B53,
     $B54,B55,B61,B62,B63,B64,B65,B66
C
C     Initial conditions for the system.
C
      GAUSS= DEXP(-6.161788865D6*ANG**2)
      E(1)=DCMPLX(0D0,0D0)
      E(2)=DCMPLX(GAUSS,0D0)
      E(3)=DCMPLX(0D0,0D0)
      E(4)=DCMPLX(0D0,0D0)
```

Figure A.1 (cont')

```
C
C     Set the step size for the solver and loop through the steps
C       necessary to cover the sound column.  The loop shown as comments
C       is for a second sound column. If it is used and additional
C       subroutine, EP2, must be added containing the equations for the
C       system as it is represented in that column.  EP2 is the same as
C       EP1 with different equations for EPR(I).
C
      H= .040
      DO 10 J=1,26
        X= (J-1)*H
        XT= X
        DO 1 L=1,M
    1     ET(L)= E(L)
        CALL EP1(XT,ET,K1,AL,BRAGG)
        XT= X+ H*B21
        DO 2 L=1,M
    2     ET(L)= E(L)+ H*B21*K1(L)
        CALL EP1(XT,ET,K2,AL,BRAGG)
        XT= X+ H*B31
        DO 3 L=1,M
    3     ET(L)= E(L)+ H*(B32*K1(L)+B33*K2(L))
        CALL EP1(XT,ET,K3,AL,BRAGG)
        XT= X+ H*B41
        DO 4 L=1,M
    4     ET(L)= E(L)+ H*(B42*K1(L)+B43*K2(L)+B44*K3(L))
        CALL EP1(XT,ET,K4,AL,BRAGG)
        XT= X+ H
        DO 5 L=1,M
    5     ET(L)= E(L)+ H*(B52*K1(L)+B53*K2(L)+B54*K3(L)+B55*K4(L))
        CALL EP1(XT,ET,K5,AL,BRAGG)
        XT= X+ H*B61
        DO 6 L=1,M
    6     ET(L)= E(L)+ H*(B62*K1(L)+B63*K2(L)+B64*K3(L)+B65*K4(L)+
     $                    B66*K5(L))
```

Figure A.1 (cont')

```
      CALL EP1(XT,ET,K6,AL,BRAGG)
      DO 7 L=1,M
    7   E(L)= E(L)+H*(A1*K1(L)+A3*K3(L)+A4*K4(L)+A5*K5(L)+A6*K6(L))
   10 CONTINUE
C
C     DO 20 J=22,41
C       X= (J-1)*H
C       XT= X
C       DO 11 L=1,M
C   11    ET(L)= E(L)
C       CALL EP2(XT,ET,K1,AL,BRAGG)
C       XT= X+ H*B21
C       DO 12 L=1,M
C   12    ET(L)= E(L)+ H*B21*K1(L)
C       CALL EP2(XT,ET,K2,AL,BRAGG)
C       XT= X+ H*B31
C       DO 13 L=1,M
C   13    ET(L)= E(L)+ H*(B32*K1(L)+B33*K2(L))
C       CALL EP2(XT,ET,K3,AL,BRAGG)
C       XT= X+ H*B41
C       DO 14 L=1,M
C   14    ET(L)= E(L)+ H*(B42*K1(L)+B43*K2(L)+B44*K3(L))
C       CALL EP2(XT,ET,K4,AL,BRAGG)
C       XT= X+ H
C       DO 15 L=1,M
C   15    ET(L)= E(L)+ H*(B52*K1(L)+B53*K2(L)+B54*K3(L)+B55*K4(L))
C       CALL EP2(XT,ET,K5,AL,BRAGG)
C       XT= X+ H*B61
C       DO 16 L=1,M
C   16    ET(L)= E(L)+ H*(B62*K1(L)+B63*K2(L)+B64*K3(L)+B65*K4(L)+
C     $                     B66*K5(L))
C       CALL EP2(XT,ET,K6,AL,BRAGG)
C       DO 17 L=1,M
C   17    E(L)= E(L)+H*(A1*K1(L)+A3*K3(L)+A4*K4(L)+A5*K5(L)+A6*K6(L))
C   20 CONTINUE
```

Figure A.1 (cont')

```
      RETURN
      END
C
C
C
C
C     Subroutine holding the system of equations to be solved by RK.
C
      SUBROUTINE EP1(XT,ET,EPR,AL,BRAGG)
      IMPLICIT COMPLEX*16 (E)
      DIMENSION ET(5),EPR(5)
      EJ=DCMPLX(0.D0,1.D0)
      PI=4.0*ATAN(1.0)
C
C     A system representing four light orders with a Hamming sound
C       amplitude distribution.  The Hamming distribution can be removed
C       by putting a "C" in column 1 in front of HAM and removing "*HAM"
C       from ECAL.
C
      Q= 187.
      HAM= .54 + .46*SIN(PI*XT)
      ECAL= -EJ*AL*.5*HAM
      ECEX= EJ*.5*Q*XT
      EPR(1)= ECAL*(CDEXP(ECEX*(BRAGG-1.))
     $        *ET(2))
      EPR(2)= ECAL*(CDEXP(-ECEX*(BRAGG-1.))*ET(1)+CDEXP(ECEX*(BRAGG+1.))
     $        *ET(3))
      EPR(3)= ECAL*(CDEXP(-ECEX*(BRAGG+1.))*ET(2)+CDEXP(ECEX*(BRAGG+3.))
     $        *ET(4))
      EPR(4)= ECAL*(CDEXP(-ECEX*(BRAGG+3.))*ET(3))
      RETURN
      END
C
C
```

Figure A.1 (cont')

```
      SUBROUTINE FFT(X,N,INV)
C
C--------------------------------------------------------------------
C SUBROUTINE: FFT
C JIM COOLEY'S SIMPLE FFT PROGRAM USING DECIMATION IN TIME ALGORITHM
C--------------------------------------------------------------------
C
C   X = 2**M COMPLEX ARRAY THAT INITIALLY CONTAINS INPUT
C       AND ON RETURN CONTAINS TRANSFORM.
C   N = 2**M POINT
C INV = 0, DIRECT TRANSFORM
C INV = 1, INVERSE TRANSFORM
C
      COMPLEX*16 X(N),U,W,T,CMPLX
      M = ALOG(FLOAT(N))/ALOG(2.) + .1
      NV2 = N/2
      NM1 = N - 1
      J = 1
      DO 40 I=1,NM1
        IF (I.GE.J) GO TO 10
        T = X(J)
        X(J) = X(I)
        X(I) = T
   10   K = NV2
   20   IF (K.GE.J) GO TO 30
        J = J - K
        K = K/2
        GO TO 20
   30   J = J + K
   40 CONTINUE
      PI = 4.0*ATAN(1.0)
      DO 70 L=1,M
        LE = 2**L
        LE1 = LE/2
        U = (1.0,0.0)
```

Figure A.1 (cont')

