NPS-57C072121A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

APPLICATION OF HOLOGRAPHIC INTERFEROMETRY TO DENSITY FIELD DETERMINATION IN TRANSONIC CORNER FLOW

by

D. J. Collins

and

R. A. Kosakoski

December 1972

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93940

Rear Admiral A. S. Goodfellow
Superintendent

M. U. Clauser
Provost

ABSTRACT

The successful application of holographic interferometry to the study of density fields around opaque bodies in wind tunnel experiments has been reported in the literature. The present report extends this technique to the study of the asymmetric flow fields encountered near the wing-fuselage junction of an aerodynamic model in the transonic flow regime. Finite fringe interferometry has been used to investigate the three-dimensional density field about a partially transparent wing-body structure. The resulting asymmetric density field and shock wave structure are shown to be an accurate representation of the phenomena encountered in aerodynamic corner flow.

This report was supported by : ONR Req. 00014-2-000033, 15 Sep 71 and NASC W. R. 2-6059, 12 Jul 71.

NPS-57C072121A

TABLE OF CONTENTS

INTRODUCTION

This is a report on holographic investigations directed towards the determination of three-dimensional density fields in transonic corner flows. The report consists essentially of Appendix I, which includes the thesis of LT R. A. Kosakoski on holographic investigations of transonic corner flow. This investigation is the first to use transparent models in holographic interferometry.

A paper based on this thesis, "Application of Holographic Interferometry to Density Field Determination in Transonic Corner Flow," by R. A. Kosakoski and D. J. Collins (No. 73-156) has been presented at the AIAA 11th Aerospace Sciences Meeting in Washington, D. C. The paper is now under consideration for publication in the AIAA Journal.

The integral inversion method used in the thesis and paper has now been essentially superseded by a more efficient method based on Fourier Transforms. This latter method is the subject of another report.

Further work on investigation into the application of fiber obptics in holographic interferometry is continuing. This study also involves the consideration of flow with limited fields of view.

APPENDIX I

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

APPLICATION OF
HOLOGRAPHIC INTERFEROMETRY
TO
DENSITY FIELD DETERMINATION
IN TRANSONIC CORNER FLOW

by

Robert Anthony Kosakoski

Thesis Advisor D. J. Collins

December 1972

Application of Holographic Interferometry
to Density Field Determination
in Transonic Corner Flow

by

Robert Anthony Kosakoski
Lieutenant, United States Navy
B.S.M.E., University of Rochester, 1965

Submitted in partial fulfillment of the
requirements for the degree of

AERONAUTICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1972

ABSTRACT

The successful application of holographic interferometry to the study of density fields around opaque bodies in wind tunnel experiments has been reported in the literature. The present report extends this technique to the study of the three-dimensional asymmetric flow fields encountered near the wing-fuselage junction of an aerodynamic model in the transonic flow regime. Finite fringe interferometry has been used to obtain fringe information about a partially transparent wing-body structure. A FORTRAN computer program was utilized to invert the fringe information and produce a plot of the density field around the model. The resulting asymmetric density field and shock wave structure are shown to be an accurate representation of the phenomena encountered in aerodynamic corner flow.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This writer wishes to gratefully acknowledge Dr. D. J. Collins for his most valuable guidance and assistance during the course of this investigation; the Aerodynamics Division of the Naval Ship Research and Development Center, Carderock, Maryland, under Dr. R. Furey, particularly B. Ellis and O. Gilmore, for their tireless effort during the wind tunnel testing phase; the technical staff of the Department of Aeronautics under R. Besel and T. Dunton, particularly N. Leckenby and G. Middleton, for their able assistance in the fabrication and assembly of the models and equipment used in this study; and my wife and family for their noble patience and encouragement.

I. INTRODUCTION

The field of flow measurement has been revolutionized in recent years with the perfection of holography and holographic interferometry techniques. High power Q-switched and dye-switched lasers and sophisticated double-pulsing trigger mechanisms provide exposure times on the order of twenty nanoseconds, thereby "freezing" the flow during the hologram production process. The precision optical quality components and measurement techniques of Mach-Zehnder interferometry have given way to the much less restrictive requirements of holographic interferometry which provide high quality interferograms in three dimensions.

Techniques for the application of holography to interferometry have been reported by Heflinger, et al. [1], and by Brooks [2]. In the determination of the density field around a free jet in the supersonic regime, Matulka [3, 4] expressed the fringe data in a series of orthogonal polynomials and transformed them to polynomials representing density using an inversion technique reported in [5, 6]. The method was extended by Jagota [7, 8] to the determination of the three-dimensional density field around a ten-degree half angle cone in a supersonic wind tunnel. The ability to produce readable holograms in wind tunnel studies using transparent phase objects was verified by Heyer [9]. In the present report the aforementioned techniques have been combined to study the three-dimensional density field near the wing-fuselage junction of an aerodynamic model in transonic flow. The experiment was conducted at the Naval Ship Research and Development Center, Carderock, Maryland in an eighteen inch transonic blow-down wind tunnel at a Mach number of 0.937, using a semi-transparent model of original design.

Since reasonably small variations in density were anticipated, a finite fringe technique was used in obtaining the interferograms. The horizontal finite fringe field was produced by a vertical translation of a diffusing glass in the scene beam a distance of 0.003 inches between the two exposures of the holographic plate. Fringe data obtained from the interferograms were reduced to density information using a modified form of the inversion computer program used in [7]. A self-testing procedure incorporated in the program verified the resulting density data as an accurate representation of the actual flow around the model.

II. EXPERIMENTAL APPARATUS

A. THE WIND TUNNEL

The investigation was conducted in the Naval Ship Research and Development Center blowdown supersonic wind tunnel. Transonic flow conditions were produced through incorporation of slotted upper and lower tunnel walls. The tunnel is a nominal eighteen inch blow-down-to-vacuum facility with a test section fourteen inches by eighteen inches in cross-section and twenty-nine inches in length with the slotted surfaces installed. Optical quality windows twenty-two inches in diameter in the side walls provided complete viewing of the flow in the test section as the model was rolled through 180 degrees for hologram production. A functional schematic of the wind tunnel is shown in Figure 1.

B. THE MACH NUMBER AND PRESSURE MEASUREMENT PROCEDURE

The Mach number at the test section is determined as a function of total and static pressure measurements and is maintained by carefully controlled butterfly valve settings. Total pressure is determined by recording atmospheric pressure prior to tunnel operation and accounting for the pressure loss between the plenum and the test section during operation. Static pressure is measured directly at a central wall port in the test section. Data recordings were made on a Beckman Instruments Company 210 Digital Recorder, shown in Figure 2, and were read out on line via a Franklin Strip Tape Printer.

C. THE HOLOGRAPHIC ARRANGEMENT

The holographic arrangement is illustrated in Figure 3 and shown in photographs included as Figures 4, 5, 6, and 7. Two large wooden tables were constructed and linked together with two-by-four beams under the tunnel to form the experimental platform. Thick rubber pads were attached to the table legs to dampen possible floor vibrations. The bulk of the platform provided sufficient stability and vibration damping for the experiment. The monochromatic light source used was a KORAD K-1 pulsed ruby laser operating at a wavelength of 6943 Angstroms, together with a Pockels cell Q-switching device. The resultant effective exposure time was approximately twenty nanoseconds, eliminating the problems due to possible model vibration during hologram exposure. To maintain the laser head and output etalon at a constant temperature of 27.0 degrees Centigrade, a LAUDA constant temperature circulator Model K2R was used.

The reference beam was directed under the wind tunnel by four front surface mirrors, and the beam size was manipulated by means of lenses (Figure 3). The scene beam was routed through the test section to intersect the reference beam on the holographic plate at an angle of approximately fifty degrees. A diffuse glass, mounted on a precision X-Y translation table in the scene beam, was used to produce light field holograms. Alignment of the Q-switched laser and system optics was accomplished using a continuous wave, low-power helium-neon laser. Reference grids were mounted accurately on the outer surfaces of the tunnel windows using a survey's transit. Details of the model mounting and reference grids are shown in Figure 7. To enable hologram production during daylight hours, the entire tunnel room was blacked out using drop curtains and light shields.

D. THE WIND TUNNEL MODEL

The aerodynamic model used is shown in Figures 8, 9, and 10. The metal portions of the model were stainless steel. The greater part of the modified double wedge platform wing was constructed of optical lucite, as was the portion of the fuselage at the wing root. Detailed model dimensions are shown in Figure 11. The choice of aerodynamic design provided good flow characteristics and a relatively stable lambda-type shock wave on the wing; the largely transparent construction facilitated hologram production through 180 degrees of view.

The model was rotated about its sting mount in the wind tunnel from the zero degree position, wings level, to the 180 degree position, wings level inverted. Alignment for the desired rotation angle was accomplished by manually aligning prescribed lines on the sting mount collar with a scribed mark on the sting support.

III. ANALYTICAL EVALUATION OF THE DENSITY FIELD

A. THE BASIC EQUATION OF INTERFEROMETRY

Interferograms are created when two originally coherent light beams are superimposed and projected on a viewing screen. The two rays will reinforce or annul each other, depending on their relative phase difference at the screen. This phase difference is directly a function of the optical pathlengths traversed by the two waves. Consider a coherent beam which is split and then recombined on a viewing screen. A difference in optical pathlengths of the two component beams may be achieved by causing the beams to traverse through different media prior to recombination, with their physical pathlengths maintained equal. Each component beam will travel at a speed c_0/n where c_0 is the speed of light in a vacuum and n is the index of refraction of the medium traversed. The difference in optical pathlength is then given by

$$L = L\,(n_2 - n_1) = c_0 \Delta t \qquad (1)$$

where Δt is the time difference of travel in the two media. If the optical pathlength is changed by an amount $N\lambda$, where λ is the wavelength of the light source and N is an integer, then the order of interference changes by an amount N. In other words, a shift of N fringes occurs in the interference pattern. The fringe shift may be expressed as

$$g = L/\lambda \tag{2}$$

where g = fringe shift

λ = light source wavelength

L = change in optical pathlength

Substituting equation (1) into equation (2) yields

$$g = \frac{L}{\lambda}(n_2 - n_1) \tag{3}$$

The index of refraction for a given medium is a function of density. In the case of gases, since the speed of light is very nearly the same as in a vacuum, the index of refraction is well represented by the first two terms of a Taylor series expansion [10]:

$$n = 1 + \beta \frac{\rho}{\rho_s} \tag{4}$$

where β = dimensionless constant related to the Gladstone-Dale constant by $K = \beta/\rho_s$

ρ_s = reference density at 0° C, 760 mm. Hg.

The value of β for air at λ = 5893 Angstroms (deep red light) is 0.000292; variation with wavelength is very small.

For a fixed difference in the index of refraction between the two component beams the fringe shift relation becomes:

$$g = \beta \frac{L}{\lambda}\left(\frac{\rho_2 - \rho_\infty}{\rho_s}\right) \tag{5}$$

For variable density in the test section, the net change in optical pathlength is the integrated effect along the beam path, or

$$g = \frac{\beta}{\lambda \rho_s}\int_0^L (\rho - \rho_\infty)\,ds = Q \int_0^L f(x, y, z_c)\,ds \tag{6}$$

where: $$Q = \frac{\beta \rho_\infty}{\lambda \rho_s} \qquad (7)$$

$$f(x, y, z_c) = \frac{\rho(x, y, z_c)}{\rho_\infty} - 1 \qquad (8)$$

z_c = plane of constant z

ds = incremental distance along beam path

Equation (6) is the basic integral equation for the unknown density. With known fringe shift values from an interferogram, the equation is inverted to obtain the density along a beam path.

B. THE INTEGRAL INVERSION

The integral inversion procedure utilized in this investigation was first reported by C. D. Maldonado, et al [5, 6]. It was used subsequently by R. D. Matulka [3] and R. C. Jagota [7] to determine the density variation in an asymmetric free jet and about a cone at angle of attack in supersonic flow, respectively. The procedure involves the representation of the function $f(x, y, z_c)$ of Equation (6) in a complete set of orthogonal functions, with the expansion coefficients evaluated by use of the orthogonality condition between the functions f and g of Equation (6). The set of functions used is orthogonal over the entire (x, y) plane for every z_c and remains an orthogonal set under a rotation of the coordinate system. The coordinate system used for the inversion is shown in Figure 12. It consists of (a) a set of fixed coordinates x, y and (b) a set of moving coordinates x', y' in which the fringe number function is defined and which rotates with respect to x, y as the viewing angle through the test section is varied.

In operator form, Equation (6) can be represented as

$$g(\xi, y', z_c) = \mathcal{T} f(x, y, z_c) \tag{9}$$

and f is evaluated by inversely transforming the equation to obtain

$$f(x, y, z_c) = \mathcal{T}^{-1} g(\xi, y', z_c) \tag{10}$$

This inversion is achieved by utilizing a pair of orthogonal polynomials $U_{m+2k}^{\pm m}(\alpha x, \alpha y)$ and $H_{m+2k}(\alpha y')$ which are related by the transform relationship

$$\mathcal{T}\left[U_{2k}(\alpha x, \alpha y) e^{-\alpha^2 x'^2}\right] = \frac{e^{\pm i m \xi}}{[k!(m+k)!]^{1/2}} \cdot \frac{1}{2^{m+2k}} \cdot H_{m+2k}(\alpha y') \tag{11}$$

where $H_{m+2k}(\alpha y')$ are Hermite polynomials of order m+2k. The unknown function $f(x, y, z_c)$ is expanded in a set of functions $U_{m+2k}^{\pm m}$ as

$$f(x, y, z_c) = \sum_{m=0}^{\infty} \sum_{k=0}^{\infty} \varepsilon_m \Big\{ C_{m+2k}^{+m}(\alpha) U_{m+2k}^{+m}(\alpha x, \alpha y) + C_{m+2k}^{-m}(\alpha) U_{m+2k}^{-m}(\alpha x, \alpha y) \Big\} e^{-(\alpha^2 x^2 + \alpha^2 y^2)} \tag{12}$$

where $\varepsilon_m = 1/2$ for m = 0, $\varepsilon_m = 1$ for m = 1, 2, 3, . . . , and $C_{m+2k}^{\pm m}$ are the unknown coefficients of the expansion. α is an arbitrary scale factor which may be considered the reciprocal of a non-dimensionalizing coefficient.

Utilizing the transform relation of Equation (11), Equation (6) becomes

$$g(\xi, y', z_c) = \sum_{m=0}^{\infty} \sum_{k=0}^{\infty} \varepsilon_m \left[k!\,(m+k)!\; 2^{2(m+2k)}\right]^{1/2}$$
$$\times \left[C_{m+2k}^{+m}(\alpha)\, e^{im\xi} + C_{m+2k}^{-m}(\alpha)\, e^{-im\xi}\right] H_{m+2k}(\alpha)\, e^{-\alpha^2 y'^2} \tag{13}$$

Equation (13) is subject to the orthogonality condition

$$\int_{-\pi}^{\pi} e^{\pm im\xi} e^{\mp in\xi} d\xi \int_{-\infty}^{+\infty} H_{m+2k}(\alpha y') H_{n+2\ell}(\alpha y')\, e^{-\alpha^2 y'^2} dy'$$
$$= \frac{2\pi^{3/2}}{\alpha} \left[(m+2k)!\,(n+2\ell)!\, 2^{m+2k} 2^{n+2\ell}\, \delta_{mn}\, \delta_{(m+2k)(n+2\ell)}\right] \tag{14}$$

where δ is the Kronecker delta function. The solution of Equation (14) applied to Equation (13) yields the series coefficients

$$C_{m+2k}^{\pm m}(\alpha) = \frac{\alpha}{2\pi^{3/2}} \left[\frac{k!\,(m+k)!}{(m+2k)!}\right] \int_{-\pi}^{\pi}\int_{-\infty}^{\infty} g(\xi, y', z_c)\, H_{m+2k}(\alpha y')\, e^{\mp im\xi}\, dy'\, d\xi \tag{15}$$

With the substitution of the coefficients of Equation (15), Equation (7) becomes

$$f(x, y, z_c) = \frac{\alpha}{\pi^{3/2}} \sum_{m=0}^{\infty} \sum_{k=0}^{\infty} \varepsilon_m \frac{[k!\,(m+k)!\,]^{1/2}}{(m+2k)!}\, e^{-(\alpha^2 x^2 + \alpha^2 y^2)}$$
$$\times \mathrm{Re}\left[\int_{-\pi}^{\pi}\int_{-\infty}^{\infty} g(\xi, y', z_c)\, e^{-im\xi} H_{m+2k}(\alpha y')\, dy'\, d\xi \times U_{m+2k}^{m}(\alpha x, \alpha y)\right] \tag{16}$$

The functions $U_{m+2k}^{\pm m}$ are defined as

$$U_{m+2k}^{\pm m}(\alpha x, \alpha y) = (-1)^k \alpha \left[\frac{k!\,(\alpha^2 x^2 + \alpha^2 y^2)^m}{\pi (m+k)!}\right]^{1/2} e^{\pm im\phi}\, L_k^m(\alpha^2 x^2 + \alpha^2 y^2) \tag{17}$$

where $\phi = \tan^{-1}(y/x) - (\pi/2)$ and L_k^m are the associated Laguerre polynomials

$$L_k^m(\alpha^2x^2+\alpha^2y^2) = \sum_{s=0}^{\infty} \frac{(m+k)!}{(k-s)!(m+s)!\,s!}\left[(-1)(\alpha^2x^2+\alpha^2y^2)\right]^s \tag{18}$$

Insertion of Equation (17) into Equation (16) yields

$$f(x,y,z_c) = \left(\frac{\alpha}{\pi}\right)^2 \sum_{m=0}^{\infty}\sum_{k=0}^{\infty} \varepsilon_m \frac{(-1)^k k!}{(m+2k)!}(\alpha^2x^2+\alpha^2y^2)^{m/2} L_k^m(\alpha^2x^2+\alpha^2y^2)$$
$$\times\left[B_{m+2k}^m(\alpha)\cos(m\phi)+D_{m+2k}^m(\alpha)\sin(m\phi)\right]e^{-(\alpha^2x^2+\alpha^2y^2)} \tag{19}$$

where

$$B_{m+2k}^m(\alpha) = \int_{-\pi}^{\pi}\int_{-\infty}^{\infty} g(\xi,y',z_c)\cos(m\xi)\,H_{m+2k}(\alpha y')\,dy'\,d\xi \tag{20}$$

$$D_{m+2k}^m(\alpha) = \int_{-\pi}^{\pi}\int_{-\infty}^{\infty} g(\xi,y',z_c)\sin(m\xi)\,H_{m+2k}(\alpha y')\,dy'\,d\xi \tag{21}$$

Equations (19), (20), and (21) are the basic equations used to obtain the density distribution from the experimentally determined fringe variations in a completely asymmetric flow field.

C. THE NUMERICAL PROCEDURE

Because the function $g(\xi, y', z_c)$ is an experimentally determined quantity the unknown coefficients $B_{m+2k}^m(\alpha)$ and $D_{m+2k}^m(\alpha)$ in the series representation of $f(x, y, z_c)$ in Equation (19) cannot be calculated analytically. It is therefore necessary to evaluate the double integrals of Equations (20) and (21) numerically. This is accomplished by noting in Figure 12 and Equation (8) that there is an area outside which the density is invariant, namely outside the test section where the known density is ρ_∞. Since the function f(x, y, z_c) = 0 outside this circular domain, the limits of integration of + ∞ and - ∞ in Equations (20) and (21) can be replaced by finite values. The fringe distribution is then approximated by small increments over the test domain, resulting in the representation of the B and D coefficients as double series:

$$B_{m+2k}^{m}(\alpha) = \sum_{i=1}^{I-1}\sum_{j=0}^{J-1} g(\xi_j + \Delta\xi_j, X_i + \Delta X_i) \int_{\xi_j}^{\xi_{j+1}} \cos(m\xi)\, d\xi \int_{X_i}^{X_{i+1}} H_{m+2k}(\alpha x)\, dx \tag{22}$$

and

$$D_{m+2k}^{m}(\alpha) = \sum_{i=1}^{I-1}\sum_{j=0}^{J-1} g(\xi_j + \Delta\xi_j, X_i + \Delta X_i) \int_{\xi_j}^{\xi_{j+1}} \sin(m\xi)\, d\xi \int_{X_i}^{X_{i+1}} H_{m+2k}(\alpha x)\, dx \tag{23}$$

Using the derivative formula for Hermite polynomials, Equations (22) and (23) can be manipulated to yield workable series expressions:

$$B_{m+2k}^{m}(\alpha) = \left[\frac{1}{2\alpha m} \cdot \frac{1}{(m+2k+1)}\right] \sum_{i=0}^{I-1}\sum_{j=0}^{J-1} g(\xi_j + \Delta\xi_j, X_i + \Delta X_i)$$
$$\times \left[\sin(m\xi_{j+1}) - \sin(m\xi_j)\right]\left[H_{m+2K+1}(\alpha X_{i+1}) - H_{m+2K+1}(\alpha X_i)\right] \tag{24}$$

$$D_{m+2k}^{m}(\alpha) = -\left[\frac{1}{2\alpha m} \cdot \frac{1}{(m+2k+1)}\right] \sum_{i=0}^{I-1}\sum_{j=0}^{J-1} g(\xi_j + \Delta\xi_j, X_i + \Delta X_i)$$
$$\times \left[\cos(m\xi_{j+1}) - \cos(m\xi_j)\right]\left[H_{m+2K+1}(\alpha X_{i+1}) - H_{m+2k+1}(\alpha X_i)\right] \tag{25}$$

Since it is impossible to sum over an infinite number of terms, Equation (19) is necessarily expressed as the sum of a finite series:

$$f(x,y,z_c) = \left(\frac{\alpha}{\pi}\right)^2 \sum_{k=0}^{K} \sum_{m=0}^{M} \varepsilon_m (-1)^k \left[\frac{k!}{(m+2k)!}\right] (\alpha^2x^2 + \alpha^2y^2)$$

$$\times \left[L_k^m(\alpha^2x^2+\alpha^2y^2) \left[B_{m+2k}^{m}(\alpha)\cos(m\phi) + D_{m+2k}^{m}(\alpha)\sin(m\phi) \right] e^{-(\alpha^2x^2+\alpha^2y^2)} \right. \qquad (26)$$

It has been demonstrated that judicious selection of the parameters $\Delta\xi$, Δ x, K, M, and α yields density distributions with very good accuracy [3, 6].

IV. EXPERIMENTAL PROCEDURE

A. LABORATORY TECHNIQUES

In order to visualize the general flow patterns and localize shock or expansion waves about the model a series of standard Schlieren photographs were taken at varying flow Mach numbers. Pictures were produced for roll angles of 0^{o}, 45^{o}, and 90^{o} at Mach numbers from 0.925 to 1.10. A representative series of Schlieren photographs is shown in Figures 13, 14, and 15. Analysis of the Schlieren photographs dictated a flow Mach number of 0.937 for the experimental study; this Mach number yielded uniform upstream flow conditions and located the lambda-type shock wave ideally near the center of the lucite section of the model wing.

The coherence length of the pulsed ruby laser was approximately ten centimeters for the output power utilized. This reduced the normally critical requirement for pathlength equality in the scene and reference beams that must be fulfilled in the classical Mach-Zehnder interferometric approach. Consequently, a length of string proved to be a sufficiently accurate measuring device to maintain the two beam pathlengths within the coherence length of the laser, a requirement for interferogram production. To compensate for the fact that the scene beam traversed approximately five inches of glass tunnel walls and lucite grids which the reference beam did not, the scene beam was adjusted to be some 2.5 inches shorter than the reference beam. Reference beam pathlength was maintained at approximately 138 inches throughout the experiment.

Holograms produced using the basic holographic setup shown in Figure 3 exhibited clear, well-defined fringe patterns in nearly every instance. In deciding on the final arrangement, several techniques were tested to improve upon fringe pattern definition. Horizontal, vertical and diagonal translations of the diffuser plate in the scene beam were considered, varying from 0.001 inches to 0.005 inches. A vertical translation of 0.003 inches yielded clear horizontal fringes that were quite easily analyzed. The transverse mode selector aperture was varied from 1.0 mm. to 3.0 mm. in increments of 0.5 mm; best lighting of the model resulted with use of a 2.5 mm. aperture. The temperature of the cooling water circulated through the laser head and etalon was varied from 26.0°C. to 28.0°C. in increments of 0.2°C., with 27.0°C. providing the best fringe definition. Finally, a variety of beam splitters and lenses were tested prior to final selection of the best available optics arrangement for the experiment. A 2:1 reference to scene beam strength ratio was found to yield very good holograms.