```
      W = CMPLX(COS(PI/FLOAT(LE1)),-SIN(PI/FLOAT(LE1)))
      IF (INV.NE.0) W = CONJG(W)
      DO 60 J=1,LE1
        DO 50 I=J,N,LE
          IP = I + LE1
          T = X(IP)*U
          X(IP) = X(I) - T
          X(I) = X(I) + T
50      CONTINUE
        U = U*W
60    CONTINUE
70  CONTINUE
    IF (INV.EQ.0) RETURN
    DO 80 I=1,N
      X(I) = X(I)/CMPLX(FLOAT(N),0.)
80  CONTINUE
    RETURN
    END
```


Figure A.1 (cont')

```
C
C          This program will accept data in the form of decimal numbers
C     in array form (maximum size 32x100) and produce three dimensional
C     plot on a graphics display linked to an IBM PC.  It will then
C     print the plot to a graphics printer, if attached, and after the
C     GRAPHICS.COM program in the PC DOS has been executed.  It reads in
C     up to 100 vectors (set when the number of plots is asked for) of
C     up to 32 elements (set when the number of points per plot is asked
C     for) from a data file specified interactively.  The first point is
C     plotted to the left rear of the graph.  Subroutines written by Ken
C     Becker.
C
      PROGRAM TDP32
      DIMENSION DAT(32,100),WK1(32,100),WK2(32,100)
      CHARACTER*32 DATF
C
C     Input: data file, number of vectors (max. 100), number of elements
C            per vector (max. 32).
C
      WRITE(*,'(A\)')' INPUT DATA FILE NAME '
      READ(*,'(A)') DATF
      OPEN(7,FILE=DATF)
      WRITE(*,'(A\)') ' INPUT NUMBER OF PLOTS '
      READ(*,*) M
      WRITE(*,'(A\)') ' INPUT NUMBER OF POINTS/PLOT '
      READ(*,*) N
C
C     Read in the data.
C
      DO 10 J=1,M
         READ(7,*) (DAT(I,J),I=1,N)
   10    CONTINUE
C
C     Set the size and orientation of the plot for a broadside view.
C
```

Figure A.2

```
      ID= 32
      XLEN= 4.0
      YLEN= 2.0
      ZLEN= 13.8
      PHI=  180.
      THETA=-25.0
      XREF= 7.0
      YREF= 6.0
C
C     Plot a broadside view.  Wait for a keystroke to print.
C
C     CALL PLOT3D(DAT,ID,N,M,XLEN,YLEN,ZLEN,PHI,THETA,XREF,YREF,WK1,WK2)
C     READ(*,'(A)') THING
C     CALL PRSCQQ
C
C     Set plot size and orientation for a 3/4 view.
C
      XLEN= 6.0
      ZLEN= 7.5
      PHI=  135.
      THETA=-40.0
      XREF= 4.5
      YREF= 6.5
C
C     Plot a 3/4 view and wait for a keystroke to print.
C
      CALL PLOT3D(DAT,ID,N,M,XLEN,YLEN,ZLEN,PHI,THETA,XREF,YREF,WK1,WK2)
      READ(*,'(A)') THING
      CALL PRSCQQ
C
C     Reset console to normal text mode.
C
      CALL SCRNQQ(3)
  999 STOP
      END
```

Figure A.2 (cont')

```
$storage:2
      SUBROUTINE PLOT3D(DAT,IA,N,M,XLEN,YLEN,ZLEN,PHI,THETA,XREF,YREF,
     *                  WK1,WK2)
C
C THIS SUBROUTINE GENERATES A CROSSHATCHED 3D PLOT.
C
C   DAT(IA,M)-  REAL ARRAY, DIMENSIONED N BY M, WITH DATA TO BE
C               PLOTTED.
C
C   IA        -  ROW DIMENSION OF DAT ARRAY, EXACTLY AS DIMENSIONED
C               IN THE CALLING PROGRAM.
C   N         -  INTEGER 'X' DIMENSION OF THE ARRAY.
C   M         -  INTEGER 'Y' DIMENSION OF THE ARRAY.
C   XLEN      -  REAL VALUE DENOTING THE LENGTH OF THE X AXIS IN
C               INCHES.
C   YLEN      -  REAL VALUE, DENOTING THE LENGTH OF THE Y AXIS (THE AXIS
C               THAT IS DISPLAYED VERTICALLY ON THE SCREEN) IN INCHES.
C   ZLEN      -  REAL VALUE, DENOTING THE LENGTH OF THE Z AXIS (THE AXIS
C               THAT APPEARS TO 'COME OUT' OF THE SCREEN) IN INCHES.
C   PHI       -  REAL VALUE, IN DEGREES, THAT DENOTES THE AMOUNT OF
C               CLOCKWISE ROTATION ABOUT THE Y AXIS.
C   THETA     -  REAL VALUE, IN DEGREES, THAT DENOTES THE AMOUNT OF
C               CLOCKWISE ROTATION ABOUT THE X AXIS.
C   XREF, YREF- REAL VALUES, IN INCHES, DENOTING THE POSITION WHERE
C               DAT(1,1) WILL BE PLOTTED ON THE SCREEN.  NOTE THAT THE
C               SCREEN IS 10 INCHES WIDE (IN X) AND 8 INCHES HIGH (IN
C               (Y).  AVOID PLACING PORTIONS OF THE PLOT OUTSIDE THE
C               VIEWING AREA.  0 <= XREF <= 10 AND 0 <= Y <= YREF.
C   WK1        - REAL ARRAY USED AS A WORKING VECTOR.  IT MUST BE
C               DIMENSIONED MAX(N,M).
C   WK2        - REAL ARRAY USED AS A WORKING VECTOR.  IT MUST BE
C               DIMENSIONED MAX(N,M).
C
C   FINAL NOTE: WITH PHI=THETA=0, DAT(1,1) WILL BE PLOTTED IN THE
C     LEFT FOREGROUND AND DAT(N,1) WILL BE PLOTTED IN THE RIGHT
```

Figure A.2 (cont')