Two double exposure holograms were taken for each model viewing angle. The first, labeled a double-static exposure, consisted of two exposures in a no-flow condition with a 0.003 inch vertical translation of the diffuser plate between exposures. The fringe patterns in this hologram provided a measure of the effect of tunnel wall glass, grid plexiglass and model lucite on the subsequent double exposure. The second, or static-dynamic, exposure consisted of a no-flow exposure, a 0.003 inch diffuser translation, and finally an exposure at flow Mach number 0.937. The fringe deviations recorded in the region behind the lucite portion of the model by the double-static hologram were measured and subtracted from the fringe shifts measured in the corresponding static-dynamic hologram.

Holograms were produced on Agfa-Gaevert 8E75 holographic plates, 4 inches by 5 inches in size. As recommended by Collier, et al. [11] the development process included:

1. Five minutes in Kodak D-19 developer
2. Thirty seconds in a flowing water bath
3. Five minutes in standard rapid fixer
4. Thirty seconds in a flowing water bath
5. One and one-half minutes in Kodak Hypo Clearing agent
6. Five minutes in a flowing water bath
7. Five minutes in methanol bath
8. One minute in a flowing water bath
9. Drying

B. PHOTOGRAPHIC TECHNIQUES

Normal reconstructions of the original scene were made by illuminating the holograms with a seven milliwatt continuous wave helium-neon laser beam at a wavelength of 6328 Angstroms. There was some slight distortion in the reconstructed scene because of the difference in wavelengths of the original scene beam and the reconstruction beam; however, the effect was almost totally negated by shrinkage of the holographic plate emulsion during the development process.

A common technique of image reconstruction was employed, utilizing a conjugate reference beam to reilluminate the exposed hologram, as shown in Figure 16. The resulting scene was recorded on photographic film. Individual points on the photograph are produced by a series of non-parallel rays originating from various source points on the diffuser plate in the scene beam. Using a reilluminating beam of small diameter has the effect of a small aperture at the focal point of the imaging lens, filtering out all but a set of nearly parallel rays, as shown in Figure 17. The real images produced in this manner have a large depth of field, permitting simultaneous projection on the film of front and rear grids, the model and the fringe patterns. The imaging lens was focused as near to the plane of the fringes as possible, producing photographs at various planes of constant z_c.

C. DATA REDUCTION

Photographic interferograms were obtained using the arrangement shown in Figure 16, with the camera viewing screen in the position of the real image. The line of sight in the plane desired was achieved by translating and elevating the hologram until common points on the front and rear grids were aligned. The Graphic View camera, with the wide angle lens aperture set fully open at f7.8, was adjusted to yield the best focus on the fringe plane. Exposure times of from 1/5 to 3/4 seconds were used to produce workable interferogram photographs on Polaroid Type 55 P/N film.

Fringe shift analysis was accomplished on 8 inch by 10 inch enlargements of the 4 inch by 5 inch film used to record the images. The enlargements were placed face down on a light table, and the fringes, model contours and shock wave were traced on the back surface at the desired cross-sectional plane. Fringe shift values were recorded by measuring the distance between (1) the intersection of the hypothetically undeviated fringe and the cross-sectional plane and (2) the intersection of the deviated fringe with the same plane. Fringe shifts so obtained were corroborated by placing the negative in a photographic enlarger and tracing the lines of interest directly onto graph paper.

The known model fuselage diameter of 1.1 inches was compared with that measured in each individual photograph to yield magnification factors relating projected dimensions to actual dimensions. These factors were then used as corrections to the fringe shift measurements. A grid reference point located on the cross-sectional plane of interest served as the datum for all fringe shift measurements. A base point was located at the intersection of the cross-sectional plane of interest and the body longitudinal axis. Fringe shift measurements were corrected using this base point as the new datum so that the inversion circle was properly centered on the body axis. The radius of the inversion circle was selected to be nearly equal to the semi-span of the wing. Fringe shift measurements were converted to fringe numbers using the average free stream spacing, while fringe locations were nondimensionalized using the inversion circle diameter. From the data so obtained, the radial variation of fringe number was plotted for each viewing angle. Fringe numbers at 201 equidistant points, as required for input into Mode 3 of the computer program, HOLOFER, were recorded from the resulting curves. Further details concerning this inversion computer program are outlined in Appendix B. A typical reduction of an interferogram to obtain the radial variation of fringe number at a particular cross-sectional plane is detailed in Appendix A.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A pair of double exposure holograms was taken of the model at 11¼ degree intervals through a 180 degree field of view. Experimental data from the wind tunnel runs are recorded in Table I. Initial resulting density patterns indicated relatively smooth contours across adjacent intervals; the interval was therefore doubled to 22½ degrees to simplify and speed the analysis. Fringe data were first inserted into the inversion computer program along nine lines of sight in the 180 degree field of view. A numerical comparison of views from 0 degrees to 90 degrees and from 90 degrees to 180 degrees verified to within 0.20 percent the assumption of a single plane of symmetry in the experiment. The fringe data input was then reduced to five lines of sight in a 90 degree field of view, as shown in Figure 18. The resulting output was an inverted density field along nine radial lines spanning a 180 degree field of view, with a mirror image on the opposite side of the plane of symmetry, as shown in Figure 19.

The static-dynamic photographic interferogram for the 0 degree view, along with its corresponding double-static interferogram, is shown in Figure 20. The diffraction effects caused by the presence of the lucite portions of the model are clearly visible in the double-static exposure, where the free stream fringe lines are bent and displaced toward the model axis. This displacement was measured and subtracted from subsequent measurements made on the static-dynamic exposure, as outlined in Section IV.A. Photographic interferograms of the remainder of the static-dynamic exposures are shown in Figures 21 through 24. Clearly visible and reduceable in nearly all views were (1) the region of uniform subsonic flow, commonly called the free stream condition, (2) the transition from local subsonic to local supersonic flow, and (3) the lambda-type shock wave on the model wing. These characteristics are shown in schematic representation in Figure 25.

Contour plots of the density function, as expressed in Equation (8) of Section III.A., for successive z-planes of analysis are shown in Figures 26 and 27. The cross-sectional plane of analysis for the plot of Figure 26 was located at 186.75 mm. from the model nose along the longitudinal axis. For Figure 27, the plane of interest was 195.25 mm. from the model nose. It is apparent from both contour plots that the model went to a very small angle of attack under the loading forces produced during tunnel operation; this is evidenced by the compression of the contour lines above the model and the corresponding expansion of the contours below the model. Measurements made from photographic interferograms confirm this angle of attack to be, at most, 0.05 degrees. The closed contours above and adjacent to the wing surface in both figures may very well be the result of a vortex originating at the intersection of the wing leading edge and the fuselage on either side of the model and traveling aft and outward over the wing surface.

A comprehensive quantitative analysis of the shock wave structure was not undertaken, with the exception of estimating the strength of the shock by comparison of fringe line separation immediately ahead of and aft of the shock wave. Fringe line separation measurements on either side of the shock wave were converted first to density information and thence to pressure information, disregarding compressibility effects. An approximate strength value of 0.207 was computed using the accepted definition of $(p_2 - p_1)/p_1$. This corresponds to a local Mach number of 1.08 in the supersonic region just ahead of the shock wave. Qualitative shock wave analysis resulted in the construction of a three-dimensional structural representation as shown in Figure 28, using input information from several interferogram viewing angles. While location of the leading and trailing edges of the lambda-type shock wave was very accurate, interior structure was largely indefinable due to "smearing" and blurring of the fringes transiting the shock wave itself.

As a preliminary step to possible future studies in this field, photographic interferograms were made from holograms produced with the aerodynamic model set at small angles of attack. Orientations included angles of attack of five and ten degrees, with roll angles varying from zero to ninety degrees. Although the holograms themselves were of very good quality, the photographic reproductions were relatively poor due to the fact that an inferior photographic arrangement had to be used. They were therefore omitted from this report. It was inferred from the holograms, however, that a complete study at angle of attack using the basic procedures followed in the present study would be both totally feasible and rewardingly fruitful.

The original character of the experimental data prevented comparison with published results. Qualitative studies of transonic phenomena are widely available, and the general characteristics of the resulting density field and shock wave structure serve to bear out the schematics based on theoretical and mathematical models. Moreover, the self-testing mode of the inversion computer program, HOLOFER, verified the consistency and reproductibility of the resulting density distributions to within 2.0 percent through proper choice of the input parameters, primarily the slope-matching parameter α. The errors encountered in the final results are due primarily to errors in the fringe data input to the inversion program. The intrinsic presence of laser speckle, the extended pathlengths of the scene and reference beams and the unavoidable beam scattering and diffraction within the lucite model sections, created difficulty in obtaining precisely the slope of the fringe lines behind the lucite sections. Fringe spacing measurements in the free stream flow were conservatively judged accurate to within 0.5 mm. This assumption was quite reasonable since all measurements were effected with a scale graduated at half millimeter intervals. This figure of 0.5 mm., combined with the mean free stream fringe spacing of 5.197 mm. for all interferograms, indicated measurement accuracy to within one tenth (0.1) of a fringe. The mean systematic error of the free stream spacing in each view was computed to be a maximum of 3.9 percent. Associated with this systematic error was a random error of 2.1 percent in the measurement of fringe shifts in each view to a conservative accuracy of 0.5 mm. The resulting error for each viewing angle was therefore a maximum of 6.0 percent, found by merely adding the two types of error for each view. The minimum error limit was found by considering the error resulting from the reproduction of the same interferogram view five separate times. Statistically, with 6.0 percent error in each view, the composite error for the repeated view is 2.6 percent. As five different views, or lines of sight, were used for the data between zero and ninety degrees, the final total error in the analysis was therefore in the interval between 2.6 percent and 6.0 percent. To insure contour clarity and guard against overlapping, the maximum error figure of 6.0 percent was used in construction of the plots shown in Figures 26 and 27. In general, the rather large fringe shifts led to a very low mean fringe sensitivity value of 0.1259. This coefficient indicated a resulting density function (Equation (8)) inaccuracy of less than 1.5 percent for a fringe shift measurement misreading of 0.5 mm.

Physical limitations of beam diameter and hologram plate area dictated the choice of an inversion circle diameter somewhat smaller than the full data circle normally used in the finite fringe procedure. This, in effect, introduced an inconsistency in reference density into the analysis since the density on the selected circle and immediately outside it was not the calculated ρ_∞ ; the density function was therefore not zero outside the actual inversion region. To alleviate this inconsistency, a new, updated reference density was computed for each cross-sectional plane of analysis by averaging the density values on the selected circle from the first inversion process. The actual reference densities used were ρ = 1.777 mg/cc for the 186.75 mm. plane and ρ = 1.642 mg/cc for the 195.25 mm. plane. This procedure was justified since all density values between the selected circle and the full data circle were constant to within approximately fifteen percent. The updated reference densities were then used to produce the final output density field. The net effect was a scaled, uniform shift toward density function values slightly lower than those computed on the basis of the original reference density.

VI. CONCLUSIONS

The finite fringe procedure for the production of holographic interferograms has been applied successfully to the determination of the three-dimensional density distribution of the flow near the wing-fuselage junction of a partially transparent aerodynamic model in the transonic regime. Density contours accurate to within six percent enabled a thorough analysis of the flow field to be conducted, highlighting flow characteristics and the presence of the shock wave. Subsequent studies of similar models at angle of attack have been shown to be entirely feasible. Procedures used in the experiment also exhibit promise for the direct analysis of duct and inlet flows as well as comprehensive study of shock wave structure.

The inversion computer program, HOLOFER, was found to be adequate in handling a general asymmetric flow field analysis. However, it was considered quite cumbersome and difficult to modify for various experimental situations. Subsequent analysts will find the procedures advocated by Sweeney and Vest [12] for the recording and analyzing of interferograms of considerable interest. In addition, the efforts of Van Houton [13], who utilized the method proposed by Junginger and van Haeringen [14], may prove valuable in reducing computer time significantly.

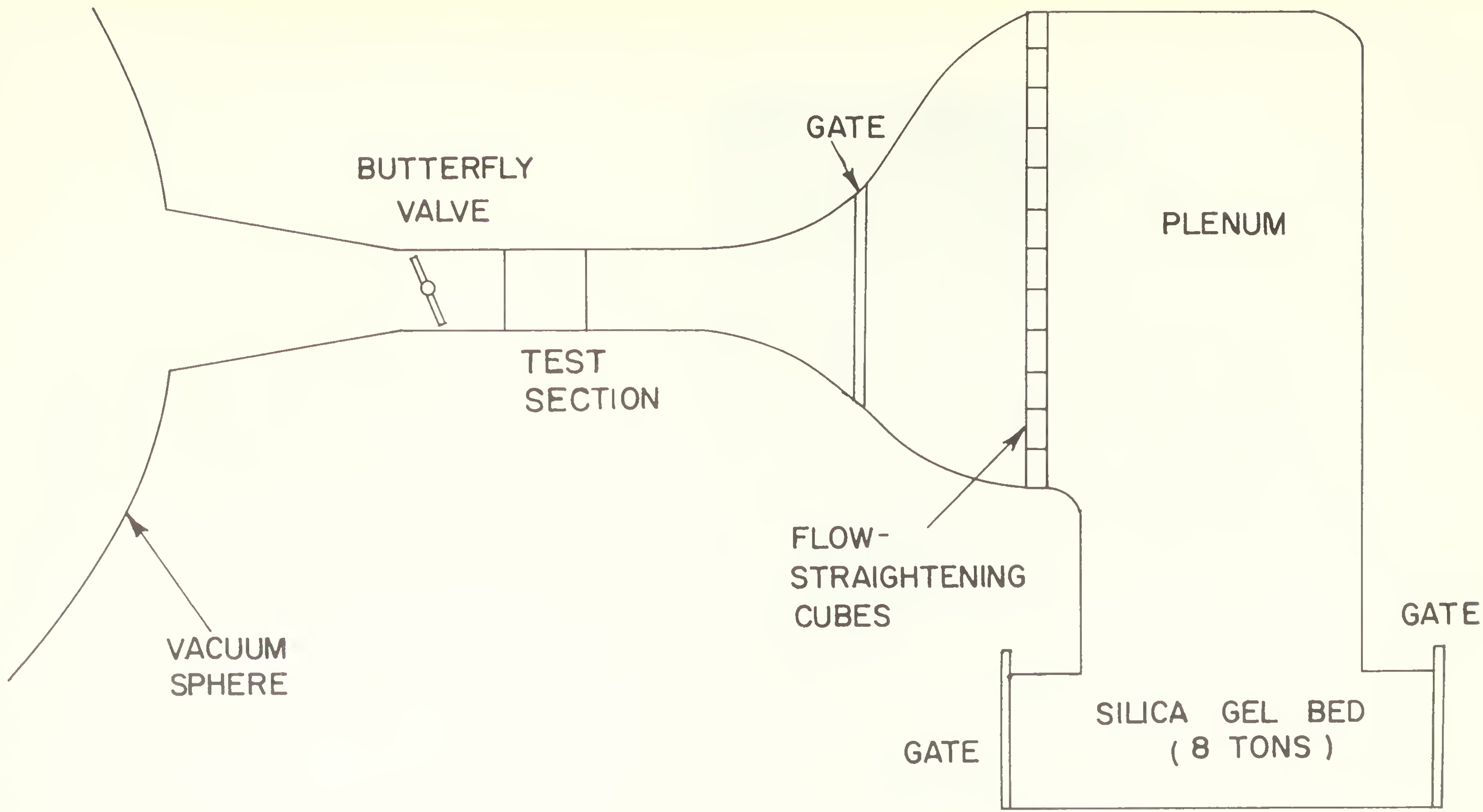


FIGURE 1. FUNCTIONAL SCHEMATIC OF NSRDC WIND TUNNEL

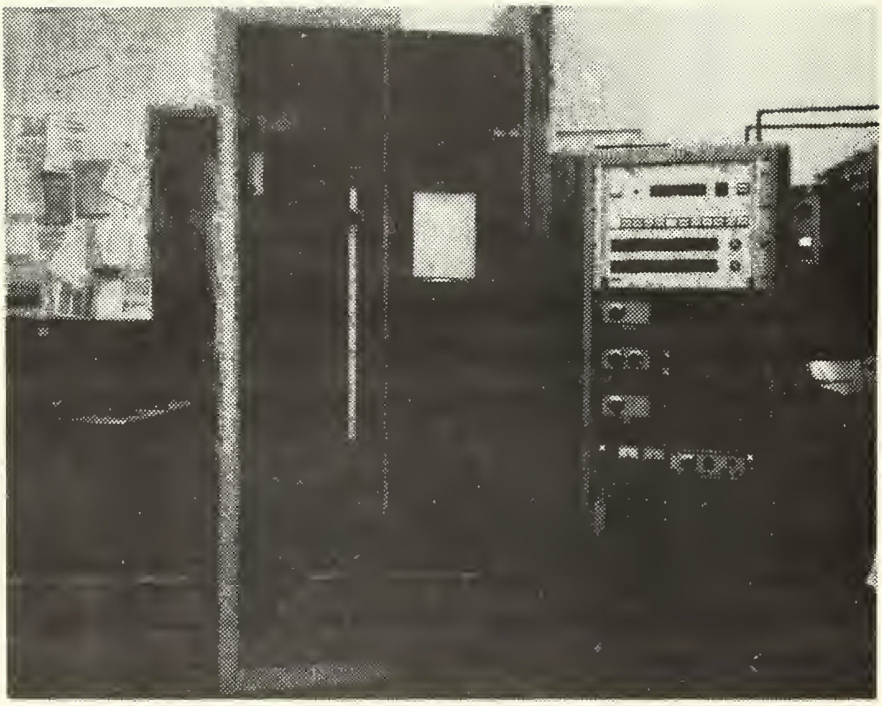


FIGURE 2. BECKMAN 210 DATA RECORDING SYSTEM

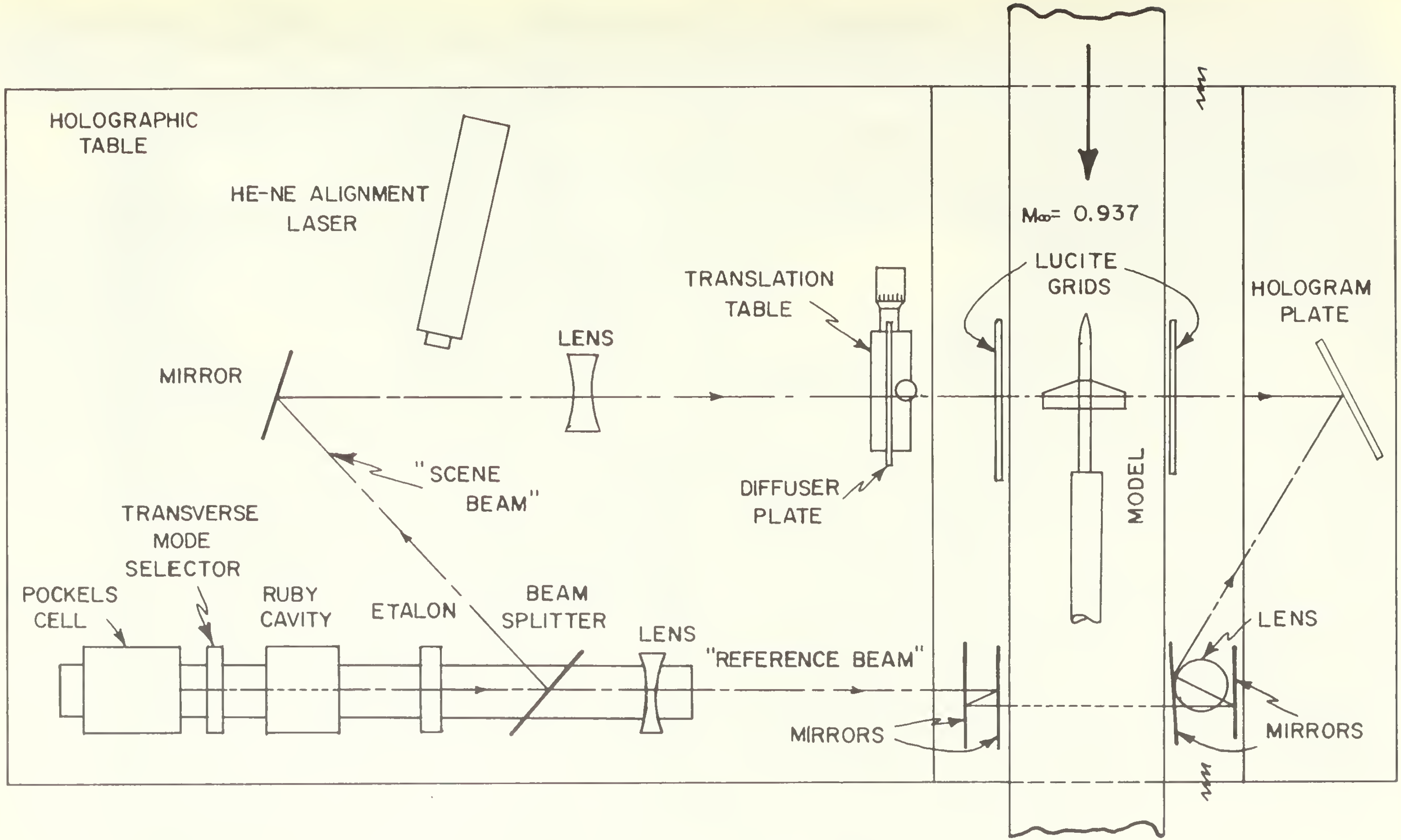


FIGURE 3. SCHEMATIC DRAWING OF THE HOLOGRAPHIC ARRANGEMENT

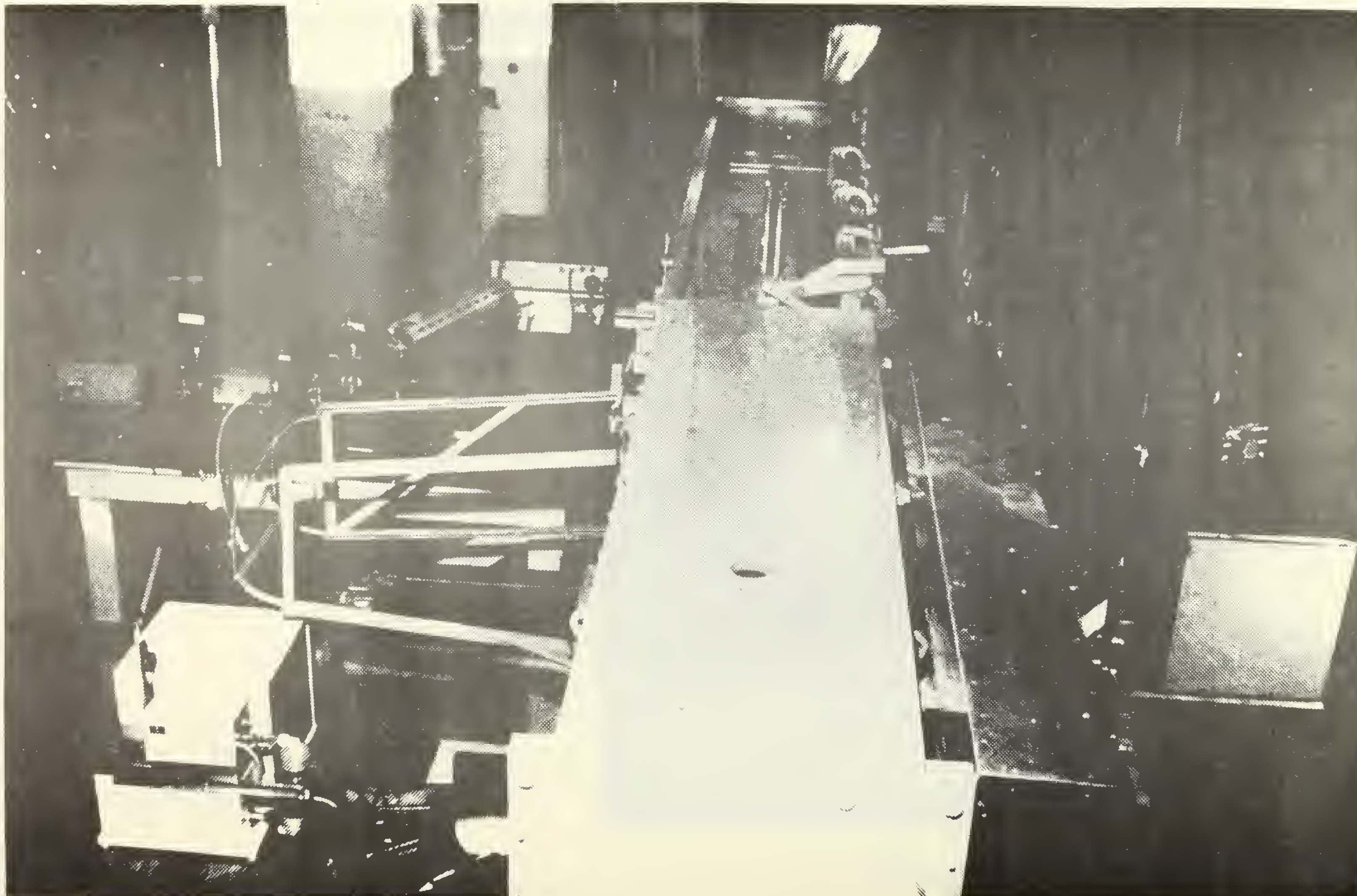


FIGURE 4. OVERHEAD VIEW OF TUNNEL AND ENTIRE HOLOGRAPHIC SYSTEM

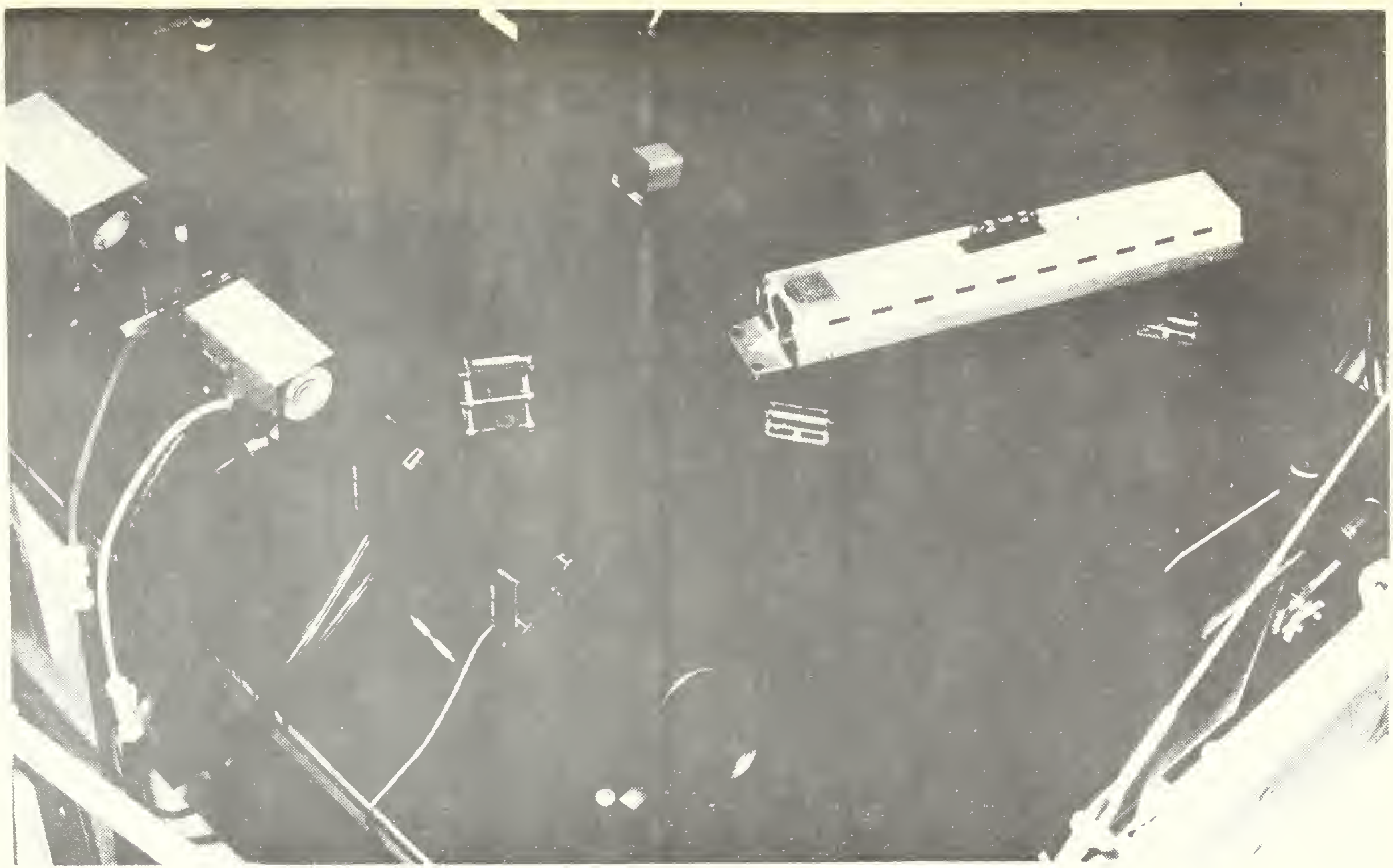


FIGURE 5. OBLIQUE VIEW OF HOLOGRAPHIC TABLE

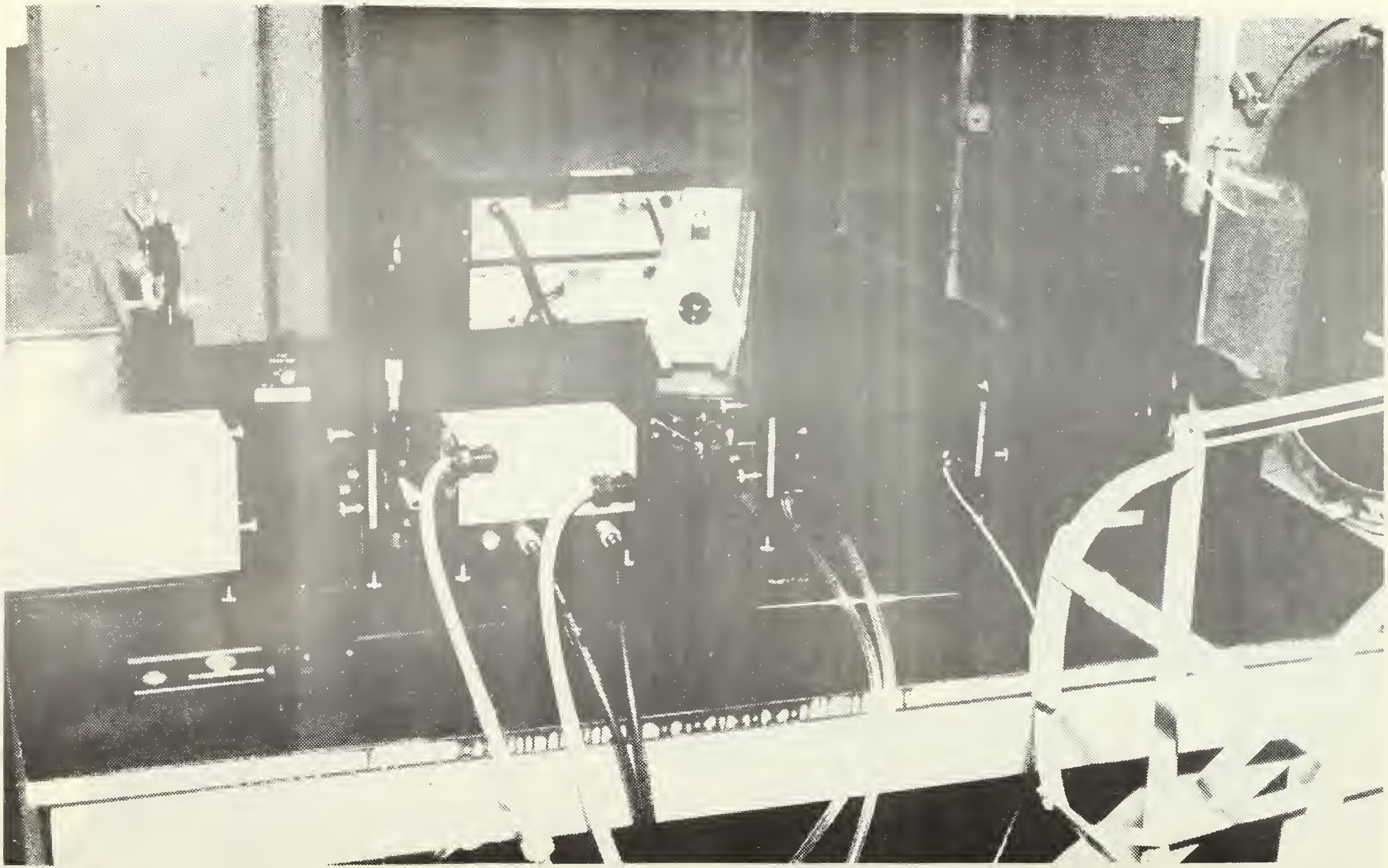


FIGURE 6. OBLIQUE VIEW OF HOLOGRAPHIC TABLE

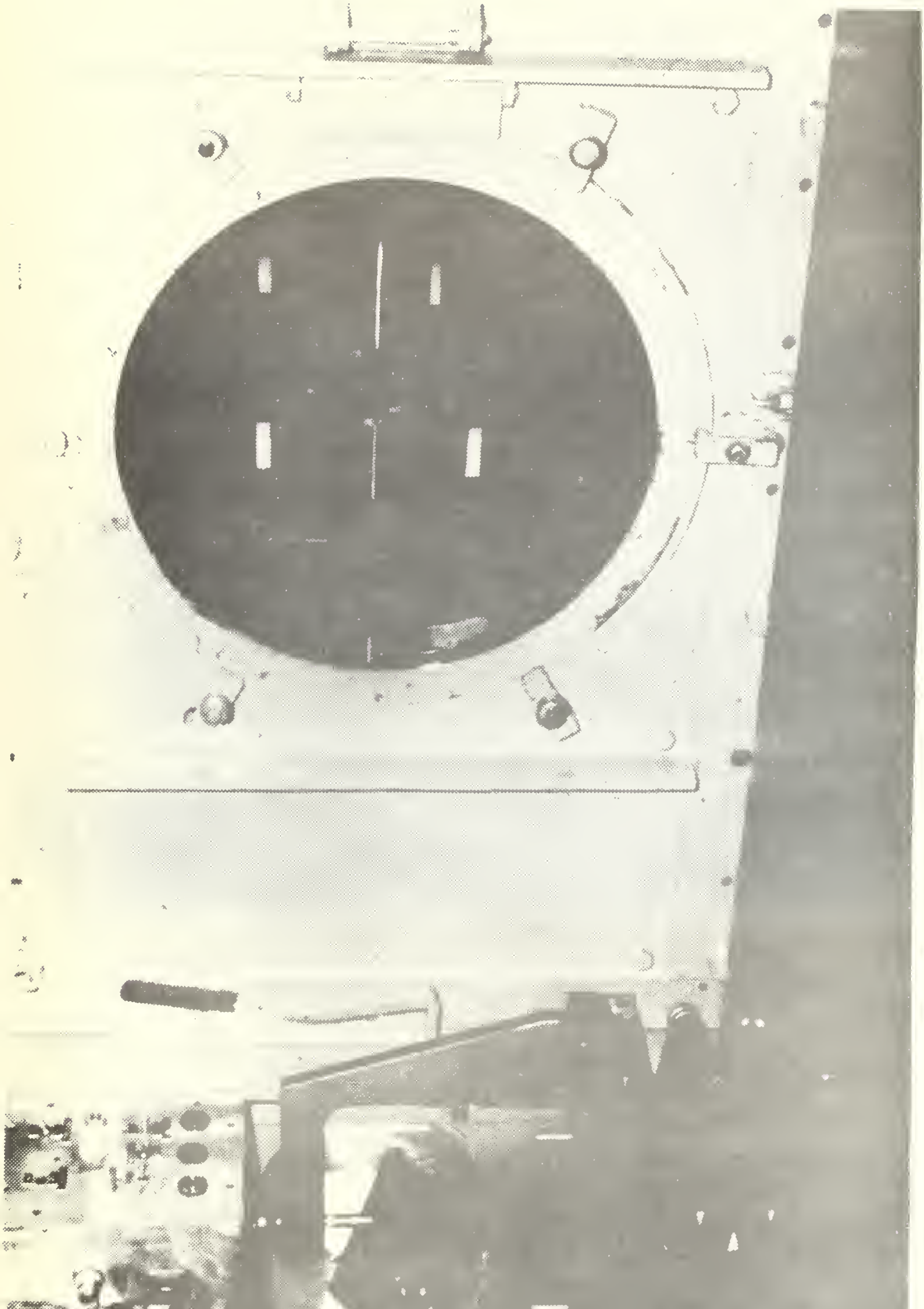


FIGURE 7. DETAIL OF MODEL MOUNTING AND REFERENCE GRIDS

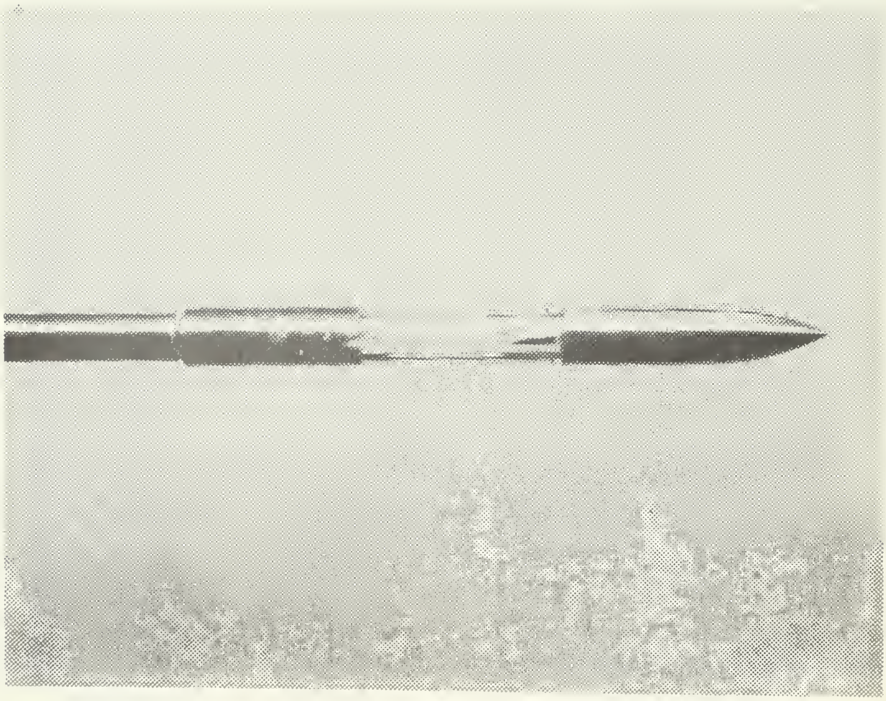


FIGURE 8. AERODYNAMIC TEST MODEL; 0 DEG. ROLL ANGLE

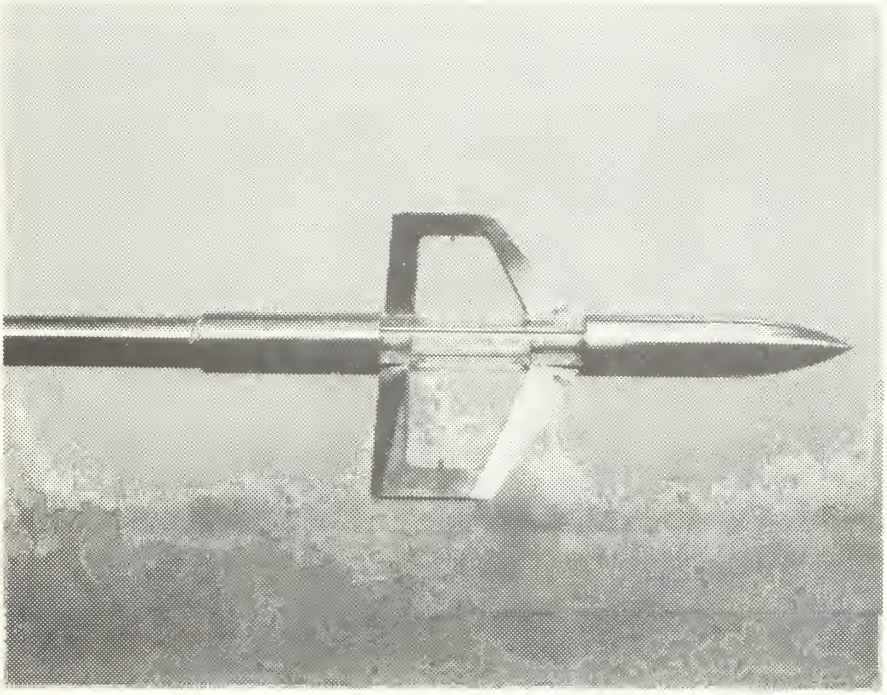


FIGURE 9. AERODYNAMIC TEST MODEL; 45 DEG. ROLL ANGLE

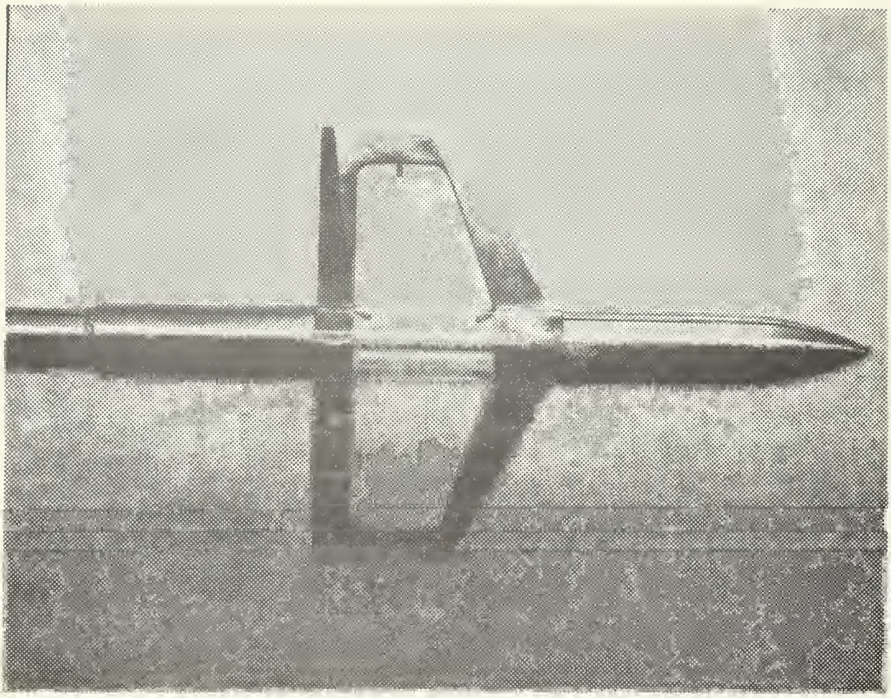