```
C     FOREGROUND.
C
C REFERENCE:  WATKINS, S.L., "ALGORITHM 483, MASKED THREE-DIMENSIONAL
C             PLOT PROGRAM WITH ROTATIONS," COLLECTED ALGORITHMS FROM
C             CACM, 26 MARCH 1973, PP. 483-P 1-0 TO 5-0.
C
      INTEGER IA,N,M,ORIGX,ORIGY,I,J,K,NLINE,MASK(2000),OLDX,OLDY,VER
      REAL DAT(IA,M),XLEN,YLEN,WK1(*),WK2(*),YMAX,YMIN,XSCALE,
     *     YSCALE,ZSCALE,ZLEN,XLENTH,YLENTH,XREF,YREF,PHI,THETA,
     *     VERTEX(16),TEMP,PI
      LOGICAL FORWRD
C
      COMMON /PL3QQE/ORIGX,ORIGY,OLDX,OLDY
      DATA XLENTH/10.0/
      PI = 4.*ATAN(1.)
      ORIGX = 0
      ORIGY = 0
      OLDX = 0
      OLDY = 0
      CALL SCRNQQ(6)
      XSCALE = XLEN/REAL(N-1)
      ZSCALE = ZLEN/REAL(M-1)
      YMIN = DAT(1,1)
      YMAX = DAT(1,1)
      DO 10 I = 1,N
        DO 20 J = 1,M
          IF (DAT(I,J).GT.YMAX) YMAX = DAT(I,J)
          IF (DAT(I,J).LT.YMIN) YMIN = DAT(I,J)
   20   CONTINUE
   10 CONTINUE
      YSCALE = YLEN/(YMAX-YMIN)
C
C GENERATE FIRST FIGURE
C
C
```

Figure A.2 (cont')

```
C GET VALUES OF PHI AND THETA ON -180 < PHI,THETA < 180
C
      IF (SIN(PHI*PI/180.).GE.0.0) THEN
        PHI = (180./PI)*ACOS(COS(PHI*PI/180.))
       ELSE
        PHI = -(180./PI)*ACOS(COS(PHI*PI/180.))
      ENDIF
      IF (SIN(THETA*PI/180.).GE.0.0) THEN
        THETA = (180./PI)*ACOS(COS(THETA*PI/180.))
       ELSE
        THETA = -(180./PI)*ACOS(COS(THETA*PI/180.))
      ENDIF
C
C DETERMINE WHICH DIRECTION Z IS GOING IN THIS FIGURE
C
      IF (ABS(PHI).LE.90.0) THEN
        FORWRD = .TRUE.
       ELSE
        FORWRD = .FALSE.
      ENDIF
      IF (ABS(THETA).GT.90.0) FORWRD = (.NOT.FORWRD)
      IF (FORWRD) THEN
        DO 30 NLINE = 1,M
          DO 40 I = 1, N
            WK1(I) = DAT(I,NLINE)-YMIN
40        CONTINUE
          CALL PL3QQA(10,0.0,WK1,0.0,XSCALE,YSCALE,-ZSCALE,
     *      NLINE, N, PHI, THETA, XREF, YREF, XLENTH, MASK, 0.0)
30      CONTINUE
       ELSE
        DO 35 NLINE = 1,M
          DO 45 I = 1, N
            WK1(I) = DAT(I,(M-NLINE+1))-YMIN
45        CONTINUE
          CALL PL3QQA(10,0.0,WK1,-ZLEN,XSCALE,YSCALE,ZSCALE,
```

Figure A.2 (cont')

```
     *      NLINE, N, PHI, THETA, XREF, YREF, XLENTH, MASK, 0.0)
   35   CONTINUE
        ENDIF
C
C GENERATE SECOND FIGURE
C
        IF (PHI.LE.0.0) THEN
          FORWRD = .TRUE.
         ELSE
          FORWRD = .FALSE.
        ENDIF
        IF (ABS(THETA).GT.90.0) FORWRD = (.NOT.FORWRD)
        IF (FORWRD) THEN
          DO 50 J = 1,M
            WK2(J) = -ZSCALE*(J-1)
   50   CONTINUE
        DO 60 NLINE = 1, N
          X = XSCALE*(NLINE-1)
          DO 70 I = 1, M
            WK1(I) = DAT(NLINE,I)-YMIN
   70     CONTINUE
          CALL PL3QQA(1011,X,WK1,WK2,0.0,YSCALE,1.0,NLINE,M,PHI,THETA,
     *      XREF,YREF,XLENTH,MASK,VERTEX)
   60   CONTINUE
       ELSE
        DO 55 J = 1,M
          WK2(J) = -ZSCALE*(J-1)
   55   CONTINUE
        DO 65 NLINE = 1, N
          X =  XLEN - XSCALE*(NLINE-1)
          DO 75 I = 1, M
            WK1(I) = DAT((N-NLINE+1),I)-YMIN
   75     CONTINUE
          CALL PL3QQA(1011,X,WK1,WK2,0.0,YSCALE,1.0,NLINE,M,PHI,THETA,
     *      XREF,YREF,XLENTH,MASK,VERTEX)
```

Figure A.2 (cont')

```
   65   CONTINUE
        ENDIF
C
C DRAW A FRAME
C
        IF (ABS(THETA).LT.90) THEN
          VER = 3
         ELSE
          VER = 4
        ENDIF
        CALL PL3QQB(VER,VERTEX,MASK)
        RETURN
        END
C
C
        SUBROUTINE PL3QQA(IVXYZ, XDATA, YDATA, ZDATA, XSCALE,
       * YSCALE, ZSCALE, NLINE, NPNTS, PHI, THETA, XREF,
       * YREF, XLENTH, MASK, VERTEX)
C MASKED 3-DIMENSIONAL PLOT PROGRAM WITH ROTATIONS
C THIS ROUTINE WILL ACCEPT 3-DIMENSIONAL DATA IN VARIOUS
C FORMS AS INPUT, ROTATE IT IN 3-SPACE TO ANY ANGLE,
C AND PLOT THE PROJECTION OF THE RESULTING FIGURE ONTO THE
C XY PLANE.  LINEAR INTERPOLATION IS USED BETWEEN DATA
C POINTS.  THOSE LINES OF A FIGURE WHICH SHOULD BE HIDDEN BY
C A PREVIOUS LINE ARE MASKED.
C THE MASKING TECHNIQUE USED BY THIS ROUTINE IS BASED ON
C TWO PREMISES =
C         LINES IN THE FOREGROUND (POSITIVE Z DIRECTION)
C         ARE PLOTTED BEFORE LINES IN THE BACKGROUND.
C         A LINE OR PORTION OF A LINE IS MASED (HIDDEN) IF
C         IT LIES WITHIN THE REGION BOUNDED BY PREVIOUSLY
C         PLOTTED LINES.
C EACH CALL TO PL3QQA CAUSES ONE LINE OF A FIGURE TO BE
C PLOTTED.
C TWO PARAMETERS OF THE PLOTTER ARE SET ON THE INITIAL CALL
```

Figure A.2 (cont')