FIGURE 10. AERODYNAMIC TEST MODEL; 90 DEG. ROLL ANGLE

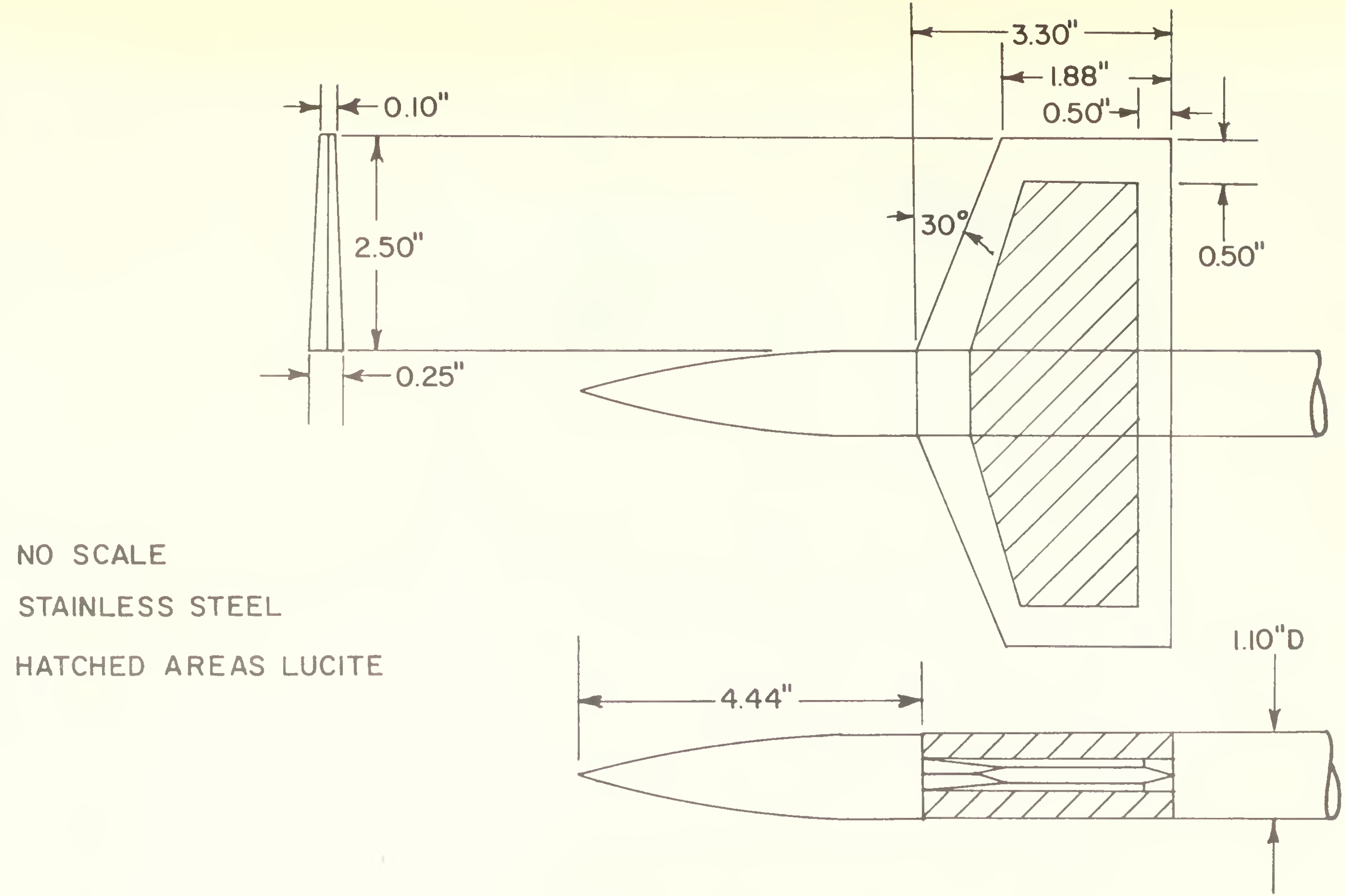


FIGURE 11. DETAILS OF THE AERODYNAMIC TEST MODEL

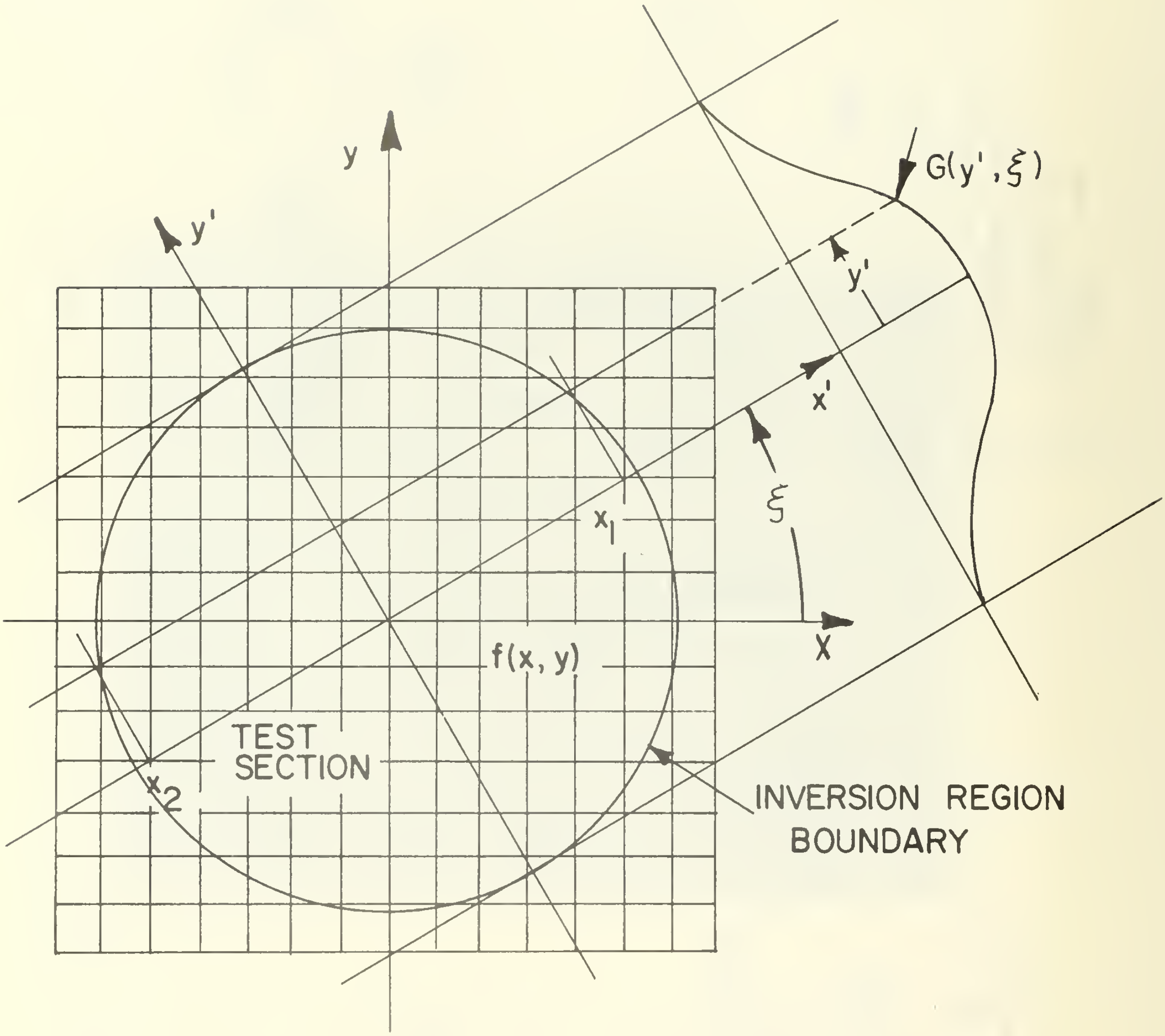


FIGURE 12. CO-ORDINATE SYSTEM USED FOR INVERSION OF FRINGE NUMBER TO DENSITY

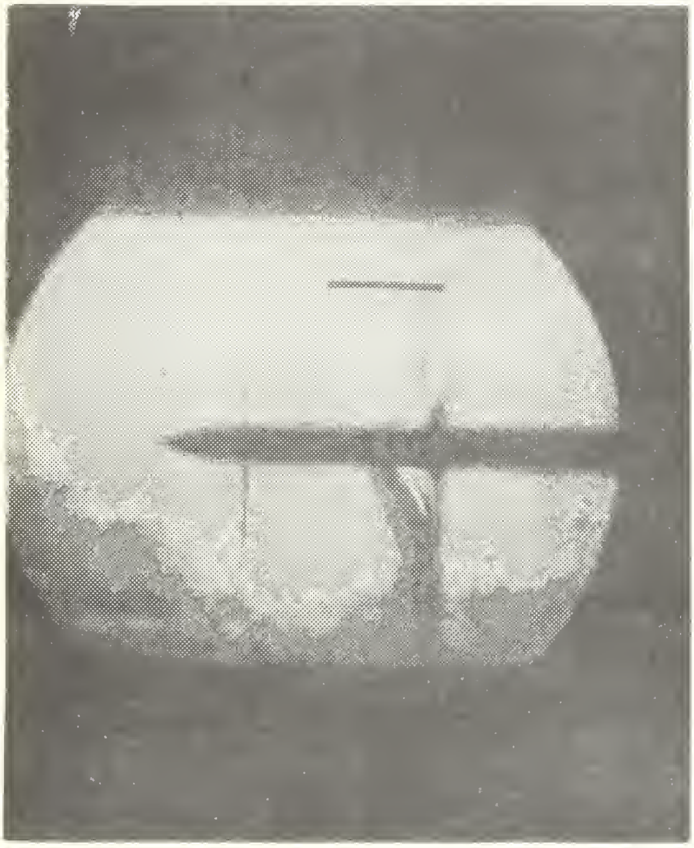


FIGURE 13. SCHLIEREN PHOTOGRAPH; 0 DEG. ROLL ANGLE, 0.967 MACH NUMBER

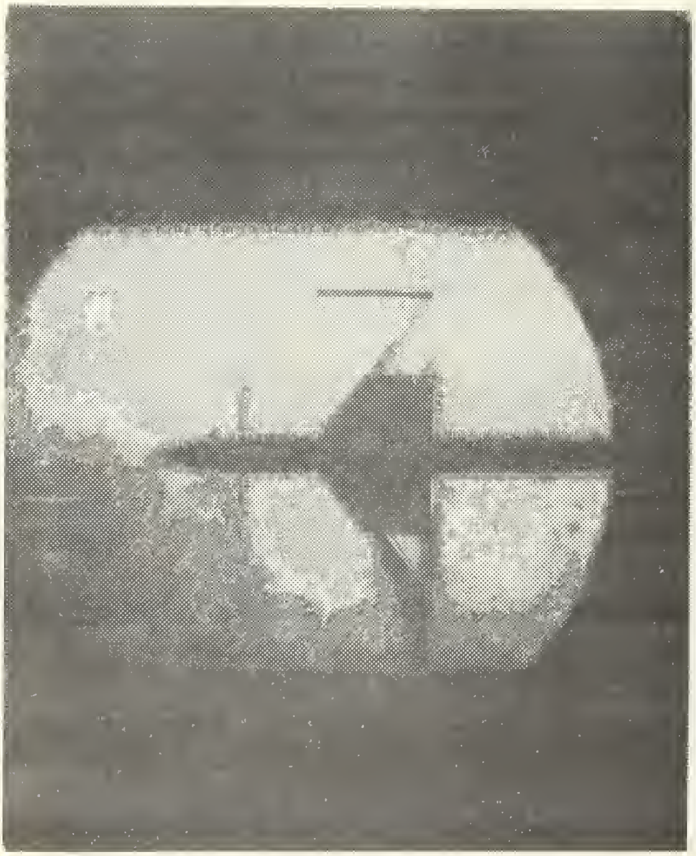


FIGURE 14. SCHLIEREN PHOTOGRAPH; 45 DEG. ROLL ANGLE, 0.967 MACH NUMBER

FIGURE 15. SCHLIEREN PHOTOGRAPH; 90 DEG. ROLL ANGLE, 0.967 MACH NUMBER

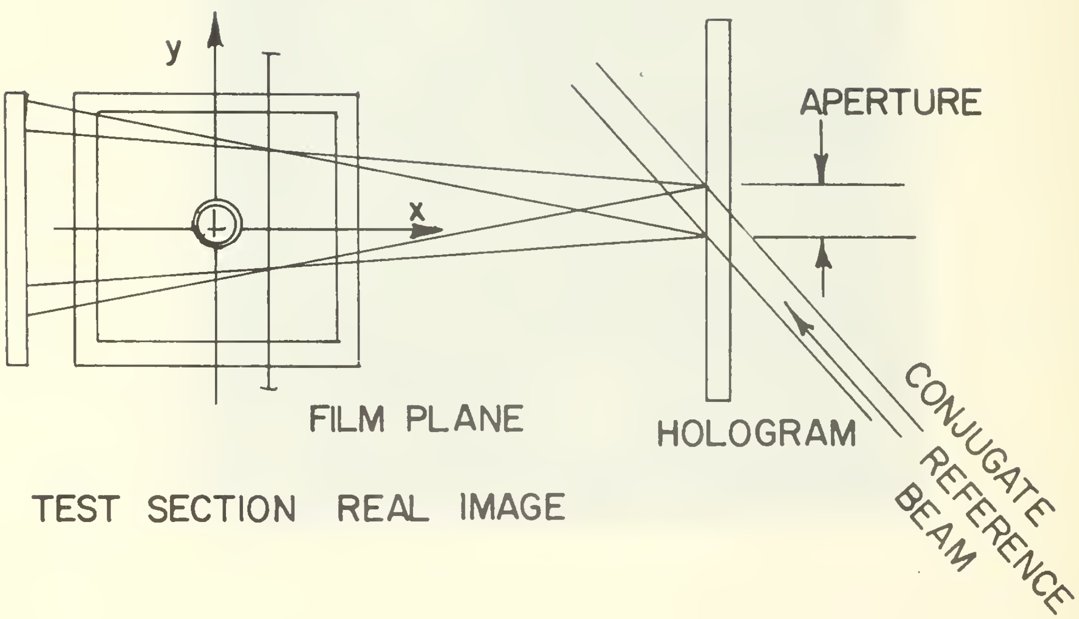


FIGURE 16. LENSLESS PHOTOGRAPHIC TECHNIQUE USING A CONJUGATE REFERENCE BEAM OF SMALL DIAMETER

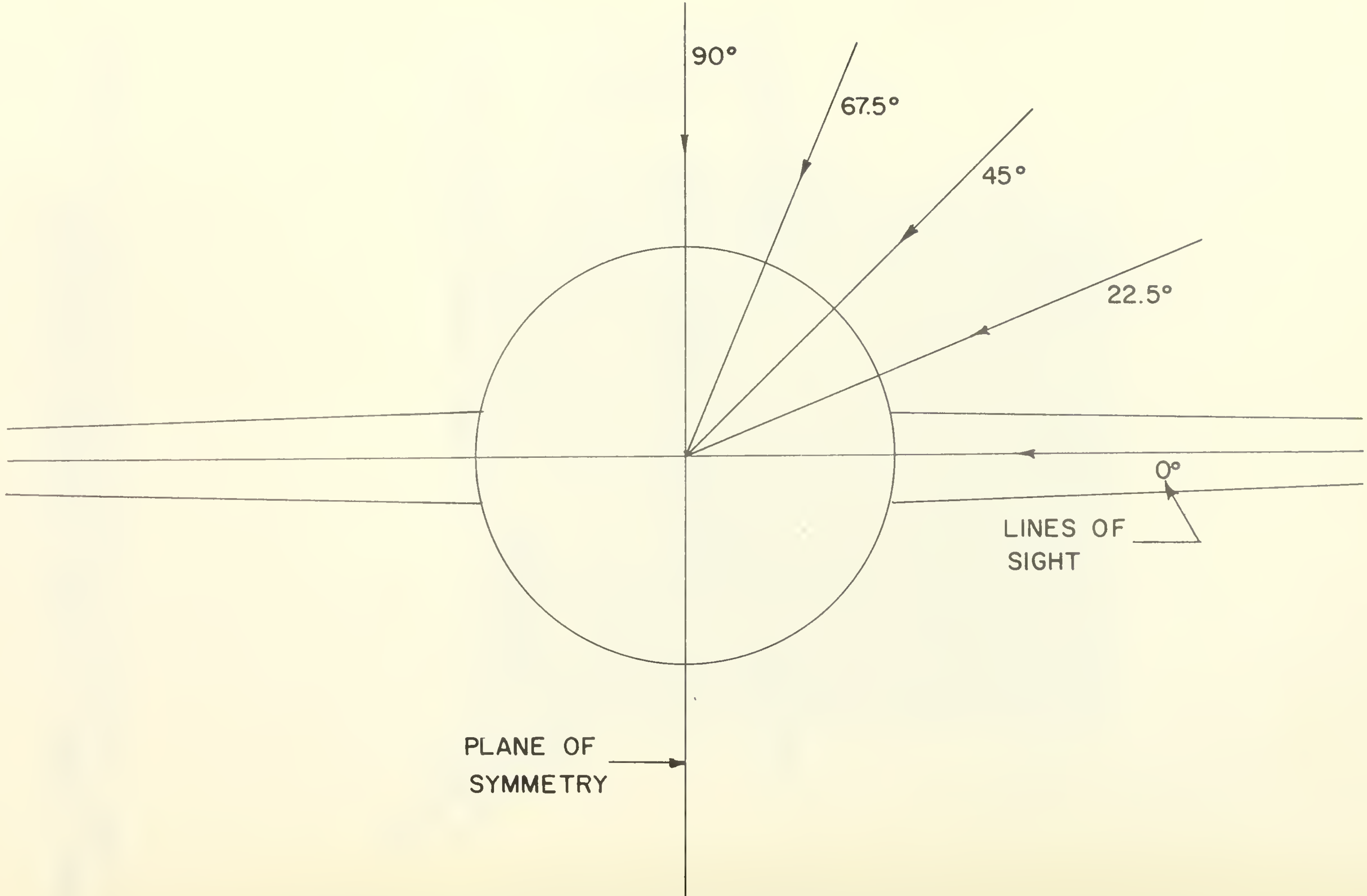


FIGURE 18. FRINGE DATA INPUT INFORMATION

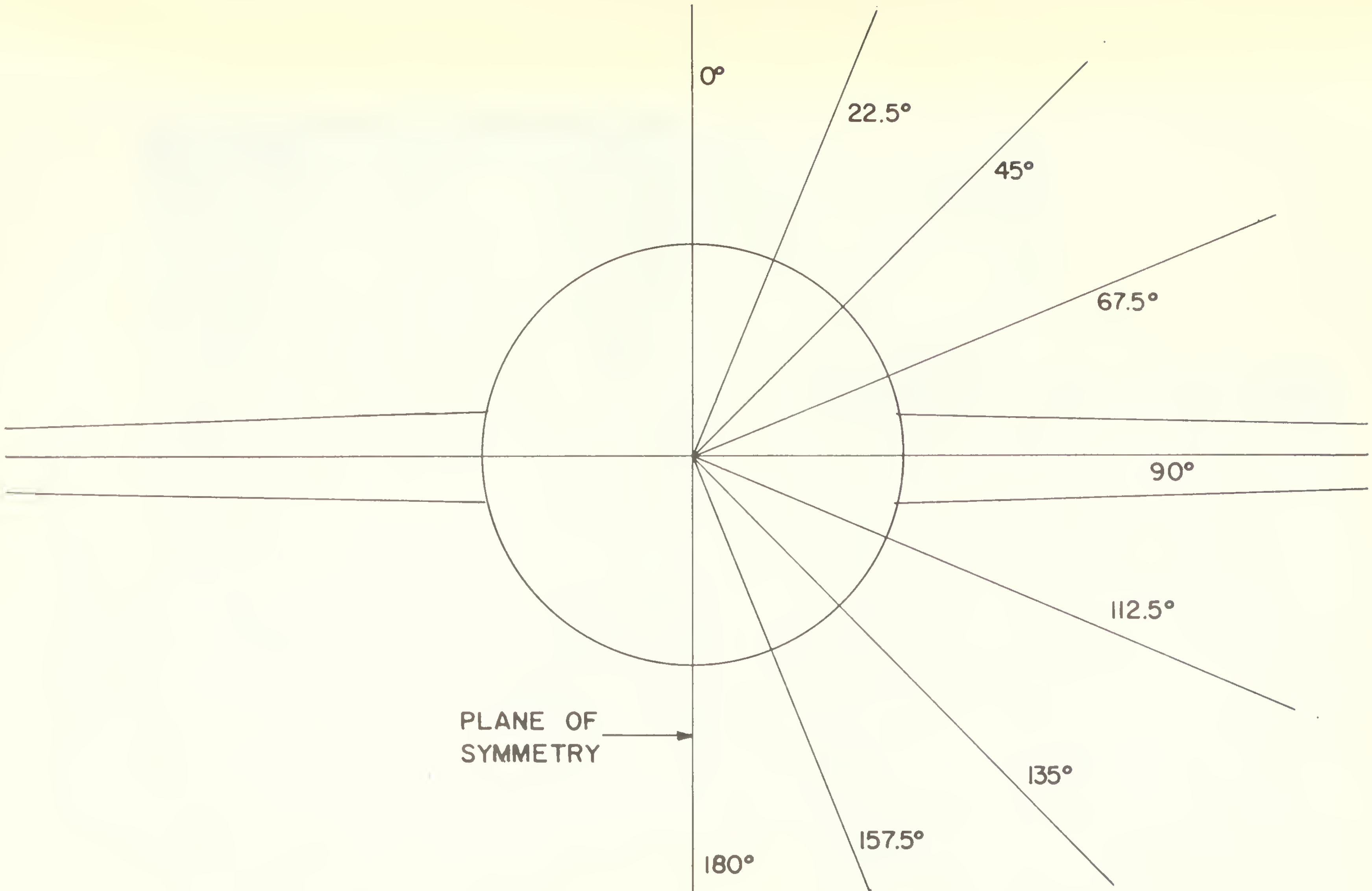
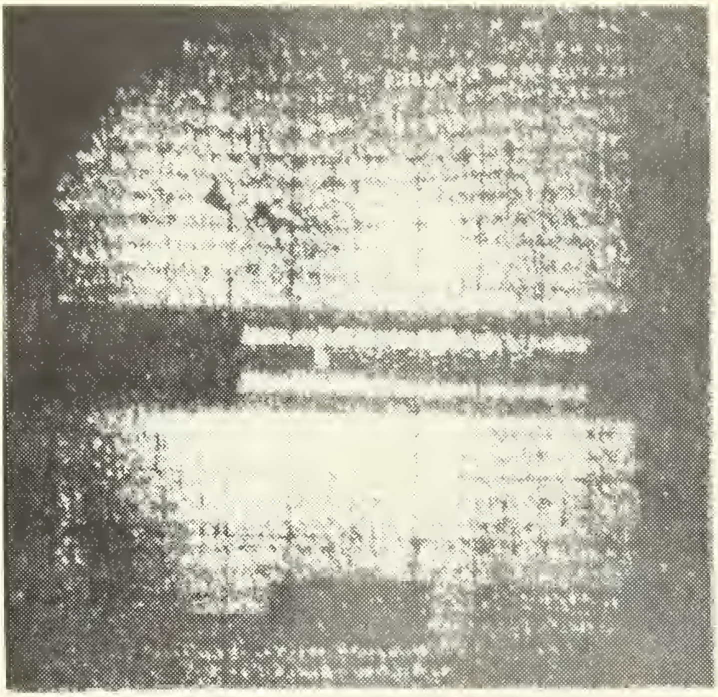


FIGURE 19. DENSITY DATA OUTPUT INFORMATION



DOUBLE-STATIC INTERFEROGRAM

STATIC-DYNAMIC INTERFEROGRAM

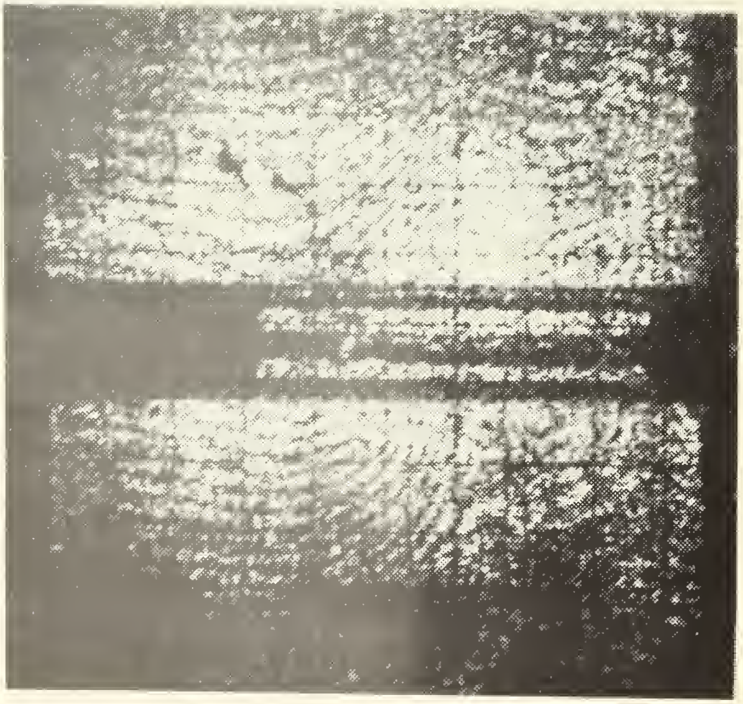


FIGURE 20. PHOTOGRAPHIC INTERFEROGRAMS FOR 0 DEG. VIEWING ANGLE

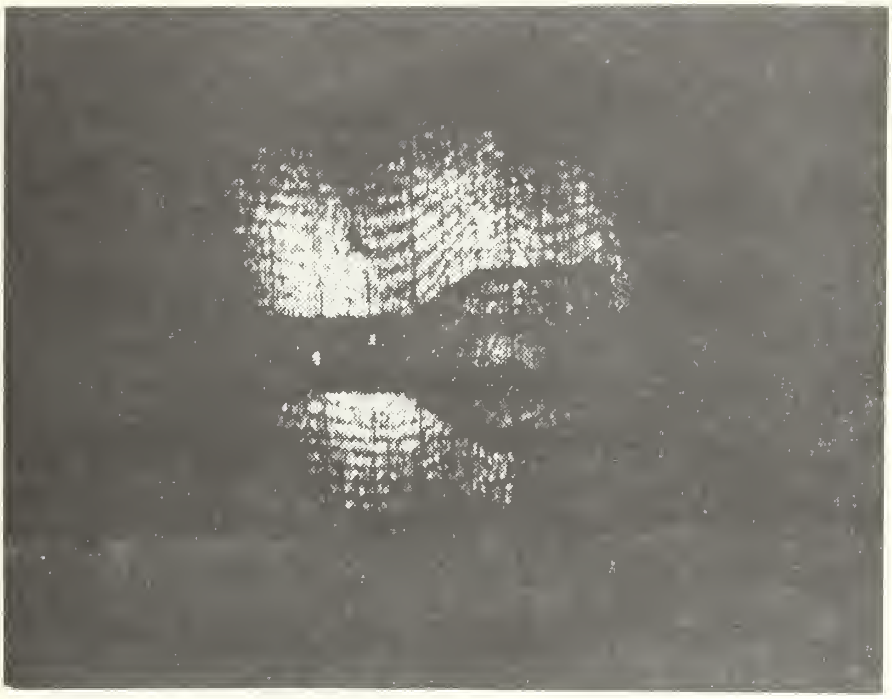


FIGURE 21. STATIC-DYNAMIC INTERFEROGRAM FOR 22½ DEG. VIEWING ANGLE

FIGURE 22. STATIC-DYNAMIC INTERFEROGRAM FOR 45 DEG. VIEWING ANGLE

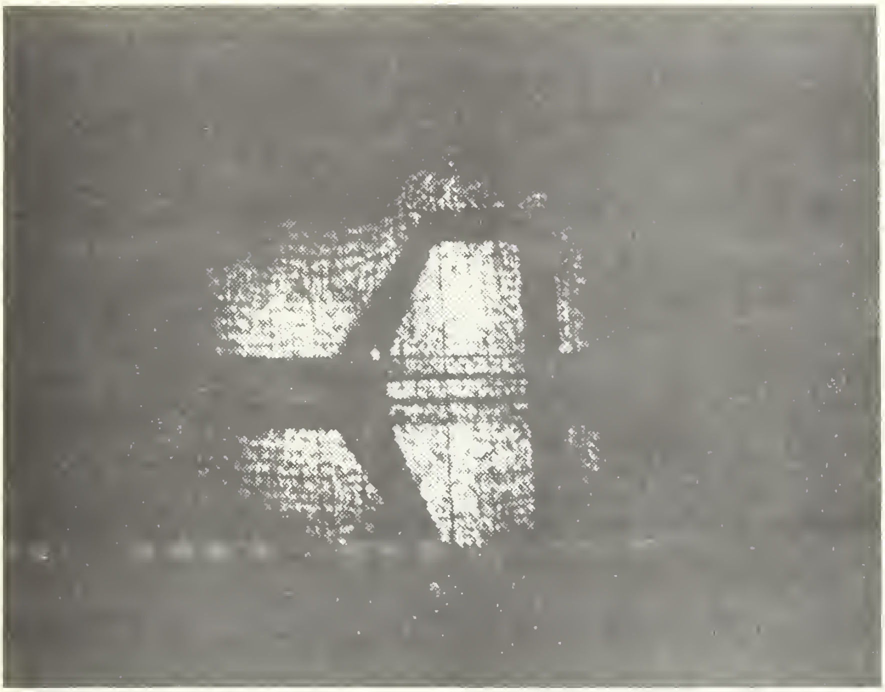


FIGURE 23. STATIC-DYNAMIC INTERFEROGRAM FOR 67½ DEG. VIEWING ANGLE

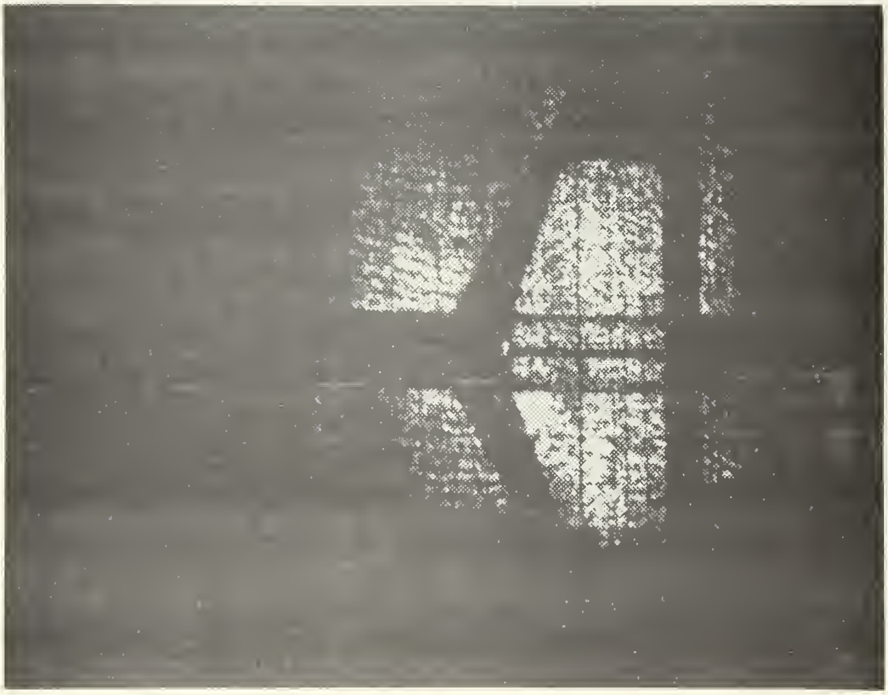


FIGURE 24. STATIC-DYNAMIC INTERFEROGRAM FOR 90 DEG. VIEWING ANGLE

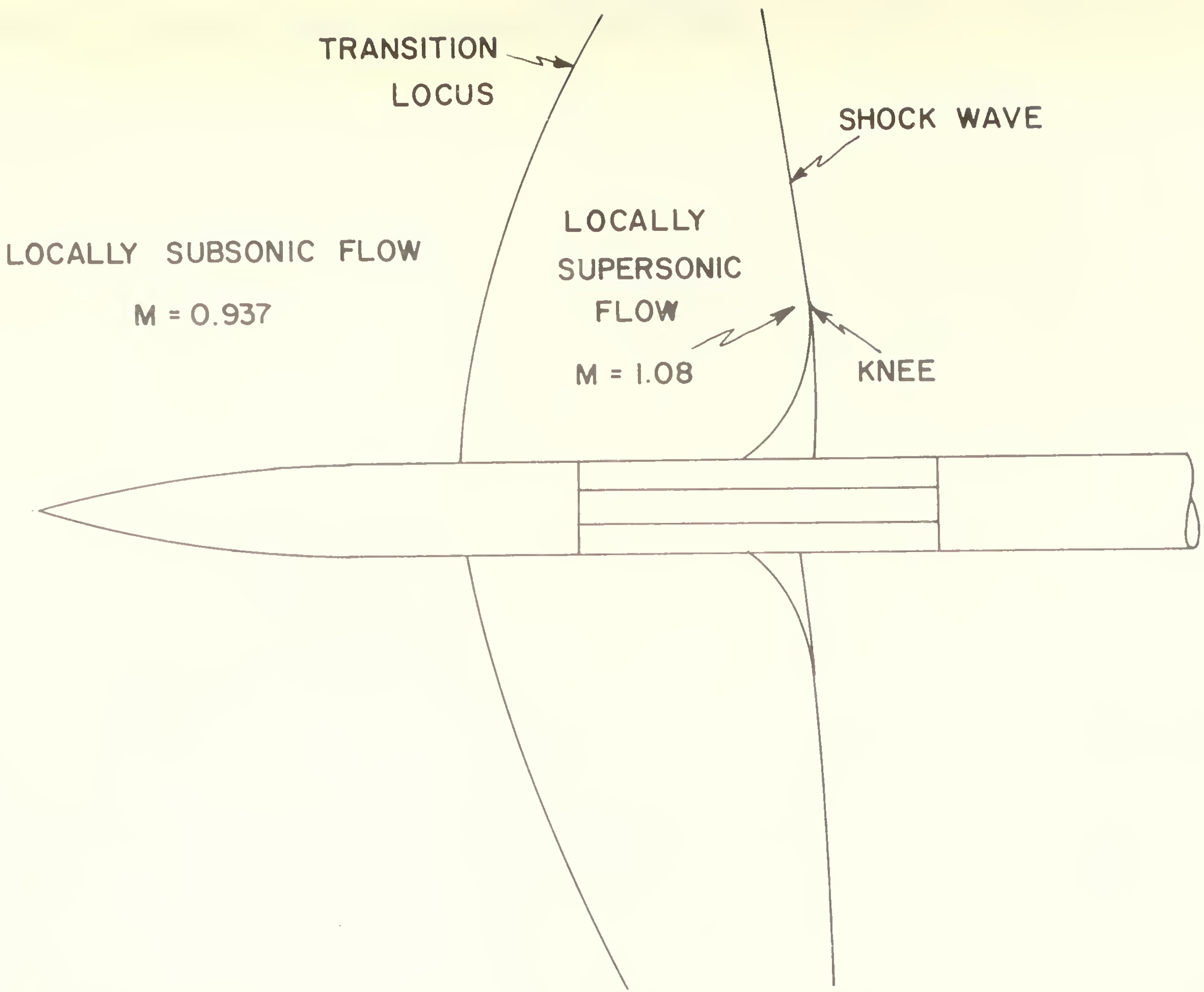


FIGURE 25.
CHARACTERISTIC TRANSONIC FLOW REGIONS; FROM SEVERAL INTERFEROGRAMS

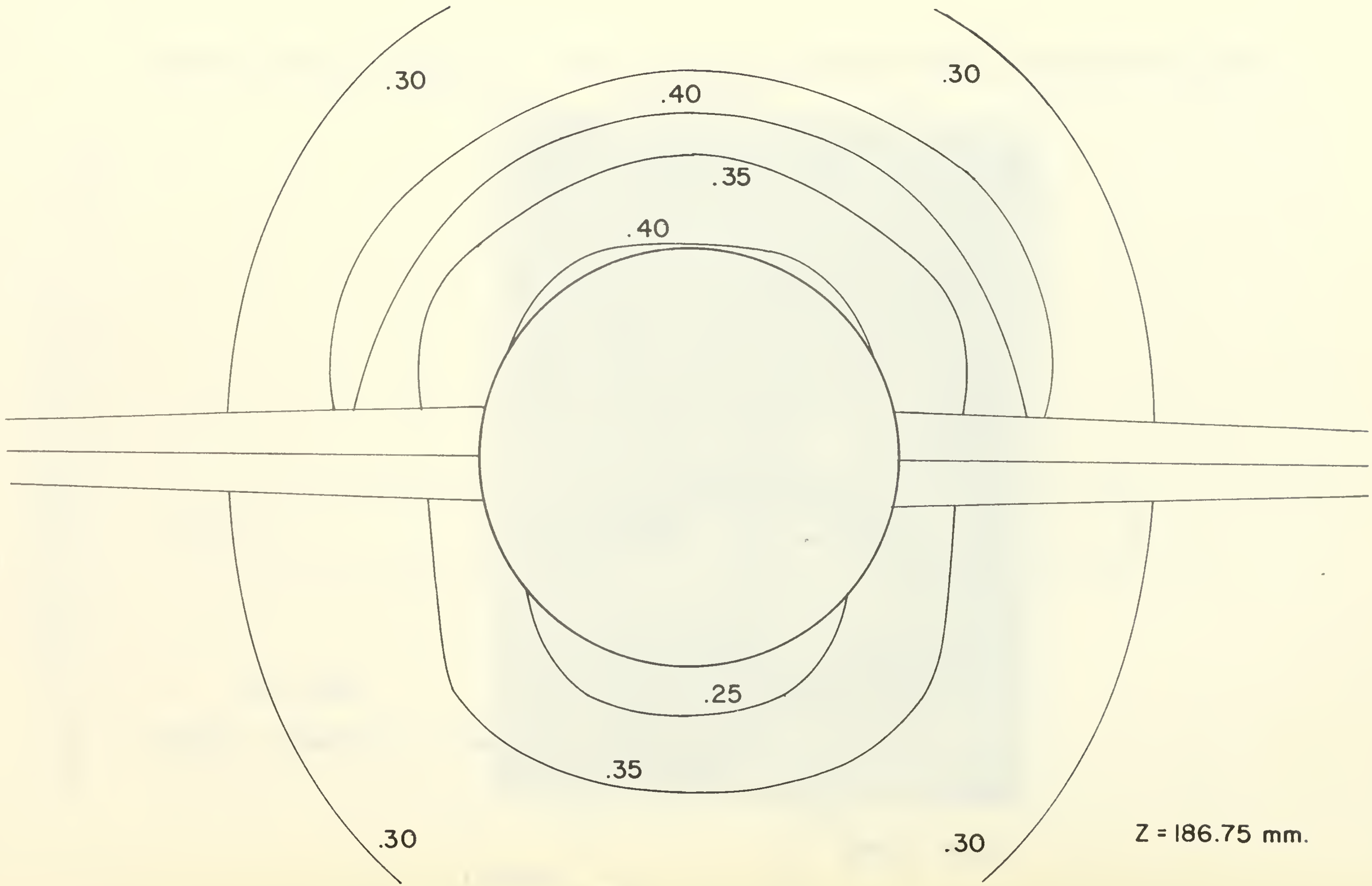


FIGURE 26. CONTOUR PLOT OF DENSITY FUNCTION, $(\rho / \rho_\infty) - 1$, FOR GIVEN CROSS-SECTIONAL PLANE

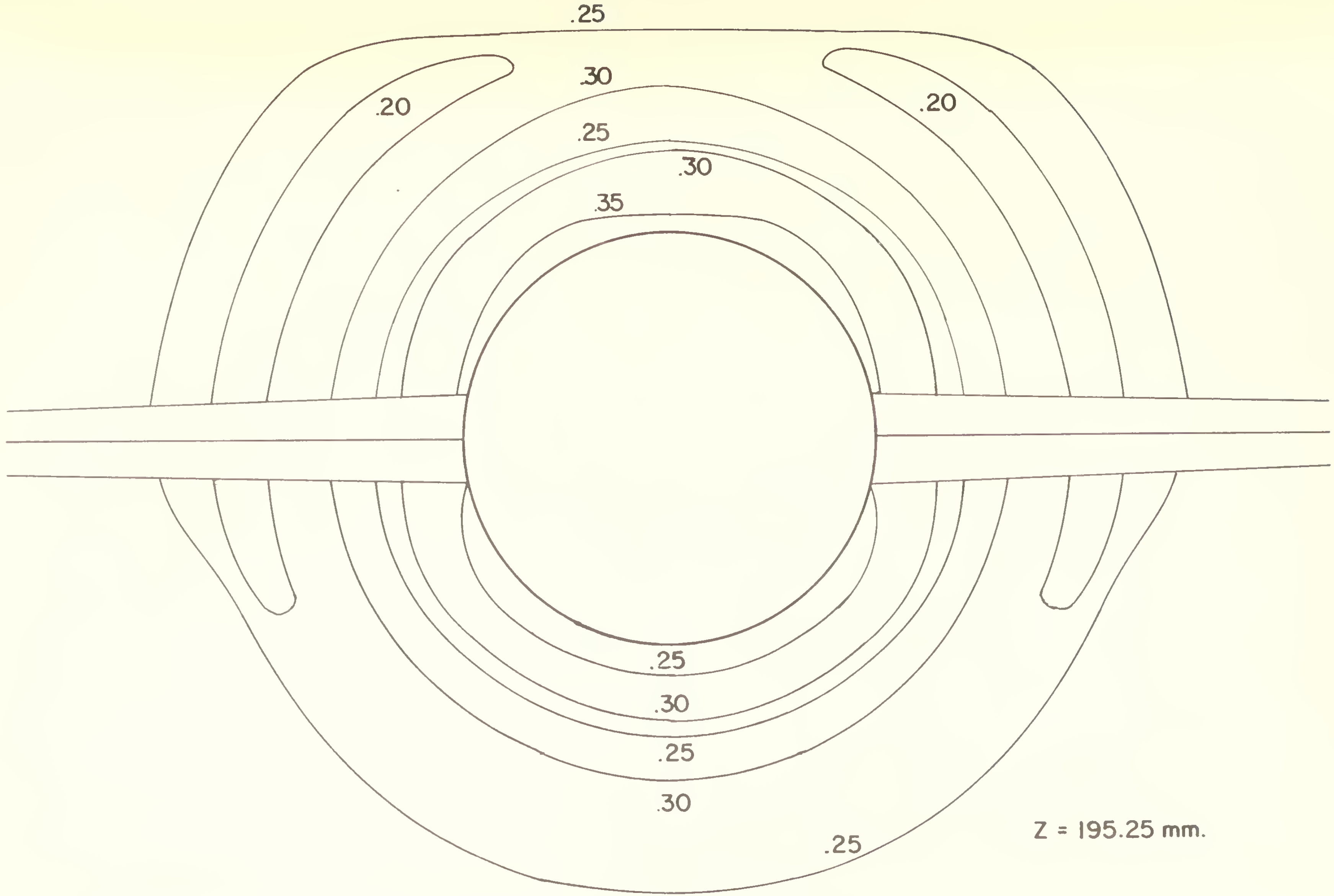


FIGURE 27.
CONTOUR PLOT OF DENSITY FUNCTION, $(\rho / \rho_\infty) - 1$, FOR GIVEN CROSS-SECTIONAL PLANE

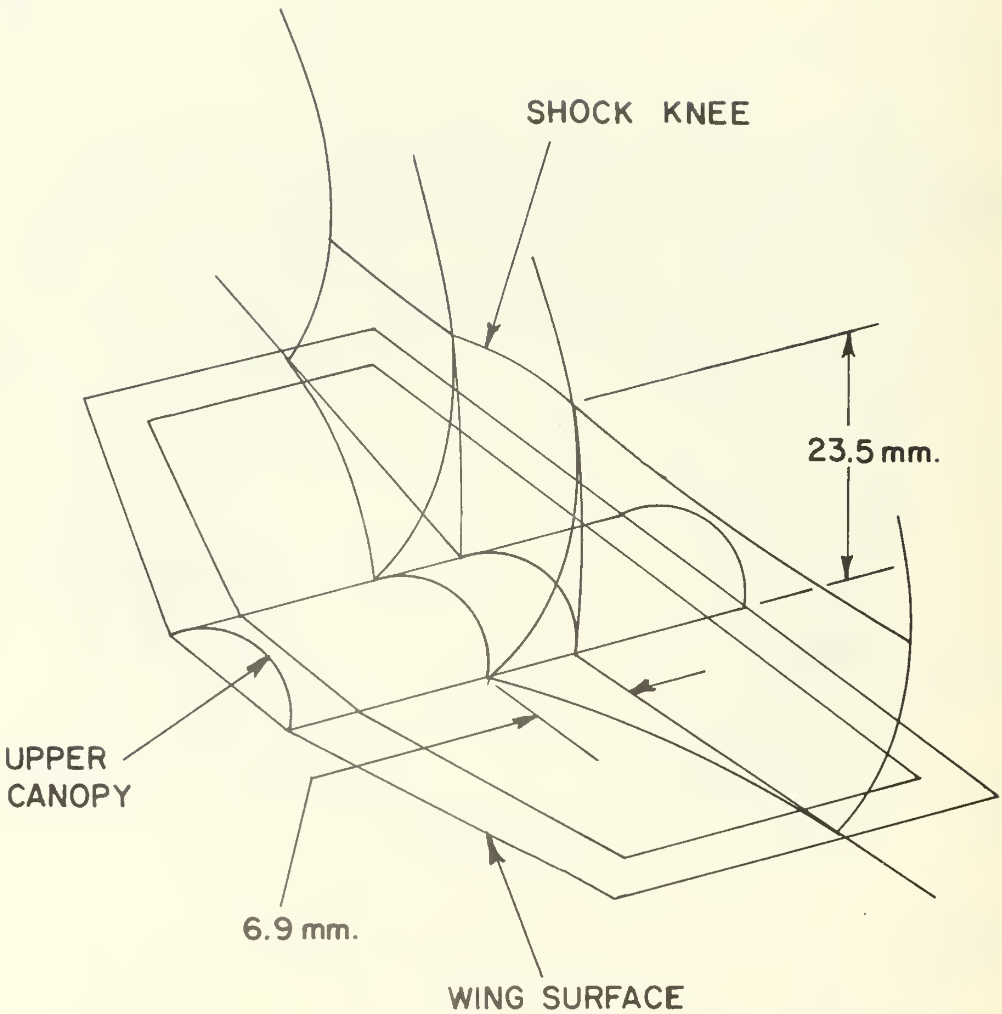


FIGURE 28. THREE DIMENSIONAL SCHEMATIC OF SHOCK WAVE STRUCTURE; CONSTRUCTED USING SEVERAL INTERFEROGRAM VIEWS

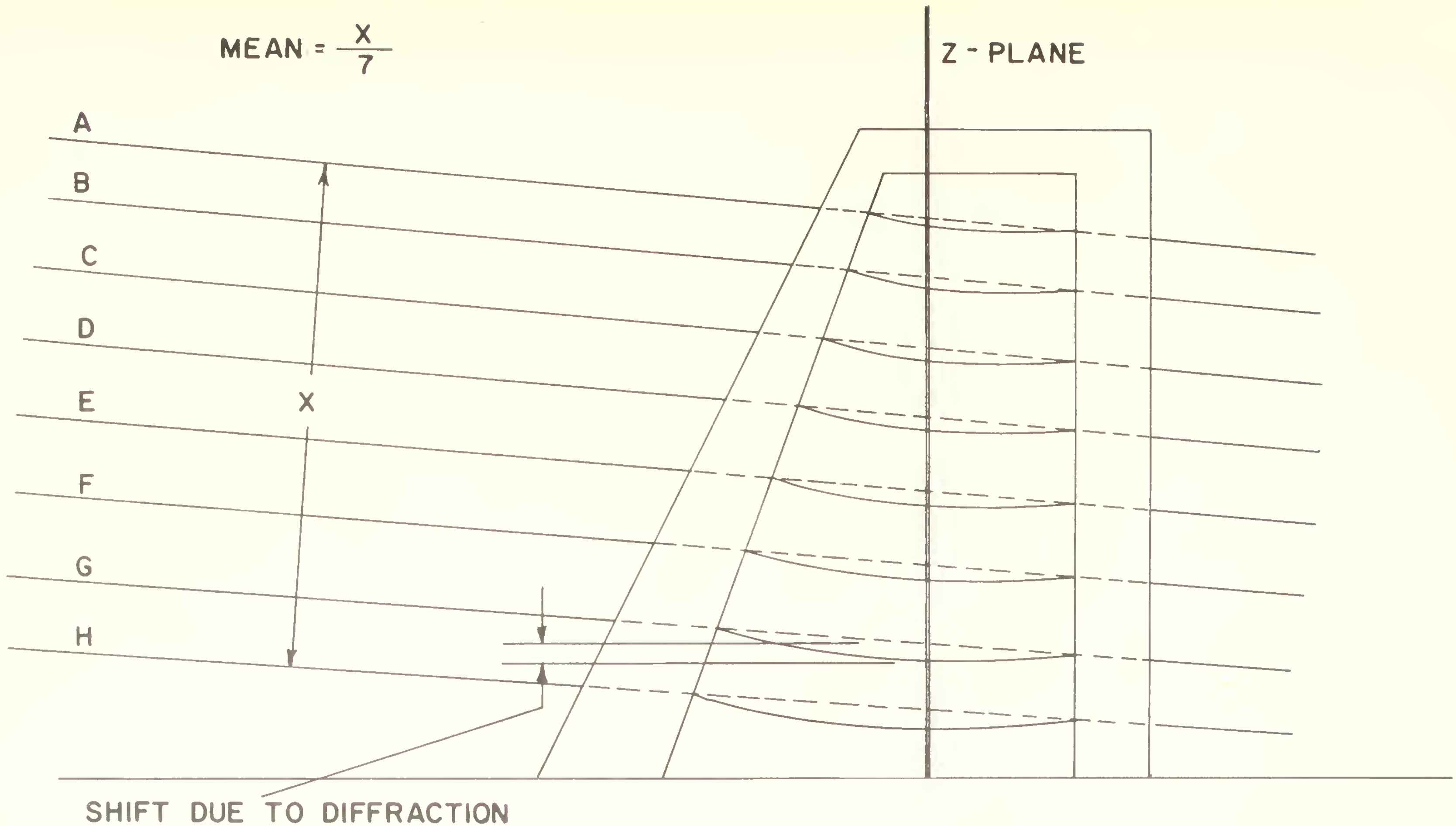


FIGURE 29. DOUBLE-STATIC HOLOGRAM REDUCTION PROCESS; Z = 186.75 mm.

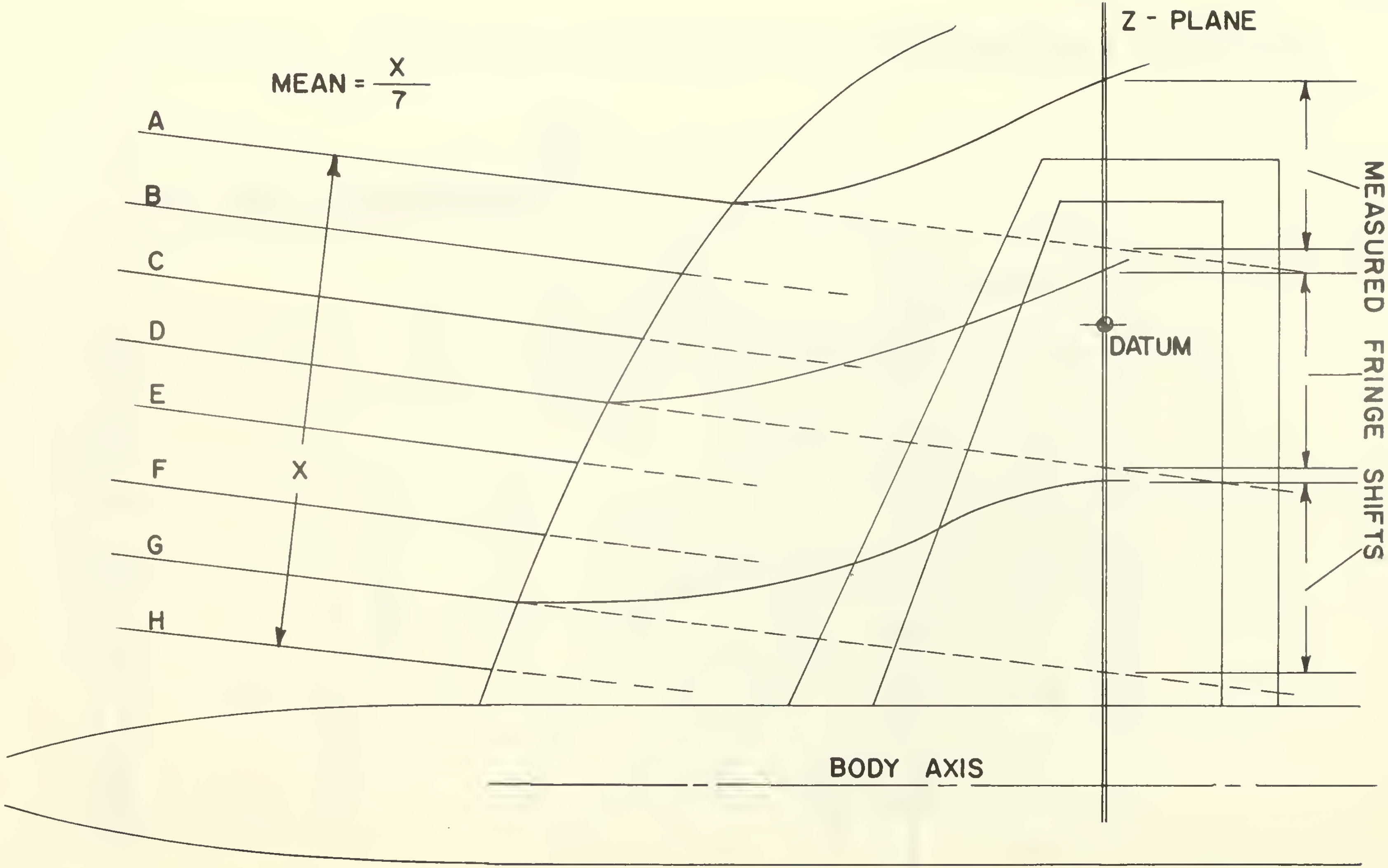


FIGURE 30. STATIC-DYNAMIC HOLOGRAM REDUCTION PROCESS; Z = 186.75 mm.