```
C FOR EACH FIGURE -
C (PIPI) IS THE NUMBER OF PLOTTER INCREMENTS PER INCH
C (NYPI) IS THE NUMBER OF INCREMENTS AVAILABLE ACROSS THE
C WIDTH OF THE PAPER (Y-DIRECTION).
C WHEN A NEW FIGURE IS INITIATED, THE PLOTTER ORIGIN IS SET
C AT THE BOTTOM OF THE PAPER BY PL3QQA AND SHOULD NOT BE
C MOVED UNTIL THE FIGURE IS COMPLETED.
C INPUT PARAMETERS -
C (IVXYZ) IS A FOUR DIGIT DECIMAL INTEGER WHICH IS USED TO
C SELECT VARIOUS INPUT/OUTPUT OPTIONS.  THESE DIGITS, IN
C DECREASING ORDER OF MAGNITUDE, WILL BE REFERRED TO AS V,
C XY, Y, AND Z.
C IF V .NE. 0, THE VERTICES OF THE CURRENT FIGURE AND THEIR
C PROJECTION ONTO THE Y=0 PLANE, WILL BE STORED IN A 16
C ENTRY REAL ARRAY (VERTEX) AND WILL BE UPDATE4D AS EACH
C LINE IS PLOTTED.  THESE COORDINATES ARE IN INCHES AND
C RELATIVE TO THE CURRENT PLOTTER ORIGIN.  THE X Y PAIRS
C ARE ORDERED SO THAT THE FIRST PAIR CORRESPONDS
C TO THE LAST POINT OF THE FIRST LINE,A ND THE FOLLOWING
C PAIRS ARE ORDERED IN A CIRCULAR FASHION.  THE PAIRS ON THE
C Y=0 PLANE OF THE FIGURE, THEN FOLLOW IN THE SAME ORDER.
C IF V=0, THE VERTEX PARAMETER IS IGNORED, BUT SHOULD NOT BE DELETED
C IF X=0, THE X-COMPONENTS OF THIS LIEN ARE ASSUMED TO BE
C EQUALLY SPACED, AND ARE COMPUTED BY
C                X(I) = XDATA+(I-1)*XSCALE
C WHERE (XDATA) IS THE INITIAL VALUE IN INCHES AND (XSCALE)
C IS THE SPACING BETWEEN POINTS IN INCHES.  IF X .NE. 0, THE
C X-COMPONENTS OF THIS LINE ARE READ FROM AN ARRAY AND
C MODIFIED BY
C                X(I) = XDATA(I)*XSCALE
C WHERE (XSCALE) IS A SCALE FACTOR.
C THE SAME RELATIONS HOLD FOR THE Y-COMPONENTS, THAT IS, IF
C Y=0
C                Y(I) = YDATA+(I-1)*YSCALE
C AND IF Y .NE. 0
```

Figure A.2 (cont')

```
C                Y(I) = YDATA(I)*YSCALE
C IF Z=0, THE Z-COMPONENTS OF THIS LINE ARE ALL ASSUMED TO
C BE EQUAL, AND ARE COMPUTED BY
C                Z(I)=ZDATA+(NLINE-1)*ZSCALE
C WHERE (NLINE) IS SOME INTEGER ASSOCIATED WITH THIS LINE.
C IF Z .NE. 0, AGAIN WE HAVE
C                Z(I) = ZDATA(I)*ZSCALE
C WHEN (NLINE) IS EQUAL TO ONE, IT INDICATES THE BEGINNING
C OF A NEW FIGURE.  A CALL TO PL3QQA WITH (NLINE) EQUAL TO
C ZERO BEFORE INITIATING A NEW FIGURE SIMULATES A LINE DRAWN
C AT THE BOTTOM OF THE PAGE.  THEREFORE ONLY THOSE PORTIONS
C OF A LINE LYING ABOVE ALL PREVIOUS LINES WILL BE PLOTTED.
C ALL OTHER PARAMETERS ARE IGNORED ON SUCH A CALL.
C (NPNTS) IS THE NUMBER OF POINTS ON THIS LINE, AND MAY BE
C ALTERED FROM LINE TO LINE.
C (PHI) AND (THETA) ARE THE TWO ANGLES (IN DEGREES) USED TO
C SPECIFY THE DESIRED 3-DIMENSIONAL REOTATION.  THE FOLLOWING
C TWO DEFINITIONS OF THESE REOTATIONS ARE EQUIVALENT -
C IN TERMS OF ROTATIONS OF AXES, THE INITIAL SYSTEM OF AXES,
C XYZ, IS ROTATED BY AN ANGLE (PHI) COUNTERCLOCKWISE ABOUT
C THE Y-AXIS, AND THE RESULTANT SYSTEM IS LABELED THE TUV
C AXES.  THE TUV AXES ARE THEN ROTATED BY AN ANGLE (THETA)
C COUNTERCLOCKWISE ABOUT THE T-AXIS, AND THIS FINAL SYSTEM
C IS LABELED THE PQR AXIS.  THE PLOTTED FIGURE IS THE
C PROJECTION OF THE ORIGINAL FIGURE ONTO THE PQ-PLANE.
C IN TERMS OF ROTATIONS OF COORDINATES, THE FIGURE IS FIRST
C ROTATED BY AN ANGLE (THETA) CLOCKWISE ABOUT THE X-AXIS.
C THE RESULTANT FIGURE IS THEN ROTATED BY AN ANGLE (PHI)
C CLOCKWISE ABOUT ITS Y-AXIS.  THE PLOTTED FIGRE IS THE
C PROJECTION OF THIS FINAL FIGURE ONTO THE XY-PLANE.
C WARNING.  SOME ROTATIONS WILL ALTER THE FOREGROUND/
C BACKGROUND RELATIONSHIPS BETWEEN THE LINES, AND
C THUS THE ORDER IN WHICH THEY SHOULD BE PLOTTED.
C (XREF) AND (YREF) ARE THE COORDINATES, IN INCHES,
C RELATIVE TO THE PLOTTER ORIGIN, TO BE USED AS THE ORIGIN
```

Figure A.2 (cont')