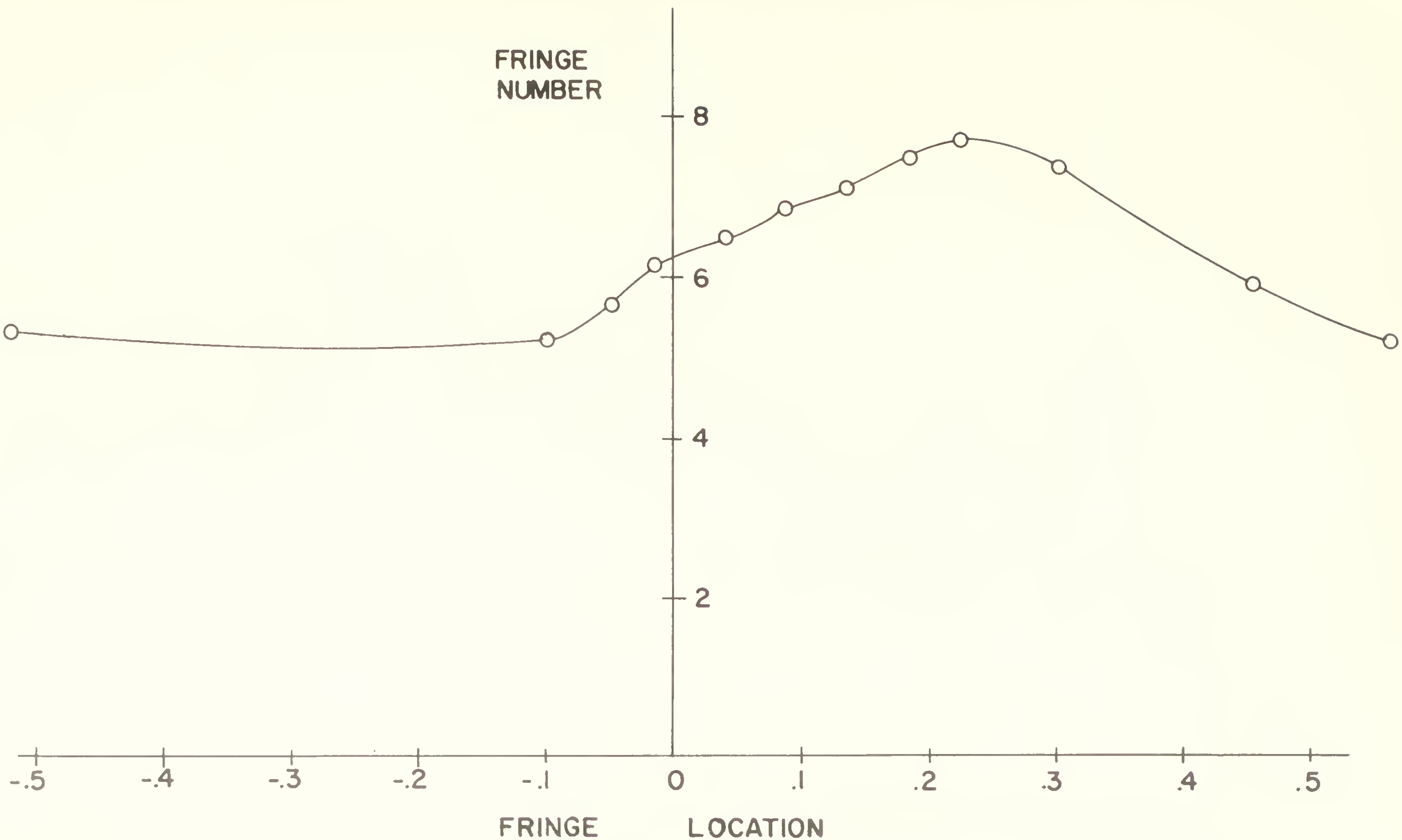


FIGURE 31. RADIAL VARIATION OF FRINGE NUMBER; ZERO DEGREE VIEW, Z = 186.75 mm.

RUN NUMBER	HOLOGRAM NUMBER	DOUBLE EXPOSURE	ROLL ANGLE	P_A (psi)	P_T (psi)	T_T (deg. F)	MACH NUMBER
1	9	S/S	0.00				
2	10	S/D	0.00	14.781	14.181	64.6	.9371
3	23	S/S	11.25				
4	24	S/D	11.25	14.820	14.228	64.2	.9398
5	19	S/S	22.50				
6	20	S/D	22.50	14.784	14.187	64.6	.9361
7	15	S/S	33.75				
8	16	S/D	33.75	14.800	14.206	64.2	.9392
9	11	S/S	45.00				
10	12	S/D	45.00	14.785	14.185	64.6	.9391
11	25	S/S	56.25				
12	26	S/D	56.25	14.830	14.237	64.3	.9361
13	21	S/S	67.50				
14	22	S/D	67.50	14.787	14.195	64.3	.9365
15	17	S/S	78.75				
16	18	S/D	78.75	14.870	14.285	64.8	.9342
17	13	S/S	90.00				
18	14	S/D	90.00	14.787	14.195	64.6	.9402
19	35	S/S	101.25				
20	36	S/D	101.25	14.781	14.274	61.5	.9392
21	27	S/S	112.50				
22	28	S/D	112.50	14.875	14.281	62.5	.9371
23	37	S/S	123.75				
24	38	S/D	123.75	14.878	14.284	61.0	.9369
25	29	S/S	135.00				
26	30	S/D	135.00	14.875	14.278	62.0	.9372
27	39	S/S	146.25				
28	40	S/D	146.25	14.881	14.286	60.3	.9352
29	31	S/S	157.50				
30	32	S/D	157.50	14.875	14.276	61.8	.9350
31	41	S/S	168.75				
32	42	S/D	168.75	14.881	14.287	61.0	.9371
33	33	S/S	180.00				
34	34	S/D	180.00	14.875	14.282	62.0	.9357

TABLE I. EXPERIMENTAL DATA FOR WIND TUNNEL RUNS AT NSRDC

FRINGE I. D.	MEASURED SHIFT	CORRECTED SHIFT	FRINGE NUMBER	DISTANCE FROM DATUM	DEMAG. DISTANCE	DISTANCE FROM AXIS	NONDIMEN. LOCATION
A	25.0	21.475	3.53	-27.0	-25.1	-63.8	-.798
B	29.6	26.075	4.29	-23.2	-21.6	-59.8	-.748
C	30.4	26.875	4.42	-18.5	-17.2	-55.4	-.693
D	31.6	28.075	4.62	-13.0	-12.1	-50.3	-.623
E	35.5	31.975	5.23	- 9.5	- 8.8	-47.0	-.588
F	36.0	32.475	5.34	- 3.6	- 3.4	-41.6	-.520
G	34.0	30.475	5.23	+32.8	+30.5	- 7.7	-.096
H	36.4	32.875	5.64	+37.0	+34.4	- 3.8	-.048
I	40.0	36.475	6.26	+40.0	+37.2	- 1.0	-.013
J	41.5	37.975	6.51	+44.4	+41.3	+ 3.1	+.039
K	43.5	39.975	6.86	+48.5	+45.2	+ 7.0	+.088
L	45.3	41.775	7.17	+52.7	+49.1	+10.9	+.136
M	47.4	43.875	7.53	+57.0	+53.1	+14.9	+.186
N	48.6	45.075	7.73	+60.5	+56.3	+18.1	+.226
O	46.8	43.275	7.42	+67.0	+62.4	+24.2	+.303
P	38.5	34.975	5.99	+80.3	+74.8	+36.6	+.458
Q	34.0	30.475	5.23	+89.6	+83.4	+45.2	+.565

Mean free stream spacing = 5.96 mm. Diffraction Correction = 3.525 mm.

Magnification factor = 1.074 Body axis location = +38.2 mm

All measurements in millimeters - = above, + = below

TABLE II.
RECORDED DATA FOR ZERO DEGREE VIEW AT Z = 186.75 mm. FROM BODY NOSE

APPENDIX A

REDUCTION OF AN INTERFEROGRAM TO OBTAIN FRINGE SHIFT DATA

The reduction of data for a cross-sectional plane of interest involved a complete analysis of both doubly exposed holograms, and their corresponding interferograms, for each viewing angle.

For each view, or line of sight, the double-static exposure was first placed face down on a light table. The average fringe spacing was recorded on the back by measuring the perpendicular distance between two sufficiently separated fringe lines in the free stream and dividing by the number of spacings in between. The method is portrayed in Figure 29. Of primary importance was the measurement of the uniform shift of fringe lines due to the beam diffractive quality of the lucite sections of the model; this was taken to be the average distance between hypothetically unaltered fringe lines and the corresponding shifted fringes at their intersection with the z-plane.

The static-dynamic exposure was then placed face down on the light table and the fringe and model contours traced on the back, as shown in Figure 30. Again, the average free stream fringe line spacing was measured and checked against the value from the double-static exposure. A reference point, or datum, was selected at the intersection of a well-defined horizontal grid line and the z-plane of interest. Fringe shifts were computed in the following manner:

1. The distance from datum to a hypothetically undeviated fringe line at its intersection with the z-plane was measured in millimeters.
2. The distance from datum to the actual deviated, or shifted, fringe at its intersection with the z-plane was measured in millimeters. Measured shifts should be made perpendicular to the fringe direction. The present procedure is convenient and only introduces an error of about 1% in the overall level of the density field.
3. The raw fringe shift distance was then the distance in 1. above minus the distance in 2.
4. To correct for diffraction error, the uniform shift measured in the double-static exposure was then subtracted from the distance in 3.

Fringe numbers were calculated by dividing the shift for each fringe by the average free stream spacing for the exposure. Magnification factors were computed for each exposure by comparing photographically recorded model diameter with actual model diameter.

Since the datum location varied slightly from exposure to exposure, it was necessary to reference all measurements to a central point for each plane of analysis. This point was taken to be the intersection of the longitudinal axis of the model fuselage and the cross-sectional plane of interest. The intersections of shifted fringe lines with the z-plane were referenced to this fixed point and demagnified. The resulting distances were then nondimensionalized using the selected inversion circle radius. Table II contains typical data recorded for the zero degree view at the 186.75 mm. plane of interest.

A plot of fringe numbers versus nondimensionalized fringe location was produced for each viewing angle using data obtained from the interferograms. Fringe numbers at 201 equidistant points, as required for input into Mode 3 of the inversion computer program, were recorded from the resulting curves. The curve plotted using the data from Table II is shown in Figure 31.

APPENDIX B

APPLICATION OF COMPUTER PROGRAM "HOLOFER"

The computer program used in this study is designed to invert an array of fringe numbers across a field to obtain the associated density field using a special adaptation of the inversion technique first proposed by C. D. Maldonado [5, 6]. Three different modes of operation are available to the operator, as described below:

(a) Mode 1

This mode provides the basic self-testing capability of the computer program. It can either generate its own input density field using Subroutine FUNCT or read in a density field through Subroutine FREAD. The fringe number array corresponding to the input density data is first generated; this information is then used as the input for the inversion to obtain the original density distribution once again. This self-testing procedure was utilized in the present investigation to determine the best value of the scale factor, α , required to assure accurate density in the region of inversion.

(b) Mode 2

This mode generates the fringe array at regular intervals across the test field from irregularly spaced fringe values read in through Subroutine SHEET. Simulated fringe arrays may be generated by one of the functions specified in Subroutine FUNCT if NCODE = 1 is specified.. Mode 2 has not been utilized in the present investigation.

(c) Mode 3

Mode 3 operation reads in a regularly spaced array of input density values directly utilizing Subroutine READ, which is first called by Subroutine GARRAY. The parameters in the first two cards preceding the input fringe data serve to identify the size, location and symmetry of the fringe field. The following parameters were used in calculating the asymmetric density field in the present experiment:

PARAMETER	INPUT
NOF	Run Number
IMAX	201
JMAX	20
ISYM	1
JSYM	1
IMS	201
JMS	5
Z	0.0, 1.0
XO	0.0
YO	0.0
PHISYM	0.0

Amplifying details and applications of the inversion computer program are found in References [3, 7, 9]. A listing of the program is included in this appendix for reference.

```
C
C     ***************************HOLOVERT*******************************
C
C     HOLOVERT INVERTS THE  FIELD AT AN ARRAY OF POINTS IN A VARIABLE
C     COORDINATE SYSTEM.  IT SURVEYS NPTS POINTS EACH FOR A SET OF NLINS
C     LINES ACROSS THE FIELD.
C
      CCMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      COMMON /TAB/ INDEX,KEXTRA,MEXTRA,KLIMIT,MLIMIT,KOUT,MOUT
      COMMON /TAB2/ IPT,KPT,LPT,BND,NPTS,NLINS,RHOINF,RLAMDA,BETA
      COMMON /OUT/ XP,THEO,CALC,ERR,RHO,CA,FA
      CCMMON /EQPARA/ A,B,C,D,E,P,Q,S,T,U,V,W,RO,RA,NO,NA,NOF,NAF
      CCMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
      COMMON /IO/ CMS,IN1,IN2,IN4
      DATA BL,PL,ST,EX,OH,SC,DH,BR/1H ,1H+,1H*,1HX,1HO,1H:,1H-,1H|/
      DIMENSION RB(7),TL(62),RO(101),RA(101)
C-----------------------------------------------------------------------
      DIMENSION G(5151),GA(5151),H(202,5),SCF(73,6),BDA(4000)
      DIMENSION THEO(51,11),CALC(51,11),ERR(51),RHO(51)
      DIMENSION CA(51),FA(51,11),AR(42),XP(51),YP(51)
      CMS=0.
C-----------------------------------------------------------------------
      REWIND 3
      IN1=5
      IN2=5
      IN4=5
      IF (CMS.NE.1.) GO TO 20
      IN1=1
      IN2=2
      IN4=4
      IF (CMS.EQ.1.) REWIND 4
20    READ (IN4,89) (AR(I),I=1,42)
      IMAX=AR(1)
      JMAX=AR(2)*2.
      IF (JMAX.LE.0) JMAX=1
      KLIMIT=AR(3)
      MLIMIT=AR(4)
      KEXTRA=AR(5)
      MEXTRA=AR(6)
      JSYM=AR(14)
      ALPHA=AR(7)
      SIZE=AR(8)
      EPS=AR(9)
      MODE=AR(13)
      DGN=AR(17)
      RHOINF=AR(10)
      RLAMDA=AR(11)*1.E-8
      BETA=AR(12)
```

```
      NPTS=AR(15)                                                       CAL00480
      NLINS=AR(16)                                                      CAL00490
      SD=AR(18)                                                         CAL00500
      PHIZ=AR(19)                                                       CAL00510
      DELPHI=AR(20)                                                     CAL00520
      YPZERO=AR(21)                                                     CAL00530
      YPRNG=AR(22)                                                      CAL00540
      XPZERO=AR(23)                                                     CAL00550
      XPRNG=AR(24)                                                      CAL00560
      NOF=AR(25)                                                        CAL00570
      NAF=AR(26)                                                        CAL00580
      IPT=AR(27)                                                        CAL00590
      KPT=AR(28)                                                        CAL00600
      LPT=AR(29)                                                        CAL00610
      BND=AR(30)                                                        CAL00620
      A=AR(31)                                                          CAL00630
      B=AR(32)                                                          CAL00640
      C=AR(33)                                                          CAL00650
      D=AR(34)                                                          CAL00660
      E=AR(35)                                                          CAL00670
      P=AR(36)                                                          CAL00680
      S=AR(37)                                                          CAL00690
      T=AR(38)                                                          CAL00700
      U=AR(39)                                                          CAL00710
      V=AR(40)                                                          CAL00720
      W=AR(41)                                                          CAL00730
      Q=AR(42)                                                          CAL00740
      WRITE(6,90)
      WRITE(6,98)  (ST,I=1,75)
      WRITE(6,91) (AR(I),I=1,6)
      WRITE(6,92) (AR(I),I=7,12)
      WRITE(6,93) (AR(I),I=13,18)
      WRITE(6,94) (AR(I),I=19,24)
      WRITE(6,95) (AR(I),I=25,30)
      WRITE(6,96) (AR(I),I=31,36)
      WRITE(6,97) (AR(I),I=37,42)
      WRITE(6,98)  (ST,I=1,75)
   90 FORMAT(/////, ' THE INPUT PARAMETERS FOR THIS RUN ARE :',///)
   91 FORMAT (/5X,'*  IMAX   * JMAX/2  * KLIMIT  * MLIMIT  * KEXTRA   *',SET00310
     1' MEXTRA  *'/3X,6F10.0)
   92 FORMAT (/5X,'*  ALPHA  *  SIZE   *  EPS.   * RHO-INF * LAMBDA   *',SET00340
     1'  BETA   *'/3X,2F10.3,F11.6,2X,F8.7,F9.1,F11.6)                   SET
   93 FORMAT (/5X,'*  MODE   *  SYM.   * POINTS  *  LINES  * DIAGNOS *',SET00360
     1' STD.DEV *'/2X,5F10.0,F10.3)
   94 FORMAT (/5X,'* PHIZERO * DELPHI  * YPZERO  * YPRANGE * XPZERO   *',
     1' XPRANGE *'/4X,6F10.3)
   95 FORMAT (/5X,'* TST FUN * ADD FUN * GARRAY  *  GRAPH  * LIN PRT *',SET00400
     1' MAP BND *'/2X,6F10.0)
```

```
96    FORMAT (/5X,'*     A     *     B     *     C     *     D     *     E     *',SET00420
     1'    P     *'/4X,6F10.3)
97    FORMAT (/5X,'*     S     *     T     *     U     *     V     *     W     *',SET00440
     1'    Q     *'/4X,6F10.3)
98    FORMAT (/3X,75A1)
      IF (DGN.GE.4) WRITE (6,89) (AR(I),I=1,42)                          CAL00750
      NNN=2                                                              CAL00760
      IF (MODE.LT.0) NNN=1                                               CAL00770
      IF (MODE.GT.5) NNN=3                                               CAL00780
      IF (MODE.GT.5) MODE=MODE-10                                        CAL00790
      NGP=0                                                              CAL00800
      IF (KLIMIT.LT.KEXTRA) KEXTRA=KLIMIT                                CAL00810
      IF (MLIMIT.LT.MEXTRA) MEXTRA=MLIMIT                                CAL00820
      IF (IPT.LT.0) NGP=IPT                                              CAL00830
      IF (IPT.LT.0) IPT=-IPT                                             CAL00840
      ISYM=2.1-(FLOAT(JSYM)/2.-FLOAT(JSYM/2))*2                          CAL00850
      IF (JSYM.EQ.0) ISYM=1                                              CAL00860
      IF (JSYM.GT.JMAX) ISYM=2                                           CAL00870
      IF (ISYM.EQ.1) JMAX=((JMAX+1)/2)*2                                 CAL00880
      RJMX=JMAX                                                          CAL00890
      MSYM=JSYM                                                          CAL00900
      IF ((MSYM.EQ.0).OR.(MSYM.GT.JMAX)) MSYM=1                          CAL00910
      FCU=ISYM*JSYM*JMAX                                                 CAL00920
      IF ((JSYM.GT.JMAX).OR.(JSYM.EQ.0)) FCU=JMAX                        CAL00930
      QSYM=FCU/RJMX                                                      CAL00940
      IMS=(IMAX+ISYM-1)/ISYM                                             CAL00950
      JMS=JMAX                                                           CAL00960
      IF (ISYM.EQ.1) JMS=(JMAX/2+1)/2                                    CAL00970
      IF (JSYM.EQ.0) JMS=JMAX/2                                          CAL00980
      MODE=ABS(AR(13))                                                   CAL00990
      XO=0.                                                              CAL01000
      YO=0.                                                              CAL01010
      ZD=0.                                                              CAL01020
      PHISYM=0.                                                          CAL01030
      HS=SIZE/2.                                                         CAL01040
      RHOS=1.286                                                         CAL01050
      BOX=RHOINF*BETA/RHOS/RLAMDA                                        CAL01060
      RPTS=NPTS                                                          CAL01070
      XPR=0.                                                             CAL01080
      IF (NPTS.GT.1) XPR=XPRNG/(RPTS-1.)/2.                              CAL01090
      XPM=-XPR                                                           CAL01100
      PIE=3.141592653589793                                              CAL01110
      MONE=1                                                             CAL01120
        WRITE (6,58) IMAX,JMAX,IMS,JMS,ISYM,JSYM,MSYM,QSYM,FCU           CAL01130
58      FORMAT (3X,'IMAX,JMAX,IMS,JMS,ISYM,JSYM,MSYM,QSYM,FCU',/         CAL01140
     1  7I5,2F7.3/)                                                      CAL01150
      NTWO=2                                                             CAL01160
      IX=IMAX+JMAX                                                       CAL01170
```

```
      NF=IN1                                                            CAL01180
      IF ((MODE.EQ.1.).AND.(NOF.EQ.8).AND.(DGN.GE.1.)) WRITE (6,69)     CAL01190
      IF ((MODE.EQ.1).AND.(NOF.EQ.8)) CALL FREAD (NO,RO,NF,ZD)          CAL01200
      Z=ZD                                                              CAL01210
      IF (DGN.GE.1.) WRITE (6,68)                                       CAL01220
      CALL GARRAY (G,GA,NOF,DGN,MONE,XO,YO,PHISYM)                      CAL01230
      LM=1                                                              CAL01240
      IF ((LPT.EQ.0).AND.(BND.EQ.0)) LM=0                               CAL01250
      IIMX=IMAX+1                                                       CAL01260
      JJMX=JMAX+1                                                       CAL01270
      IJMX=IMAX*JMAX                                                    CAL01280
      NBD=1                                                             CAL01290
      IF (JSYM.EQ.0) NBD=2                                              CAL01300
      KBD=KLIMIT*NBD                                                    CAL01310
      DO 15 IJ=1,IJMX                                                   CAL01320
   15 GA(IJ)=0.                                                         CAL01330
      IF (NAF.EQ.0) GO TO 16                                            CAL01340
      NF=IN2                                                            CAL01350
      IF ((NAF.EQ.8).AND.(DGN.GE.1.)) WRITE (6,69)                      CAL01360
      IF (NAF.EQ.8) CALL FREAD (NA,RA,NF,ZD)                            CAL01370
      MST=MODE                                                          CAL01380
      MODE=1                                                            CAL01390
      IF (DGN.GE.1.) WRITE (6,68)                                       CAL01400
      IF (NAF.NE.0) CALL GARRAY (GA,G,NAF,DGN,NTWO,XO,YO,PHISYM)        CAL01410
      MODE=MST                                                          CAL01420
      DO 6 IJ=1,IJMX                                                    CAL01430
    6 G(IJ)=G(IJ)+GA(IJ)                                                CAL01440
   16 RLINS=NLINS                                                       CAL01450
      IF (NAF.EQ.8) WRITE (6,88) NA,(RA(L),L=1,NA)                      CAL01460
      IF (NOF.EQ.8) WRITE (6,87) NO,(RO(I),I=1,NO)                      CAL01470
      IF (LM.EQ.0) GO TO 14                                             CAL01480
      RB(1)=-1.                                                         CAL01490
      DO 1 I=2,7                                                        CAL01500
    1 RB(I)=RB(I-1)+.5                                                  CAL01510
      TPIE=2.*PIE                                                       CAL01520
      MPIE=-PIE                                                         CAL01530
      DYP=0.                                                            CAL01540
      DXP=0.                                                            CAL01550
      IF (NLINS.GT.1) DYP=YPRNG/(RLINS-1.)                              CAL01560
      IF (NPTS.GT.1) DXP=XPRNG/(RPTS-1.)                                CAL01570
      IF ((DGN.GE.1.).AND.(NNN.EQ.2)) WRITE (6,64)                      CAL01580
      IF(NNN.EQ.2) CALL BDGEN (G,H,SCF,DGN,NBD,BDA,KBD)                 CAL01590
      DO 5 J=1,NLINS                                                    CAL01600
      IF (DGN.GE.1.) WRITE (6,67) J                                     CAL01610
      RJM=J-1                                                           CAL01620
      PHI=PHIZ+DELPHI*RJM                                               CAL01630
      YP(J)=YPZERO+DYP*RJM                                              CAL01640
      PSI=(PHI+90.)*PIE/180.                                            CAL01650
```

```
      TAU=PSI-PHISYM                                                    CAL01660
      IF (LPT.EQ.0) GO TO 9                                             CAL01670
      IF (LPT.LE.1) WRITE (6,78) (ST,I=1,124)                           CAL01680
      IF (LPT.GT.1) WRITE (6,74) (ST,I=1,95)                            CAL01690
      IF ((CMS.EQ.1.).AND.(LPT.GT.1)) READ (5,79) ZZ                    CAL01700
      WRITE (6,86)                                                      CAL01710
      WRITE (6,85) Z,PHI,YP(J)                                          CAL01720
      WRITE (6,76)                                                      CAL01730
      IF (MODE.EQ.1) WRITE (6,83) (RB(I),I=1,7)                         CAL01740
      IF (MODE.GT.1) WRITE (6,80) (RB(I),I=1,7)                         CAL01750
      WRITE (6,81) (DH,I=1,54),(PL,II=1,13)                             CAL01760
    9 IC=0                                                              CAL01770
      DO 3 I=1,NPTS                                                     CAL01780
      RIM=I-1                                                           CAL01790
      THEO(I,J)=0.                                                      CAL01800
      CA(I)=0.                                                          CAL01810
      FA(I,J)=0.                                                        CAL01820
      ERR(I)=0.                                                         CAL01830
      CALC(I,J)=0.                                                      CAL01840
      RHO(I)=0.                                                         CAL01850
      XP(I)=XPZERO+DXP*RIM                                              CAL01860
      XPII=ABS(XP(I))                                                   CAL01870
      IF (XPII.LT.1.E-10) XP(I)=0.                                      CAL01880
      RS=SQRT(XP(I)**2+YP(J)**2)/HS                                     CAL01890
      IF (RS.GT.1.) GO TO 13                                            CAL01900
      THT=ATANM(YP(J),XP(I))                                            CAL01910
      IF (XPII.EQ.0.) THT=0.                                            CAL01920
      SIG=TAU-PIE/2.+THT                                                CAL01930
      IF (SIG.GT.PIE) SIG=SIG-TPIE                                      CAL01940
      IF (SIG.LT.MPIE) SIG=SIG+TPIE                                     CAL01950
      SIGI=SIG                                                          CAL01960
      XS=RS*COS(SIG)                                                    CAL01970
      IF (DGN.GE.1.) WRITE (6,44) SIG                                   CAL01980
   44 FORMAT ('    SIG=',E10.3)                                         CAL01990
      YS=RS*SIN(SIG)                                                    CAL02000
      IF(DGN.GE.5) WRITE (6,57) PHI,DELPHI,PSI,TAU,THT,SIG,SIGI,XS,YS   CAL02010
   57   FORMAT ('  ANGLES=',10E10.3)                                    CAL02020
      RI=I                                                              CAL02030
      F=0.                                                              CAL02040
      IF (DGN.GE.2.) WRITE (6,66) I                                     CAL02050
      CALL FUNCT (XS,YS,FA(I,J),NAF,DGN,NTWO)                           CAL02060
      IF (MODE.EQ.1) CALL FUNCT (XS,YS,F,NOF,DGN,MONE)                  CAL02070
      THEO(I,J)=F                                                       CAL02080
      IF (NNN.GE.2) REWIND 3                                            CAL02090
      IF(NNN.GE.2) CALL FIELD (RS,SIGI,SOLN,NBD,BDA,DGN,KBD)            CAL02100
      IF(NNN.EQ.1) CALL FIELD2 (RS,SIGI,SOLN,G,H,SCF,DGN)               CAL02110
      CA(I)=SOLN/BOX/HS                                                 CAL02120
      CALC(I,J)=CA(I)-FA(I,J)                                           CAL02130
```

```
      RHO(I)=RHOINF*(CALC(I,J)+1.)
      ERR(I)=CA(I)
      IF (MODE.EQ.1) ERR(I)=(CALC(I,J)-THEO(I,J))
      IF (MODE.GT.1) THEO(I,J)=FA(I,J)
      IF (LPT.EQ.0) GO TO 3
13    LC=0
      TL(1)=BL
      TTL=0.
      IF ((XP(I).GT.XPM).AND.(XP(I).LT.XPR)) TTL=1.
      IF (IC.EQ.5) IC=0
      IF (IC.EQ.0) TL(1)=PL
      DO 2 L=2,62
      TL(L)=BL
      IF ((I.EQ.1).OR.(TTL.EQ.1.).OR.(I.EQ.NPTS)) TL(L)=PL
      IF (LC.EQ.13) LC=0
      IF ((IC.EQ.0).AND.(LC.EQ.0)) TL(L)=PL
2     LC=LC+1
      TL(2)=PL
      TL(22)=PL
      TL(62)=PL
      IC=IC+1
      RLW=(CA(I)+1.)*20.+2.5
      LW=RLW
      IF (LW.GT.62) LW=62
      IF (LW.LT.2) LW=2
      TL(LW)=SC
      RLY=(FA(I,J)+1.)*20.+2.5
      LY=RLY
      IF (LY.GT.62) LY=62
      IF (LY.LT.2) LY=2
      IF (NAF.NE.0) TL(LY)=ST
      RLX=(THEO(I,J)+1.)*20.+2.5
      LX=RLX
      IF (LX.GT.62) LX=62
      IF (LX.LT.2) LX=2
      IF (MODE.EQ.1) TL(LX)=OH
      RLZ=(CALC(I,J)+1.)*20.+2.5
      LZ=RLZ
      IF (LZ.GT.62) LZ=62
      IF (LZ.LT.2) LZ=2
      TL(LZ)=EX
      WRITE (6,82)MOUT,KOUT,INDEX,THEO(I,J),ERR(I),CALC(I,J),RHO(I),
     1XP(I),(TL(L),L=1,62)
      IF ((NPTS.LE.20).AND.(I.NE.NPTS)) WRITE (6,79)
3     CONTINUE
      IF (LPT.NE.0) WRITE (6,81) (DH,I=1,54),(PL,I=1,13)
      TMAX=0.
      TMIN=0.
```

```
      IE=0                                                              CAL02620
      BE=0.                                                             CAL02630
      EB=0.                                                             CAL02640
      DO 4 I=1,NPTS                                                     CAL02650
      TH=THEO(I,J)                                                      CAL02660
      IF (TH.GT.TMAX) TMAX=TH                                           CAL02670
      IF (TH.LT.TMIN) TMIN=TH                                           CAL02680
      ER=ABS(CALC(I,J)-TH)                                              CAL02690
      IF (ER.LE.BE) GO TO 4                                             CAL02700
      BE=ER                                                             CAL02710
      IE=I                                                              CAL02720
    4 CONTINUE                                                          CAL02730
      TMM=TMAX-TMIN                                                     CAL02740
      EB=RHOINF*(CALC(IE,J)-THEO(IE,J))                                 CAL02750
      IF (TMM.NE.0.) BE=(CALC(IE,J)-THEO(IE,J))*100./TMM                CAL02760
      IF ((MODE.EQ.1).AND.(LPT.NE.0)) WRITE (6,75) EB,XP(IE),BE         CAL02770
      IF (DELPHI.NE.0.) YP(J)=PHI                                       CAL02780
    5 CONTINUE                                                          CAL02790
      IF (BND.EQ.0.) GO TO 14                                           CAL02800
      IF (LPT.EQ.1) WRITE (6,78) (ST,I=1,124)                           CAL02810
      IF (LPT.GT.1) WRITE (6,74) (ST,I=1,95)                            CAL02820
      IF ((CMS.EQ.1.).AND.(LPT.GT.1)) READ (5,79) ZZ                    CAL02830
      IF (DGN.GE.1.) WRITE (6,63)                                       CAL02840
      CALL MAP (NPTS,NLINS,CALC,NOF,Z,BND)                              CAL02850
   14 IF (NAF.EQ.0) GO TO 10                                            CAL02860
      NAO=10*NOF+NAF                                                    CAL02870
      IF ((DGN.GE.1.).AND.(NGP.EQ.-3)) WRITE (6,62)                     CAL02880
      IF (NGP.EQ.-3) CALL GPUNCH (Z,XO,YO,PHISYM,NAO,IMAX,JMAX,G)       CAL02890
      DO 7 IJ=1,IJMX                                                    CAL02900
    7 G(IJ)=G(IJ)-GA(IJ)                                                CAL02910
   10 IF (IPT.LE.0) GO TO 11                                            CAL02920
      IF ((IPT.EQ.1).OR.(IPT.EQ.3)) WRITE (6,78) (ST,I=1,124)           CAL02930
      IF ((IPT.EQ.2).OR.(IPT.GE.4)) WRITE (6,74) (ST,I=1,95)            CAL02940
      IF ((CMS.EQ.1.).AND.((IPT.EQ.2).OR.(IPT.GE.4))) READ (5,79) ZZ    CAL02950
      CALL GPRINT (G,MONE)                                              CAL02960
      IF (NGP.EQ.-1) CALL GPUNCH (Z,XO,YO,PHISYM,NOF,IMAX,JMAX,G)       CAL02970
      IF (IPT.EQ.3) WRITE (6,78) (ST,I=1,124)                           CAL02980
      IF (IPT.GE.4) WRITE (6,74) (ST,I=1,95)                            CAL02990
      IF ((CMS.EQ.1.).AND.(IPT.GE.4)) READ (5,79) ZZ                    CAL03000
      IF (IPT.GE.3) CALL GPRINT (GA,NTWO)                               CAL03010
   11 IF (KPT.LE.0) GO TO 12                                            CAL03020
      IF ((KPT.EQ.1).OR.(KPT.EQ.3)) WRITE (6,78) (ST,I=1,124)           CAL03030
      IF ((KPT.EQ.2).OR.(KPT.GE.4)) WRITE (6,74) (ST,I=1,95)            CAL03040
      IF ((CMS.EQ.1.).AND.((KPT.EQ.2).OR.(KPT.GE.4))) READ (5,79) ZZ    CAL03050
      IF (DGN.GE.1.) WRITE (6,61)                                       CAL03060
      CALL GPLOT (G,GA,JMS)                                             CAL03070
   12 WRITE (6,73) (EX,I=1,124)                                         CAL03080
      AGAIN=ST                                                          CAL03090
```

```
      IF (CMS.NE.1.) READ(5,60) AGAIN
      IF (AGAIN.EQ.BL) GO TO 20
      WRITE (6,77)
89    FORMAT (6F12.7)
88    FORMAT (/'     THE INPUT DATA FOR ADD-ON FUNCTION NO.8 (',I3,
     1' POINTS) WAS:'/7(11F10.3/))
87    FORMAT (/'     THE INPUT DATA FOR THE TEST FUNCTION NO.8, (',I3,
     1' POINTS) WAS:'/7(11F10.3/))
86    FORMAT (1H1//'    THE INVERTED CROSS SECTION FOR:')
85    FORMAT (10X,'Z   =',F8.3,' CM.'/10X,'PHI=',F8.3,' DEGREES'/10X,
     14HY' =,F8.3,' CM.',44X,'O = ORIGINAL FUNCTION'/
     270X,'* = ADD-ON FUNCTION')
84    FORMAT (' ADJUST PAGE, HIT SPACE AND RETURN.')
83    FORMAT ('   LIMIT  MAX  ORIGINAL      ABS.  COMPUTED (MG/CC)'/'   M
     1   K TERM  FUNCTION    ERROR  FUNCTION DENSITY ',6H   X'  ,F4.1,
     26F10.1)
82    FORMAT (2X,I2,1X,I3,1X,I4,1X,F9.4,2X,F7.3,1X,F9.4,1X,F7.3,
     1F7.3,1X,62A1)
81    FORMAT (3X,54A1,2X,A1,12(4X,A1))
80    FORMAT ('   LIMIT  MAX',3X,'ADD-ON',6X,'THE',4X,'DENSITY (MG/CC)'/
     1'   M   K TERM',2X,'FUNCTION',5X,'SUM',3X,'FUNCTION DENSITY ',
     23H   X'  ,F4.1,6F10.1)
79    FORMAT (1X,F1O.3)
78    FORMAT (1X,124A1)
77    FORMAT (1X//)
76    FORMAT (4X,'FUNCTION = (RHO/RHO-INFINITY)-1.0',33X,': = THE'
     1,' INVERTED SUM'/70X,'X = COMPUTED FUNCTION')
75    FORMAT ('      LARGEST ERROR: ',F8.6,' GMS/CC;       AT ',3HX'=,F6.3
     1,':',4X,F10.2,' PERCENT.'//)
74    FORMAT (1X,47A1,' SET PAGE, HIT SPACE, RETURN ',48A1)
69    FORMAT ('   CALL FREAD')
68    FORMAT ('   CALL GARRAY')
67    FORMAT ('   LINE',I3,' DO LOOP')
66    FORMAT ('   POINT',I3,' CALL FUNCT')
65    FORMAT ('   CALL FIELD')
64    FORMAT ('   CALL BDGEN')
63    FORMAT ('   CALL MAP')
62    FORMAT ('   CALL GPUNCH')
61    FORMAT ('   CALL GPLOT')
60    FORMAT (80A1)
      STOP
      END

C000001
C
```

```
      SUBROUTINE BDGEN (G,H,SCF,DGN,NBD,BDA,KBD)
C
C     BDGEN EVALUATES THE B AND D COEFFICIENTS FOR ALL M AND K, AND WRITES
C     THE ARRAY LINEARLY ON DISK.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      COMMON /TAB/ INDEX,KEXTRA,MEXTRA,KLIMIT,MLIMIT,KOUT,MOUT
      COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
      DIMENSION G(IJMX),H(IIMX,5),SCF(JJMX,6),BDA(KBD)
C     INITIALIZE THE VALUES:
      INDEX=0
      KL2=NBD*KLIMIT
      REWIND 3
      JJMX6=JJMX*6
      IIMX2=(IIMX+1)/2
      PIE=3.141592653589793
      RIMAX=IMAX
      KLMP=KLIMIT+1
      DX=2./RIMAX
      RJMAX=JMAX
      DXI=2.*PIE/FCU
C     INITIALIZE THE MODIFIED HERMITE POLYNOMIAL ARRAY;  VECTORS:
C     (1)=H1, (2)=H2, (3)=ALPHA*X(I), (4)=HM+2 STORED, (5)=HM+1 STORED
      DO 1 II=1,IIMX2
      RII=II
      IIM=IIMX-II+1
      H(II,3)=ALPHA*(RII*DX-DX-1.)
      H(IIM,3)=-H(II,3)
      H(II,1)=2.*H(II,3)
      H(II,2)=(H(II,3)*H(II,1)-1.)/3.
      H(IIM,1)=-H(II,1)
      H(IIM,2)=H(II,2)
      H(II,5)=H(II,2)
      H(IIM,5)=H(IIM,2)
      H(II,4)=H(II,1)
    1 H(IIM,4)=H(IIM,1)
      SIGN=1.
C     INITIATE THE SIN/COS ARRAY:
      DO 2 J=1,JJMX
      RJM=J-1
      SCF(J,1)=0.
      SCF(J,2)=1.
      SCF(J,3)=SIN(RJM*DXI-PIE/2.)
      SCF(J,4)=COS(RJM*DXI-PIE/2.)
      SCF(J,5)=0.
    2 SCF(J,6)=0.
      MS=0
C     COMMENCE THE M LOOP:
```

```
      DO 7 MP=1,MLIMIT
      M=MP-1
      RM=M
      SIGN=-SIGN
      IF (DGN.LE.-4) WRITE (6,88) SCF(1,1),SCF(2,1),SCF(1,2),SCF(2,2)
C     TEST FOR SYMMETRY SKIPS:
      IF (MS.EQ.MSYM) MS=0
      TOTAL=0.
      MS=MS+1
      IF (MS.NE.1) GO TO 6
C     COMMENCE THE K LOOP:
      DO 5 KP=1,KLIMIT
      K=KP-1
      PK=KP
      RK=K
      INDEX=INDEX+1
C     CALL THE B & D COEFFICIENTS AND WRITE THEM ON DISK:
      CALL BD (M,K,G,H,SCF,B,D,JJMX6)
      IF(DGN.EQ.3.) WRITE(6,89) M,K,B,D
      IF (DGN.LE.-2) WRITE (6,89) M,K,B,D
      IF (DGN.LE.-4) WRITE (6,88) H(1,1),H(1,2),H(1,4),H(1,5)
      KK=K*NBD+1
      K2=KP*NBD
      BDA(K2)=D
      BDA(KK)=B
C     GENERATE THE NEXT ORDER OF THE SET OF HERMITE POLYNOMIALS FOR NEW K:
      ORDER=M+2*KP+1
      HA=SQRT(PK*(PK+RM))/ORDER
      HB=2.*SQRT((PK+1.)*(RM+PK+1.))/(ORDER+1.)/(ORDER+2.)
      DO 5 II=1,IIMX2
      IIM=IIMX-II+1
      H(II,1)=2.*(H(II,3)*H(II,2)-HA*H(II,1))
      H(IIM,1)=SIGN*H(II,1)
5     H(II,2)=HB*(H(II,3)*H(II,1)-ORDER*H(II,2))
C     ADVANCE THE SIN/COS ARRAY FOR THE NEXT M:
      DO 3 J=1,JJMX
      IF (DGN.LE.-5) WRITE (6,87) (SCF(J,NT),NT=1,6)
87      FORMAT ('  SIN/COS MXI:'8E10.3)
      STEMP=SCF(J,1)
      SCF(J,1)=SCF(J,1)*SCF(J,4)+SCF(J,2)*SCF(J,3)
3     SCF(J,2)=SCF(J,2)*SCF(J,4)-STEMP*SCF(J,3)
      DO 4 J=1,JMAX
      SCF(J,5)=SCF(J+1,1)-SCF(J,1)
4     SCF(J,6)=SCF(J+1,2)-SCF(J,2)
      WRITE (3) (BDA(I),I=1,KBD)
      IF (DGN.LE.-3) WRITE (6,88) (BDA(I),I=1,10)
      IF (JSYM.GT.JMAX) RETURN
6     RM=RM+1
```

```
C     REGENERATE THE HERMITE ARRAY FOR NEW M, K=0:
        DO 7 II=1,IIMX2
        IIM=IIMX-II+1
        H(II,2)=H(II,4)*SQRT(RM)/(RM+1.)
        H(II,1)=H(II,5)*(RM+2.)
        H(IIM,1)=-SIGN*H(II,1)
        H(II,2)=2.*SQRT(RM+1)*(H(II,3)*H(II,1)-(RM+1.)*H(II,2))
        H(II,2)=H(II,2)/(RM+2.)/(RM+3.)