```
C OF THE FIGURE.
C (XLENTH) IS THE LENGTH, IN INCHES, TO WHICH THE PLOT IS
C RESTRICTED.  ANY POINT WHICH EXCEEDS THIS LIMIT, OR THE
C LIMITS OF THE PAPER IN THE Y DIRECTION (NYPI) WILL BE
C SET TO THAT LIMIT.
C (MASK) IS AN INTEGER ARRAY OF 2*XLENTH*PIPI ENTRIES WHICH
C IS USED TO STORE THE MASK.  THE CONTENTS OF THIS ARRAY
C SHOULD NOT BE ALTERED DURING THE PLOTTING OF ANY GIVEN
C FIGURE.
C ALL PARAMETERS EXCEPT (MASK) AND (VERTEX) ARE RETURNED
C UNCHANGED.
C BETWEEN ANY TWO CALLS FOR THE SAME FIGURE, ANY PARAMETER
C CAN BE MEANINGFULLY CHANGED EXCEPT (XLENTH), (MASK), AND
C (VERTEX).
      INTEGER HIGH, OLDHI, OLDLOW
      DIMENSION XDATA(*), YDATA(*), ZDATA(*), MASK(*),
     * VERTEX(*)
      DATA INIT, JVXYZ, SPHI, STHETA/-1, -1, -1.0E29,
     * -1.0E29/
C INITIALIZATION PROCEDURES
C INITIALIZATION PROCEDURE FOR A NEW FIGURE
C TEST FOR SPECIAL MASK MODIFYING CALL
      IF (NLINE.EQ.0) GO TO 550
C DETERMINE IF INITIALIZATION IS REQUIRED
      IF (NLINE.NE.1) GO TO 20
C SET PLOTTER PARAMETERS
      PIPI = 64.0
      NYPI = 512
C RESET PLOTTER ORIGIN TO BOTTOM OF PLOT PAGE
      I = 0
      CALL PL3QQC(0,-I, -3)
C COMPUTE LENGTH OF PLOT PAGE IN INCREMENTS
      LIMITX = XLENTH*PIPI + 0.5
      I = LIMITX + LIMITX
C INITIALIZE MASKING ARRAY OVER THE LENGTH OF THE PLOT PAGE
```

Figure A.2 (cont')

```
      DO 10 K = 1,I
        MASK(K) = INIT
   10 CONTINUE
      INIT = -1
C SET THE NECESSARY INDICATORS FOR THE FIRST LINE OF A ENW
C FIGURE
      INCI = -1
      I = 0
C INPUT TYPE AND VERTEX INITIALIZATION
C DETERMINE IF INITIALIZATION IS REQUIRED
   20 IF (JVXYZ.EQ.IVXYZ) GO TO 70
C SET INDICATORS FOR TYPES OF INPUT DATA AND SAVING VERTICES
      JVXYZ = IVXYZ
      INDZ = 1
      INDY = 1
      INDX = 1
      INDV = 1
      IF (JVXYZ.LT.1000) GO TO 30
      INDV = 2
      JVXYZ = JVXYZ - 1000
   30 IF (JVXYZ.LT.100) GO TO 40
      INDX = 2
      JXYZ = JVXYZ - 100
   40 IF (JVXYZ.LT.10) GO TO 50
      INDY = 2
      JVXYZ = JVXYZ - 10
   50 IF (JVXYZ.LT.1) GQ TO 60
      INDZ = 2
   60 JVXYZ = IVXYZ
C ROTATION INITIALIZATION
C DETERMINE IF INITIALIZATION IS REQUIRED
   70 IF (PHI.EQ.SPHI .AND. THETA.EQ.STHETA) GO TO 80
C COMPUTE ROTATION FACTORS
      SPHI = SIN(0.0174532925*PHI)
      CPHI = COS(0.0174532925*PHI)
```

Figure A.2 (cont')

```
      STHETA = SIN(0.0174532925*THETA)
      CTHETA = COS(0.0174532925*THETA)
      A11 = CPHI
      A13 = -SPHI
      A21 = STHETA*SPHI
      A22 = CTHETA
      A23 = STHETA*CPHI
      SPHI = PHI
      STHETA = THETA
C PROCESSING PROCEDURES
C SET FLAG TO MOVE THROUGH THE DATA ARRAYS IN THE OPPOSITE
C DIRECTION
   80 INCI = -INCI
C SET INDICATOR TO THE FIRST POINT TO BE PROCESSED
      IF (I.NE.0) I = NPNTS + 1
C LOOP TO PROCESS EACH POINT IN THE DATA ARRAYS
      DO 530 K=1,NPNTS
C DATA CALCULATION
        I = I + INCI
        GO TO (90,100), INDX
   90   X = XDATA(1) + (I-1)*XSCALE
        GO TO 110
  100   X = XDATA(I)*XSCALE
  110   GO TO (120,130),INDY
  120   Y = YDATA(1) + (I-1)*YSCALE
        GO TO 140
  130   Y = YDATA(I)*YSCALE
  140   GO TO (150,160), INDZ
  150   Z = ZDATA(1) + (NLINE-1)*ZSCALE
        GO TO 170
  160   Z = ZDATA(I)*ZSCALE
C DATA ROTATION
  170   XXX = A11*X + A13*Z + XREF
        XX = XXX
        IX = NINT(XX*PIPI)
```

Figure A.2 (cont')

```
        YYY = A21*X + A23*Z + YREF
        YY = YYY + A22*Y
        IY = NINT(YY*PIPI)
C RESTRICT FIGURE TO PLOT PAGE
        IF (IX.LE.0) IX = 1
        IF (IX.GT.LIMITX) IX = LIMITX
        IF (IY.LT.10) IY = 10
        IF (IY.GT.NYPI) IY = NYPI
        IF (K.NE.1) GO TO 250
C (LOC) IS THE POSITION OF THE PREVIOUS POINT WITH RESPECT
C TO THE MASK
C        +1 ABOVE THE MASK
C         0 WITHIN THE LIMITS OF THE MASK
C        -1 BELOW THE MASK
C PROCEDURE FOR INITIAL POINT OF EACH LINE
C LOCATE INITIAL POINT WITH RESPECT TO THE MASK THEN
C UPDATE THE MASK
        LOW = IX + IX
        HIGH = LOW - 1
        MLOW = MASK(LOW)
        MHIGH = MASK(HIGH)
        IF (MHIGH-IY) 200, 210, 180
  180   IF (MLOW-IY) 190, 230, 220
  190   LOCOLD = 0
        GO TO 240
  200   MASK(HIGH) = IY
        IF (MLOW.EQ.-1) MASK(LOW) = IY
  210   LOCOLD = +1
        GO TO 240
  220   MASK(LOW) = IY
  230   LOCOLD = -1
C MOVE THE RAISED PEN TO THIS INITIAL POINT
  240   CALL PL3QQC(IX, IY, 3)
        JX = IX
        JY = IY
```

Figure A.2 (cont')

```
        IYREF = IY
C STORE VERTICES IF REQUESTED
        IF (INDV.EQ.1) GO TO 530
        INDEX = INCI + 6
        VERTEX(INDEX) = XX
        VERTEX(INDEX+1) = YY
        VERTEX(INDEX+8) = XXX
        VERTEX(INDEX+9) = YYY
        IF (NLINE.NE.1) GO TO 530
        VERTEX(1) = XX
        VERTEX(2) = YY
        VERTEX(9) = XXX
        VERTEX(10) = YYY
        GO TO 530
C SPECIAL CASE WHERE CHANGE IN X COORDINATE IS ZERO
C A SPECIAL PROVISION IS MADE AT THIS POINT SO THAT A LINE
C WILL NOT MASK ITSELF AS LONG AS THE X COORDINATE RAMAINS
C CONSTANT
  250   IF (IX.NE.JX) GO TO 260
        JY = IY
        GO TO 280
C COMPUTE CONSTANTS FOR LINEAR INTERPOLTATION
  260   YINC = FLOAT(IY-JY)/ABS(FLOAT(IX-JX))
        INCX = (IX-JX)/IABS(IX-JX)
        YJ = JY
C PERFORM LINEAR INTERPOLATION AT EACH INCREMENTAL STEP ON
C THE X AXIS
  270   JX = JX + INCX
        YJ = YJ + YINC
        JY = NINT(YJ)
C LOCATE THE CURRENT POINT WITH RESPECT TO THE MASK AT THAT
C POINT THEN PLOT THE INCREMENT AS A FUNCTION OF THE
C LOCATION OF THE PREVIOUS POINT WITH RESPECT TO ITS MASK
        LOW = JX + JX
        HIGH = LOW - 1
```

Figure A.2 (cont')

```
      MLOW = MASK(LOW)
      MHIGH = MASK(HIGH)
  280 IF (MHIGH-JY) 300, 300, 290
  290 IF (MLOW-JY) 310, 320, 320
C THE CURRENT POINT IS ABOVE THE MASK
  300 LOC = +1
      IF (LOCOLD) 360, 370, 430
C THE CURRENT POINT IS WITHIN THE MASK
  310 LOC = 0
      IF (LOCOLD) 340, 350, 330
C THE CURRENT POINT IS BELOW THE MASK
  320 LOC = -1
      IF (LOCOLD) 510, 450, 440
C PLOT FROM ABOVE THE MASK TO WITHIN THE MASK
  330 IF (MHIGH.LE.IYREF) CALL PL3QQC(JX, MHIGH, 2)
      GO TO 350
C PLOT FROM BELOW THE MASK TO WITHIN THE MASK
  340 IF (MLOW.GE.IYREF) CALL PL3QQC(JX, MLOW, 2)
C PLOT FROM WITHIN THE MASK TO WITHIN THE MASK
  350 CALL PL3QQC(JX, JY, 3)
      GO TO 520
C PLOT FROM BELOW THE MASK TO ABOVE THE MASK
  360 IF (MLOW-IYREF) 370, 380, 380
C PLOT FROM WITHIN THE MASK TO ABOVE THE MASK
  370 IF (MHIGH-IYREF) 400, 390, 390
  380 CALL PL3QQC(JX, MLOW, 2)
  390 CALL PL3QQC(JX, MHIGH, 3)
      GO TO 430
  400 IF (MHIGH.EQ.-1) GO TO 430
      OLDHI = HIGH - 2*INCX
      IF (MASK(OLDHI)-JY) 420, 420, 410
  410 CALL PL3QQC(JX, JY, 3)
      GO TO 430
  420 CALL PL3QQC(JX-INCX, MASK(OLDHI), 3)
C PLOT FROM ABOVE THE MASK TO ABOVE THE MASK
```

Figure A.2 (cont')

```
  430   MASK(HIGH) = JY
        IF (MLOW.EQ.-1) MASK(LOW) = JY
        CALL PL3QQC(JX, JY, 2)
        GO TO 520
C PLOT FROM ABOVE THE MASK TO BELOW THE MASK
  440   IF (MHIGH-IYREF) 460, 460, 450
C PLOT FROM WITHIN THE MASK TO BELOW THE MASK
  450   IF (MLOW-IYREF) 470, 470, 480
  460   CALL PL3QQC(JX, MHIGH, 2)
  470   CALL PL3QQC(JX, MLOW, 3)
        GO TO 510
  480   OLDLOW = LOW - 2*INCX
        IF (MASK(OLDLOW)-JY) 490, 500, 500
  490   CALL PL3QQC(JX, JY, 3)
        GO TO 510
  500   CALL PL3QQC(JX-INCX, MASK(OLDLOW), 3)
C PLOT FROM BELOW THE MASK TO BELOW THE MASK
  510   MASK(LOW) = JY
        CALL PL3QQC(JX, JY, 2)
  520   IYREF = JY
        LOCOLD = LOC
        IF (JX.NE.IX) GO TO 270
  530 CONTINUE
C RAISE PEN
      CALL PL3QQC(JX, JY, 3)
C STORE VERTICES IF REQUESTED
      IF (INDV.EQ.1) GO TO 540
      INDEX = -INCI + 6
      VERTEX(INDEX) = XX
      VERTEX(INDEX+1) = YY
      VERTEX(INDEX+8) = XXX
      VERTEX(INDEX+9) = YYY
      IF (NLINE.NE.1) GO TO 540
      VERTEX(3) = XX
      VERTEX(4) = YY
```

Figure A.2 (cont')

```
      VERTEX(11) = XXX
      VERTEX(12) = YYY
  540 I = I - 1
C RETURN TO CALLING PROGRAM
      RETURN
C OPTION TO MODIFY THE MASKING TECHNIQUE TO BE USED ON THE
C FOLLOWING FIGURE SO AS TO PLOT ONLY ABOVE ALL PREVIOUS
C LINES.
  550 INIT = 0
      RETURN
      END
C
C
      SUBROUTINE PL3QQB(IHCOR, VERTEX, MASK)
C ROUTINE TO PLOT A FRAME ON THE PROJECTIONOF A
C 3-DIMENSIONAL FIGURE AS DRAWN BY PL3QQA.
C INPUT PARAMETERS -
C     IHCOR    NUMBER OF THE VERTEX OF THE FIGURE WHICH
C              APPEARS TO BE FURTHEST IN THE BACKGROUND
C              (MINUS Z DIRECTION).
C     VERTEX   ARRAY CONTAINING THE COORDINATES OF THE
C              VERTICES OF THIS FIGURE AS RETURNED FROM
C              PL3QQA ON THE LAST CALL.
C     MASK     ARRAY CONTAINING THE MASK FOR THIS FIGURE
C              AS RETURNED BY PL3QQA ON THE LAST CALL.
C THE VERTICES OF THE FRAME ARE NUMBERED (1-4) IN THE SAME
C ORDER AS THEIR COORDINATES APPEAR IN VERTEX.
C THE MASK ARRAY IS ALTERED BY THIS ROUTINE,
C BUT THE PLOTTER ORIGIN IS NOT MOVED.
      DIMENSION VERTEX(1), MASK(1), ARRAY(14)
      I = 2*IHCOR
      IF (I.LT.2) I = 2
      IF (I.GT.8) I = 8
C THE VERTICES WHICH MAY BE HIDDEN
C ARE DRAWN BY A CALL TO PL3QQA.
```

Figure A.2 (cont')

```
      ARRAY(1) = VERTEX(I-1)
      ARRAY(8) = VERTEX(I)
      ARRAY(2) = VERTEX(I+7)
      ARRAY(9) = VERTEX(I+8)
      ARRAY(4) = ARRAY(2)
      ARRAY(11) = ARRAY(9)
      ARRAY(6) = ARRAY(2)
      ARRAY(13) = ARRAY(9)
      ARRAY(7) = ARRAY(1)
      ARRAY(14) = ARRAY(8)
      I = I - 2
      IF (I.EQ.0) I = 8
      ARRAY(3) = VERTEX(I+7)
      ARRAY(10) = VERTEX(I+8)
      I = I + 4
      IF (I.GT.8) I = I - 8
      ARRAY(5) = VERTEX(I+7)
      ARRAY(12) = VERTEX(I+8)
      CALL PL3QQA(110, ARRAY, ARRAY(8), 0.0, 1.0, 1.0, 0.0,
     * 2, 7, 0.0, 0.0, 0.0, 0.0, 0.0, MASK, 0.0)
C THE REMAINING VERTICES ARE DRAWN BY CALLS TO PLOT.
      CALL PL3QQD(VERTEX(I-1), VERTEX(I), 3)
      I = I - 2
      DO 10 J=1,3
        I = I + 2
        IF (I.EQ.10) I = 2
        CALL PL3QQD(VERTEX(I+7), VERTEX(I+8), 2)
   10 CONTINUE
      CALL PL3QQD(VERTEX(I-1), VERTEX(I), 2)
      I = I - 2
      IF (I.EQ.0) I = 8
      CALL PL3QQD(VERTEX(I-1), VERTEX(I), 3)
      CALL PL3QQD(VERTEX(I+7), VERTEX(I+8), 2)
      RETURN
      END
```

Figure A.2 (cont')

```
SUBROUTINE PL3QQD(X,Y,K)
INTEGER PIPI,K,II,JJ
REAL X,Y
DATA PIPI/64/
II = NINT(PIPI*X)
JJ = NINT(PIPI*Y)
CALL PL3QQC(II,JJ,K)
RETURN
END
SUBROUTINE PL3QQC(II,JJ,KK)
INTEGER II,JJ,KK,ORIGX,ORIGY,OLDX,OLDY,I1,I2,J1,J2
COMMON /PL3QQE/ORIGX,ORIGY,OLDX,OLDY
IF ((II-ORIGX).GT.640) II = 640 + ORIGX
IF ((II-ORIGX).LT.1) II = ORIGX+1
IF ((JJ-ORIGY).GT.512) JJ = 512+ORIGX
IF ((JJ-ORIGY).LT.1) JJ = ORIGY+1
IF (IABS(KK).EQ.2) THEN
  I1 = II-ORIGX-1
  I2 = OLDX - 1
  J1 = 200 - NINT((200./512.)*(JJ-ORIGY))-1
  J2 = 200 - NINT((200./512.)*OLDY) - 1
  IF (I1.LT.0) I1 = 0
  IF (I2.LT.0) I2 = 0
  IF (J1.LT.0) J1 = 0
  IF (J2.LT.0) J2 = 0
  IF (I1.GT.639) I1 = 639
  IF (I2.GT.639) I2 = 639
  IF (J1.GT.199) J1 = 199
  IF (J2.GT.199) J2 = 199
  CALL LINEQQ(I1,J1,I2,J2)
ENDIF
OLDX = II - ORIGX
OLDY = JJ - ORIGY
IF (KK.LT.0) THEN
  ORIGX = II - ORIGX
```

Figure A.2 (cont')

```
        ORIGY = JJ - ORIGY
      ENDIF
      RETURN
      END


Subroutine PRSCQQ


DATA    SEGMENT PUBLIC 'DATA'
DATA    ENDS
DGROUP  GROUP DATA
CODE  SEGMENT 'CODE'
      ASSUME CS:CODE,DS:DGROUP,SS:DGROUP;
PUBLIC PRSCQQ
PRSCQQ PROC FAR
      PUSH BP
      MOV BP,SP
      PUSH BP
      INT 5
      POP BP
      MOV SP,BP
      POP BP
      RET
      PRSCQQ ENDP
      CODE ENDS
END


Subroutine SCRNQQ


data  segment public 'DATA'
data  ends
```

Figure A.2 (cont')

```
dgroup  group DATA
code  segment 'CODE'
      assume cs:code,ds:dgroup,ss:dgroup;
public  scrnqq
scrnqq  proc  far
        push  bp    ;save framepointer
        mov   bp,sp
        les   bx,dword ptr [bp+6]
        mov   al,es:[bx]
        mov   ah,0
        int   10h
        mov   sp,bp
        pop   bp
        ret   4h
scrnqq  endp
code    ends
end
```



```
Subroutine LINEQQ

data segment public 'DATA'

delta_x dw  ?       ; |x2 - x1|
delta_y dw  ?       ; |y2 - y1|
halfy   label word  ; |y2 - y1| /2
halfx   dw  ?       ; |x2 - x1| /2
count   dw  ?       ; set to long axis
x1      dw  ?       ; first X coordinate
y1      dw  ?       ; first Y coordinate
x2      dw  ?       ; second X coordinate
y2      dw  ?       ; second Y coordinate
```

Figure A.2 (cont')

```
data  ends
dgroup  group DATA
code    segment 'CODE'
        assume cs:code,ds:dgroup,ss:dgroup;
public  lineQQ
lineQQ  proc  far
        push  bp    ;save framepointer on stack
        mov   bp,sp ;
        les bx,dword ptr [bp+18]  ;point at first parameter
        mov ax,es:[bx]  ;x1
        mov x1,ax
        les bx,dword ptr [bp+14]  ;point at second parameter
        mov ax,es:[bx]  ;y1
        mov y1,ax
        les bx,dword ptr [bp+10]  ;point at third parameter
        mov ax,es:[bx]  ;x2
        mov x2,ax
        les bx,dword ptr [bp+6]   ;point at fourth parameter
        mov ax,es:[bx]  ;y2
        mov y2,ax
        call linesub    ;draw ye line
        mov sp,bp       ;restore the framepointer
        pop bp
        ret 16d
lineQQ  endp        ;end of main part of program
;-------------------------------------------------------
linesub proc  near

;LINESUB--SUBROUTINE TO DRAW LINE
;
;Input is x1, y1 (start of line)
;         x2, y2 (end of line

;find |y2-y1| -- result is delta_y
        mov ax,y2     ;get y2
```

Figure A.2 (cont')

```
        sub ax,y1       ;subtract y1
                        ;result in AX
;figure out if delta_y is positive or negative
;  SI=1 if poistive, SI = -1 if negative
        mov si,1        ;set flag to positive
        jge store_y     ;  keep it that way
        mov si,-1       ;set flag to negative
        neg ax          ;set to abs value
store_y:
        mov delta_y,ax  ;store delta_y

;find |x2-x1| -- result is delta_x
        mov ax,x2       ;get x2
        sub ax,x1       ;subtract x1
                        ;result in AX
;figure out if delta_x is positive or negative
; DI=0 if positive, DI=1 if negative
        mov di,1        ;set flag to positive
        jge store_x     ;  keep it that way
        mov di,-1       ;set flag to negative
        neg ax          ;set to abs value
store_x:
        mov delta_x,ax  ;store delta_x

;figure out if slope is greater or less than 1
        mov ax,delta_x  ;get delta_x
        cmp ax,delta_y  ;compare deltas
        jl  csteep      ;slope > 1
        call easy       ;slope < 1, or = 1
        jmp finish

csteep:
        call steep      ;slope > 1

;DONE LINE--RETURN
```

Figure A.2 (cont')

```
finish: ret

linesub endp
;-------------------------------------------------------
easy  proc  near

;SLOPE < 1

;calculate half of delta_x, call it halfx
        mov ax,delta_x  ;get |x2-x1|
        shr ax,1        ;shift right to divide
        mov halfx,ax    ; by 2

;initialize values
        mov cx,x1       ;set x1
        mov dx,y1       ;set y1
        mov bx,0        ;initialize BX
        mov ax,delta_x  ;set count
        mov count,ax    ;to |x2-x1|
newdot:
        call dotplot    ;plot the dot
        add cx,di       ;inc/dec X
        add bx,delta_y  ;add |y2-y1| to BX
        cmp bx,halfx    ;compare to |x2-x1|/2
        jle dcount      ;  (don't inc/dec Y)
        sub bx,delta_x  ;subtract |x2-x1|
                        ;  from BX
        add dx,si       ;ind/dec Y
dcount:
        dec count       ;done line yet?
        jge newdot      ;not yet

        ret             ;done line
easy    endp
;-------------------------------------------------------
```

Figure A.2 (cont')

```
steep   proc  near

;SLOPE > 1
;calculate half of delta_y, call it halfy
        mov ax,delta_y  ;get |y2-y1|
        shr ax,1        ;shift right to divide
        mov halfy,ax    ;  by 2
;initialize values
        mov cx,x1       ;set x1
        mov dx,y1       ;set y1
        mov bx,0        ;initialize BX
        mov ax,delta_y  ;set count
        mov count,ax    ; to x2-y1

newdot2:
        call dotplot    ;plot the dot
        add dx,si       ;inc/dec Y
        add bx,delta_x  ;add |x2-x1| to BX
        cmp bx,halfy    ;compare to |y2-y1|/2
        jle dcount2     ;don't inc/dec X
        sub bx,delta_y  ;subtract |y2-y1|
                        ;  from BX
        add cx,di       ;inc/dec X
dcount2:
        dec count       ;done line yet?
        jge newdot2     ;not yet

        ret             ;return to main dline

steep   endp
;---------------------------------------------------------
dotplot proc  near

;SAVE REGISTERS AND CALL PLOT ROUTINE
```

Figure A.2 (cont')

```
        push  bx
        push  cx
        push  dx
        push  ax
        push  si
        push  di

;use ROM routine to write dot
;requires row # in DX, col in CX
        mov ax,1        ;set for black dot
        mov ah,12d      ;write dot function
        int 10h          ;vodeo BIOS routine

        pop   di
        pop   si
        pop   ax
        pop   dx
        pop   cx
        pop   bx
        ret             ;return
dotplot endp
;---------------------------------------------------------
      code  ends     ;end of code segment
end
```

Figure A.2 (cont')

The vita has been removed from the scanned document