        H(II,4)=H(II,1)
7       H(II,5)=H(II,2)
89      FORMAT ('    M=',I4,',    K=',I4,',    B=',E10.4,',    D=',E10.4)
88      FORMAT (2X,10E10.3)
        RETURN
        END
C000002
C


        SUBROUTINE FIELD (RS,SIG,SOLN,NBD,BDA,DGN,KBD)
C
C     FIELD EVALUATES THE VALUE OF THE FIELD FUNCTION AT A PARTICULAR
C     POINT DESIGNATED IN CYLINDRICAL COORDINATES, BY USING THE INVERSION
C     EQUATION OF MALDONADO, ET.AL.  FIELD USES THE ARRAY OF  B & D
C     COEFFICIENTS GENERATED IN SUBROUTINE BDGEN.
C
        COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
        COMMON /TAB/ INDEX,KEXTRA,MEXTRA,KLIMIT,MLIMIT,KOUT,MOUT
        COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
        DIMENSION BDA(KBD), STK(52), STM(52)
C     INITIALIZE THE VALUES:
        INDEX=0
        MTIMER=0
        KOUT=0
        MOUT=0
        MMAX=0
        KMAX=0
        TOTAL=0.
        JJMX6=JJMX*6
        REWIND 3
        AR=ALPHA*RS
        ARG=AR**2
        EXPON=EXP(-ARG)
        PIE=3.141592653589793
        APP=ALPHA/PIE/PIE
        M=0
        RM=M
        RIMAX=IMAX
        DX=2./RIMAX
```

```
      RJMAX=JMAX
      SIGN=1.
      STK(1)=0.
      STM(1)=0.
      SMS=0.
      CMS=1.
      SMI=SIN(SIG)
      CMI=COS(SIG)
      MEP=MEXTRA+1
      DO 16 MB=1,MEP
16    STM(MB)=0.
      FM=1.
      MS=0
C   COMMENCE THE M LOOP:
2     SIGN=-SIGN
      K=0
      RK=K
      RM=M
      ARM=1.
      IF (M.NE.0) ARM=AR**M
      KTIMER=0
      KEP=KEXTRA+1
      DO 15 KB=1,KEP
15    STK(KB)=0.
      SIGNK=-1.
C   COMPUTE THE K=0 & K=1 ORDERS OF LAGUERRE POLYNOMIAL FOR GIVEN M:
      PM=0.
      P=SQRT(1./FM)
      PP=(RM+1.-ARG)*SQRT(1./FM/(RM+1.))
C   TEST FOR SYMMETRY SKIPS:
      IF (MS.EQ.MSYM) MS=0
      MS=MS+1
      IF (MS.NE.1) GO TO 7
C   READ A LINE OF B & D COEFFICIENTS FOR GIVEN M:
      READ (3) (BDA(I),I=1,KBD)
      IF (DGN.LE.-6) WRITE (6,88) (BDA(I),I=1,10)
C   COMMENCE THE K LOOP:
3     INDEX=INDEX+1
      SIGNK=-SIGNK
C   COMPUTE THE M,K SUMMATION TERM:
      KK=K*NBD+1
      B=BDA(KK)
      D=0.
      IF (NBD.EQ.2) D=BDA(KK+1)
      BRAKET=B
      IF(RM.EQ.0.) GO TO 4
      BRAKET=B*CMS+D*SMS
4     ADD=SIGNK*BRAKET*P*ARM
```

```
      TOTAL=TOTAL+ADD                                                   SUB01920
      IF (DGN.GT.-5) GO TO 5                                            SUB01930
      STOT=TOTAL*EXPON*APP/BOX/SIZE                                     SUB01940
      WRITE (6,89) M,K,STOT,ADD,BRAKET,P,ARM,B,CMS,D,SMS                SUB01950
C     ESTABLISH CHECK AS THE RELATIVE SIZE OF THE M,K TERM OF THE SERIES: SUB01960
5     CHECK=ABS(ADD)                                                    SUB01970
      IF (TOTAL.GT.EPS) CHECK=ABS(ADD/TOTAL)                            SUB01980
C     ADVANCE THE K INDEX:                                              SUB01990
      K=K+1                                                             SUB02000
      RK=K                                                              SUB02010
      DO 10 KA=1,KEXTRA                                                 SUB02020
      KB=KEXTRA-KA+1                                                    SUB02030
10    STK(KB+1)=STK(KB)                                                 SUB02040
      STK(2)=TOTAL                                                      SUB02050
      ORDER=M+2*K+1                                                     SUB02060
C     GENERATE THE NEXT ORDER OF LAGUERRE POLYNOMIAL FOR NEW K:         SUB02070
      PM=P                                                              SUB02080
      P=PP                                                              SUB02090
      PP=P*(ORDER-ARG)-PM*SQRT(RK*(RM+RK))                              SUB02100
      PP=PP/SQRT((RK+1.)*(RM+RK+1.))                                    SUB02110
C     SET K TIMER TO PROVIDE EXTRA K TERMS AFTER CHECK < EPS:           SUB02120
      KTIMER=KTIMER+1                                                   SUB02130
      IF (K.GE.KLIMIT) GO TO 6                                          SUB02140
      IF (CHECK.GE.EPS) KTIMER=0                                        SUB02150
      IF (KTIMER.LE.KEXTRA) GO TO 3                                     SUB02160
      GO TO 7                                                           SUB02170
6     KOUT=KOUT+1                                                       SUB02180
      IF (KEXTRA.EQ.0) GO TO 7                                          SUB02190
      TOTAL=0.                                                          SUB02200
      DO 11 KA=1,KEXTRA                                                 SUB02210
11    TOTAL=TOTAL+STK(KA+1)                                             SUB02220
      RKX=KEXTRA                                                        SUB02230
      TOTAL=TOTAL/RKX                                                   SUB02240
C     END OF K LOOP:  ADVANCE M:                                        SUB02250
7     M=M+1                                                             SUB02260
      RM=M                                                              SUB02270
      STP=SMS                                                           SUB02280
      SMS=SMS*CMI+CMS*SMI                                               SUB02290
      CMS=CMS*CMI-STP*SMI                                               SUB02300
      IF (K.GT.KMAX) KMAX=K                                             SUB02310
      FM=FM*RM                                                          SUB02320
      DO 12 MA=1,MEXTRA                                                 SUB02330
      MB=MEXTRA-MA+1                                                    SUB02340
12    STM(MB+1)=STM(MB)                                                 SUB02350
      STM(2)=TOTAL                                                      SUB02360
C     SET M TIMER FOR EXTRA M TERMS:                                    SUB02370
      IF (MS.EQ.1) MTIMER=MTIMER+1                                      SUB02380
      IF (JSYM.GT.JMAX) GO TO 9                                         SUB02390
```

```
      IF (K.GT.KEXTRA) MTIMER=0
      IF (M.GE.MLIMIT) GO TO 13
      IF (MTIMER.LE.MEXTRA) GO TO 2
13    IF (MEXTRA.EQ.0) GO TO 9
      TOTAL=0.
      DO 14 MA=1,MEXTRA
14    TOTAL=TOTAL+STM(MA+1)
      RMX=MEXTRA
      TOTAL=TOTAL/RMX
C   END OF M LOOP;  COMPUTE OUTPUT SOLN.
9     MOUT=M-1
      IF (KOUT.EQ.0) KOUT=KMAX-1
      SOLN=TOTAL*EXPON*APP/2.
89    FORMAT ('    M=',I4,',   K=',I4,',   SUBTOTAL=',9E10.3)
88    FORMAT (2X,10E10.3)
      RETURN
      END
C000003
C


      SUBROUTINE BD (M,K,G,H,SCF,B,D,JJMX6)
C
C
C   BD EVALUATES THE FIRST (B) AND SECOND (D) COEFFICIENTS IN THE
C   INVERSION EQUATION, FOR A PARTICULAR SET OF INDEXES M & K.
C   BD MAKES USE OF THE HERMITE POLYNOMIAL ARRAY GENERATED BY
C   SUBROUTINE FIELD AS M & K ADVANCE.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
      DIMENSION G(IJMX),SCF(JJMX6),H(IIMX)
      PIE=3.141592653589793
      B=0.
      D=0.
      RM=M
      RK=K
      RJMAX=JMAX
      JJMX4=4*JJMX
      DXI=2.*PIE/FCU
 200  FORMAT(1X,I10,/)
      IF (JSYM.LE.0) GO TO 4
      IF (M.NE.0) GO TO 2
      S=DXI
      DO 1 J=1,JMAX
      DO 1 I=1,IMAX
      II=I+1
      IJ=IMAX*(J-1)+I
```

```
      DH=H(II)-H(I)
1     B=B+G(IJ)*S*DH
      B=B*QSYM/2.
      RETURN
2     DO 3 J=1,JMAX
      JS=J+JJMX4
      S=SCF(JS)/RM
      DO 3 I=1,IMAX
      II=I+1
      IJ=IMAX*(J-1)+I
      DH=H(II)-H(I)
3     B=B+G(IJ)*S*DH
      B=B*QSYM
      RETURN
4     IF (M.NE.0) GO TO 6
      S=DXI
      DO 5 J=1,JMAX
      DO 5 I=1,IMAX
      II=I+1
      IJ=IMAX*(J-1)+I
      DH=H(II)-H(I)
5     B=B+G(IJ)*S*DH
      B=B/2.
      RETURN
6     DO 7 J=1,JMAX
      JS=J+JJMX4
      J2=JS+JJMX
      S=SCF(JS)/RM
      C=SCF(J2)/RM
      DO 7 I=1,IMAX
      II=I+1
      IJ=IMAX*(J-1)+I
      DH=H(II)-H(I)
      B=B+G(IJ)*S*DH
66    FORMAT ('  BD:',4I5,10F6.2)
7     D=D-G(IJ)*C*DH
      RETURN
      END
C000004
C

      SUBROUTINE FIELD2 (RS,SIG,SOLN,G,H,SCF,DGN)
C
C   FIELD2 COMPUTES THE SAME INVERSION AS SUBROUTINE FIELD, EXCEPT THAT
C   THE COEFFICIENTS B AND D ARE COMPUTED INDIVIDUALLY AS USED BY
C   CALLING BD.  DISK STORAGE IS NOT REQUIRED, BUT COMPUTING TIME IS
C   MUCH GREATER.  FIELD2 IS UTILIZED BY SPECIFYING A NEGATIVE MODE ON
```

```
C     THE INPUT PARAMETER.                                                  SUB03310
C     FIELD EVALUATES THE VALUE OF THE FIELD FUNCTION AT A PARTICULAR       SUB03320
C     POINT DESIGNATED IN CYLINDRICAL COORDINATES, BY USING THE INVERSION   SUB03330
C     EQUATION OF MALDONADO, ET AL.  FIELD CALLS SUBROUTINES BD & GARRAY.   SUB03340
C                                                                           SUB03350
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z       SUB03360
      COMMON /TAB/ INDEX,KEXTRA,MEXTRA,KLIMIT,MLIMIT,KOUT,MOUT              SUB03370
      COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM                          SUB03380
      DIMENSION G(IJMX),H(IIMX,5),SCF(JJMX,6)                               SUB03390
C     INITIALIZE THE VALUES:                                                SUB03400
      INDEX=0                                                               SUB03410
      MTIMER=0                                                              SUB03420
      KOUT=0                                                                SUB03430
      MOUT=0                                                                SUB03440
      MMAX=0                                                                SUB03450
      KMAX=0                                                                SUB03460
      TOTAL=0.                                                              SUB03470
      JJMX6=JJMX*6                                                          SUB03480
      IIMX2=(IIMX+1)/2                                                      SUB03490
      AR=ALPHA*RS                                                           SUB03500
      ARG=AR**2                                                             SUB03510
      EXPON=EXP(-ARG)                                                       SUB03520
      PIE=3.141592653589793                                                 SUB03530
      APP=ALPHA/PIE/PIE                                                     SUB03540
      M=0                                                                   SUB03550
      RM=M                                                                  SUB03560
      RIMAX=IMAX                                                            SUB03570
      DX=2./RIMAX                                                           SUB03580
      RJMAX=JMAX                                                            SUB03590
      DXI=2.*PIE/FCU                                                        SUB03600
C     INITIALIZE THE MODIFIED HERMITE POLYNOMIAL ARRAY;  VECTORS:           SUB03610
C     (1)=H1, (2)=H2, (3)=ALPHA*X(I), (4)=HM+2 STORED, (5)=HM+1 STORED      SUB03620
      DO 1 II=1,IIMX2                                                       SUB03630
      RII=II                                                                SUB03640
      IIM=IIMX-II+1                                                         SUB03650
      H(II,3)=ALPHA*(RII*DX-DX-1.)                                          SUB03660
      H(IIM,3)=-H(II,3)                                                     SUB03670
      H(II,1)=2.*H(II,3)                                                    SUB03680
      H(II,2)=(H(II,3)*H(II,1)-1.)/3.                                       SUB03690
      H(IIM,1)=-H(II,1)                                                     SUB03700
      H(IIM,2)=H(II,2)                                                      SUB03710
      H(II,5)=H(II,2)                                                       SUB03720
      H(IIM,5)=H(IIM,2)                                                     SUB03730
      H(II,4)=H(II,1)                                                       SUB03740
    1 H(IIM,4)=H(IIM,1)                                                     SUB03750
      SIGN=1.                                                               SUB03760
      FM=1.                                                                 SUB03770
C     INITIATE THE SIN/COS ARRAY:                                           SUB03780
```

```
      DO 11 J=1,JJMX
      RJM=J-1
      SCF(J,1)=0.
      SCF(J,2)=1.
      SCF(J,3)=SIN(RJM*DXI-PIE)
      SCF(J,4)=COS(RJM*DXI-PIE)
      SCF(J,5)=0.
11    SCF(J,6)=0.
      MS=0
C     COMMENCE THE M LOOP:
2     SIGN=-SIGN
      K=0
      RK=K
      ARM=1.
      IF (M.NE.0) ARM=AR**M
      KTIMER=0
      SIGNK=-1.
C     COMPUTE THE K=0 & K=1 ORDERS OF LAGUERRE POLYNOMIAL FOR GIVEN M:
      PM=0.
      P=SQRT(1./FM)
      PP=(RM+1.-ARG)*SQRT(1./FM/(RM+1.))
C     ADVANCE THE SIN/COS ARRAY FOR NEW M:
      DO 12 J=1,JJMX
      SCF(J,1)=SCF(J,1)*SCF(J,4)+SCF(J,2)*SCF(J,3)
12    SCF(J,2)=SCF(J,2)*SCF(J,4)-SCF(J,1)*SCF(J,3)
      DO 13 J=1,JMAX
      SCF(J,5)=SCF(J+1,1)-SCF(J,1)
13    SCF(J,6)=SCF(J+1,2)-SCF(J,2)
C     TEST FOR SYMMETRY SKIPS:
      IF (MS.EQ.MSYM) MS=0
      TOTAL=0.
      MS=MS+1
      IF (MS.NE.1) GO TO 7
      RMS=RM*SIG
      CMS=COS(RMS)
      SMS=SIN(RMS)
C     COMMENCE THE K LOOP:
3     INDEX=INDEX+1
      SIGNK=-SIGNK
C     CALL THE B & D COEFFICIENTS AND COMPUTE THE M,K SUMMATION TERM:
      CALL BD (M,K,G,H,SCF,B,D,JJMX6)
      IF (DGN.LE.-2.) WRITE (6,89) M,K,B,D
      BRAKET=B
      IF(RM.EQ.0.) GO TO 4
      BRAKET=B*CMS+D*SMS
4     ADD=SIGNK*BRAKET*P*ARM
      TOTAL=TOTAL+ADD
C     ESTABLISH CHECK AS THE RELATIVE SIZE OF THE M,K TERM OF THE SERIES:
```

```
      CHECK=ABS(ADD)
      IF (TOTAL.GT.EPS) CHECK=ABS(ADD/TOTAL)
C    ADVANCE THE K INDEX:
      K=K+1
      RK=K
      ORDER=M+2*K+1
C    GENERATE THE NEXT ORDER OF LAGUERRE POLYNOMIAL FOR NEW K:
      PM=P
      P=PP
      PP=P*(ORDER-ARG)-PM*SQRT(RK*(RM+RK))
      PP=PP/SQRT((RK+1.)*(RM+RK+1.))
C    GENERATE THE NEXT ORDER OF THE SET OF HERMITE POLYNOMIALS FOR NEW K:
      HA=SQRT(RK*(RK+RM))/ORDER
      HB=2.*SQRT((RK+1.)*(RM+RK+1.))/(ORDER+1.)/(ORDER+2.)
      DO 5 II=1,IIMX2
      IIM=IIMX-II+1
      H(II,1)=2.*(H(II,3)*H(II,2)-HA*H(II,1))
      H(IIM,1)=SIGN*H(II,1)
5     H(II,2)=HB*(H(II,3)*H(II,1)-ORDER*H(II,2))
C    SET K TIMER TO PROVIDE EXTRA K TERMS AFTER CHECK < EPS:
      KTIMER=KTIMER+1
      IF (K.GE.KLIMIT) GO TO 6
      IF (CHECK.GE.EPS) KTIMER=0
      IF (KTIMER.LE.KEXTRA) GO TO 3
      GO TO 7
C    END OF K LOOP;  ADVANCE M AND COMPUTE NEW TOTAL:
6     KOUT=KOUT+1
7     M=M+1
      IF (K.GT.KMAX) KMAX=K
      RM=M
      FM=FM*RM
C    REGENERATE THE HERMITE ARRAY FOR NEW M, K=0:
      DO 8 II=1,IIMX2
      IIM=IIMX-II+1
      H(II,2)=H(II,4)*SQRT(RM)/(RM+1.)
      H(II,1)=H(II,5)*(RM+2.)
      H(IIM,1)=-SIGN*H(II,1)
      H(II,2)=2.*SQRT(RM+1.)*(H(II,3)*H(II,1)-(RM+1.)*H(II,2))
      H(II,2)=H(II,2)/(RM+2.)/(RM+3.)
      H(II,4)=H(II,1)
8     H(II,5)=H(II,2)
C    SET M TIMER FOR EXTRA M TERMS:
      IF (MS.EQ.1) MTIMER=MTIMER+1
      IF (JSYM.GT.JMAX) GO TO 9
      IF (K.GT.KEXTRA) MTIMER=0
      IF (M.GE.MLIMIT) GO TO 9
      IF (MTIMER.LE.MEXTRA) GO TO 2
C    END OF M LOOP;  COMPUTE OUTPUT SOLN.
```

```
9     MOUT=M-1
      IF (KOUT.EQ.0) KOUT=KMAX-1
      SOLN=TOTAL*EXPON*APP/2.
89    FORMAT ('   M=',I4,',    K=',I4,',    B=',E10.4,',    D=',E10.4)
      RETURN
      END

C000005
C

      SUBROUTINE GARRAY (G,GA,NOF,DGN,NUMB,XO,YO,PHISYM)
C
C   GARRAY FILLS THE DATA ARRAY OVER AN ORTHOGONAL AREA WITH
C   THE REGULAR DATA OBTAINED BY THE METHOD CORRESPONDING TO THE
C   PARTICULAR MODE:
C
C        MODE 1  -  DATA OBTAINED BY SAMPLING A KNOWN FUNCTION SUPPLIED
C                   IN SUBROUTINE FUNCT AND SAMPLED IN SUBROUTINE GOLF.
C
C        MODE 2  -  DATA OBTAINED BY GENERATING A REGULAR ARRAY FROM
C                   IRREGULAR EXPERIMENTAL INPUT DATA READ IN.  CALLS
C                   SUBROUTINE SHEET. (EXPERIMENTAL DATA MAY
C                   BE SIMULATED, SEE 'SHEET')
C
C        MODE 3  -  UTILIZES RAW DATA TAKEN AT THE PROPER INTERVAL,
C                   OR PREVIOUSLY GENERATED, AND READ DIRECTLY INTO THE
C                   GARRAY.  CALLS SUBROUTINE READ.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
      COMMON /IO/ CMS,IN1,IN2,IN4
      DIMENSION G(IMAX,JMAX),GA(IMAX,JMAX)
      PIE=3.14159265358979 3
      HS=SIZE/2.
      IF (MODE.GT.3) MODE=1
      RIMX=IMAX
      RJMX=JMAX
      DELR=SIZE/RIMX
      DELXI=2.*PIE/FCU
      IF (MODE.GT.1) GO TO 2
      DO 1 J=1,JMS
      RJ=J
      XI=(RJ-.5)*DELXI-PIE
      J2=J+2*(JMS-J)
      J3=J+JMAX/2
      J4=J2+JMAX/2
```

```
      DO 1 I=1,IMS
      RI=I
      II=IMAX+1-I
      R=(RI-.5)*DELR-HS
      CALL GOLF (R,XI,GIJ,NOF,DGN,NUMB)
      G(I,J)=GIJ
      IF (ISYM.EQ.2) G(II,J)=GIJ
      IF (ISYM.EQ.2) GO TO 1
      G(II,J3)=GIJ
      IF (JSYM.EQ.0) GO TO 1
      G(I,J2)=GIJ
      G(II,J4)=GIJ
1     CONTINUE
      GO TO 4
2     IF (MODE.GT.2) GO TO 3
      CALL SHEET (G,GA,XO,YO,PHISYM,NOF)
      GO TO 4
3     CALL READ (Z,XO,YO,PHISYM,NOF,IMAX,JMAX,G)
4     IF (DGN.GE.2) WRITE (6,39)
      RETURN
39    FORMAT ('   GARRAY RETURNS')
      END
C000006
C


      SUBROUTINE GOLF (R,XI,GIJ,NOF,DGN,NUMB)
C
C   GOLF COMPUTES THE FUNCTION G(R,XI) FOR A PARTICULAR LINE OF SIGHT
C   FROM A KNOWN FUNCTION CONTAINED IN SUBROUTINE FUNCT.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      ZERO=0.
      LMAX=IMAX*3
      RLMAX=LMAX
      DELXP=SIZE/RLMAX
      SXI=SIN(XI)
      CXI=COS(XI)
      DELXS=DELXP*CXI
      DELYS=DELXP*SXI
      XP=DELXP*.5-SIZE/2.
      XS=XP*CXI-R*SXI
      YS=XP*SXI+R*CXI
      GIJ=0.
      DO 1 L=1,LMAX
      RL=L
      CALL FUNCT(XS,YS,F,NOF,DGN,NUMB)
      GIJ=GIJ+F
```

```
      XS=XS+DELXS                                                       SUB00850
1     YS=YS+DELYS                                                       SUB00860
      IF (GIJ.NE.0.) GIJ=GIJ*DELXP*BOX                                  SUB00870
      IF ((SD.EQ.0.).OR.(NUMB.EQ.1)) GO TO 2                            SUB00880
      IF (DGN.GE.3) WRITE (6,28) IX                                     SUB00890
      CALL GAUSS (IX,SD,ZERO,RV)                                        SUB00900
      GIJ=GIJ+RV                                                        SUB00910
2     IF (DGN.GE.3) WRITE (6,29) R,XI,GIJ                               SUB00920
      RETURN                                                            SUB00930
29    FORMAT ('    R=',F8.3,',    XI=',F8.3,',    GIJ=',F8.3)           SUB00940
28    FORMAT ('    GAUSS, IX=',I8)                                      SUB00950
      END                                                               SUB00960
C000007                                                                 SUB00970
C                                                                       SUB00980

      SUBROUTINE FUNCT (XS,YS,F,NOF,DGN,NUMB)                           SUB00990
C                                                                       SUB01000

CP67USERID  1095BOXJ
C
C   FUNCT EVALUATES AS INPUT FUNCTION AT POSITION (X,Y) IN THE TEST     SUB01020
C   SECTION COORDINATE SYSTEM.  NOF IDENTIFIES THE EQUATION USED.       SUB01030
C                                                                       SUB01040
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z   SUB01050
      CCMMON /EQPARA/ A,B,C,D,E,P,Q,S,T,U,V,W,RO,RA,NO,NA,N1,N2         SUB01060
      DIMENSION RO(101),RA(101)                                         SUB01070
      AA=A                                                              SUB01080
      BB=B                                                              SUB01090
      CC=C                                                              SUB01100
      DD=D                                                              SUB01110
      EE=E                                                              SUB01120
      PP=P                                                              SUB01130
      IF (NUMB.LE.1) GO TO 50                                           SUB01140
      AA=S                                                              SUB01150
      BB=T                                                              SUB01160
      CC=U                                                              SUB01170
      DD=V                                                              SUB01180
      EE=W                                                              SUB01190
      PP=Q                                                              SUB01200
50    PIE=3.141592653589793                                             SUB01210
      HS=SIZE/2.                                                        SUB01220
      R=SQRT(XS**2+YS**2)/HS                                            SUB01230
      F=0.                                                              SUB01240
      IF (R.GT.1.) GO TO 11                                             SUB01250
      IF (NOF.LE.0) GO TO 11                                            SUB01260
C                                                                       SUB01270
C   1.  AXISYMMETRIC GAUSSIAN:                                          SUB01280
```

```
1     IF (NOF.GT.1) GO TO 2                                            SUB01290
      F=AA*EXP(-1.*(R*HS/BB)**2)                                       SUB01300
      GO TO 11                                                         SUB01310
C                                                                      SUB01320
C   2.  ADJUSTABLE RECTANGULAR STEP FUNCTION:                          SUB01330
2     IF (NOF.GT.2) GO TO 3                                            SUB01340
      F=PP                                                             SUB01350
      IF ((ABS(XS-DD).LE.BB).AND.(ABS(YS-EE).LE.CC)) F=AA              SUB01360
      GO TO 11                                                         SUB01370
C                                                                      SUB01380
C   3.  DISPLACABLE ELLIPTICAL GAUSSIAN:                               SUB01390
3     IF (NOF.GT.3) GO TO 4                                            SUB01400
      F=AA*EXP(-1.*(((XS-DD)/BB)**2+((YS-EE)/CC)**2))                  SUB01410
      GO TO 11                                                         SUB01420
C                                                                      SUB01430
C   4.  CONSTANT:                                                      SUB01440
4     IF (NOF.GT.4) GO TO 5                                            SUB01450
      F=AA                                                             SUB01460
      GO TO 11                                                         SUB01470
C                                                                      SUB01480
C   5.  ADJUSTABLE AND DISPLACABLE ELLIPTIC RAMP FUNCTION:             SUB01490
5     IF (NOF.GT.5) GO TO 6                                            SUB01500
      RBC=SQRT(((XS-DD)/BB)**2+((YS-EE)/CC)**2)                        SUB01510
      F=0.                                                             SUB01520
      IF (RBC.LT.1.) F=AA*((1.-RBC)**PP)                               SUB01530
      GO TO 11                                                         SUB01540
C                                                                      SUB01550
C   6.  DISPLACABLE ELLIPTIC STEP FUNCTION:                            SUB01560
6     IF (NOF.GT.6) GO TO 7                                            SUB01570
      RBC=SQRT(((XS-DD)/BB)**2+((YS-EE)/CC)**2)                        SUB01580
      F=0.                                                             SUB01590
      IF (RBC.LT.1.) F=AA                                              SUB01600
      GO TO 11                                                         SUB01610
C                                                                      SUB01620
C   7.  CIRCULAR COSINE-SQUARED FUNCTION OF BB MAXIMA:                 SUB01630
7     IF (NOF.GT.7) GO TO 8                                            SUB01640
      F=AA*COS((2.*BB-1.)*PIE*R/2.)**2                                 SUB01650
      GO TO 11                                                         SUB01660
C                                                                      SUB01670
C   8.  NUMERICAL FUNCTION:  REQUIRES AN INPUT ARRAY READ IN BY        SUB01680
C       SUBROUTINE FREAD; N FOLLOWED BY N POINT VALUES. (101 MAX)      SUB01690
C       A CONSTANT VALUE AA IS ADDED TO THE FUNCTION.                  SUB01700
8     IF (NOF.GT.8) GO TO 9                                            SUB01710
      IF (NUMB.LE.1) N=NO                                              SUB01720
      IF (NUMB.GT.1) N=NA                                              SUB01730
      NM=N-1                                                           SUB01740
      NMM=N-2                                                          SUB01750
      RN=N                                                             SUB01760
```

```
      RI=R*(RN-1.)+1.
      IR=INT(RI)
      RIR=FLOAT(IR)
      DI=RI-RIR
      IF (NUMB.LE.1) F=RO(IR)
      IF (NUMB.GT.1) F=RA(IR)
      IF ((IR.NE.N).AND.(NUMB.LE.1)) F=F+DI*(RO(IR+1)-RO(IR))
      IF ((IR.NE.N).AND.(NUMB.GT.1)) F=F+DI*(RA(IR+1)-RA(IR))
      F=F*AA+BB
      GO TO 11
C
C   9.  SPECIAL FUNCTION:  MAY BE WRITTEN FOR THE OCCASION AND
C         INSERTED IN SUBROUTINE SPFUN
9     IF (NOF.GT.9) GO TO 10
      CALL SPFUN (XS,YS,F)
      GO TO 11
C
C   EQUATIONS NO. 10 AND BEYOND ARE SET TO ZERO.
10    F=0.
C
C
11    IF (DGN.GE.4) WRITE  (6,99) XS,YS,F
99    FORMAT ('    XS=',F8.3,',    YS=',F8.3,',    F=',F8.3)
      RETURN
      END
C000008
C


      SUBROUTINE SPFUN (XS,YS,F)
C
C   SPFUN IS A SPECIAL ROUTINE FOR EQ'N NO. 9.  ANY FUNCTION MAY BE
C   ENTERED.
C
      COMMON /EQPARA/ A,B,C,D,E,P,Q,S,T,U,V,W,RO,RA,NO,NA,N1,N2
      DIMENSION RO(101),RA(101)
      F=0.
      IF ((ABS(XS).LE.B).AND.(ABS(YS).LE.C))  F=A
      RETURN
      END
C000009
C
```

```
      SUBROUTINE SHEET (G,D,XO,YO,PHISYM,NOF)
C
C    SHEET READS IRREGULARLY SPACED VALUES OF THE LINE INTEGRAL, AS
C    OBTAINED FROM HOLOGRAPHIC INTERFEROGRAMS.  THE INTEGRAL LINES MAY BE
C    DEFINED EITHER BY GRID INTERCEPT POSITIONS, OR BY ANGLE AND RADIUS
C    ABOUT THE CENTER OF THE LABORATORY COORDINATE SYSTEM CENTER.  LINES
C    MUST BE ENTERED IN CONSECUTIVE ORDER FROM LOWEST (NEG.) TO HIGHEST
C    (POS.) RADIUS.  DATA MAY BE SIMULATED BY SPECIFYING NCODE=1,
C    FOLLOWED BY APERATURE POSITIONS FOR A FUNCTION NUMBER IN 'SUBFUNCT'.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      COMMON /SYM/ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM
      COMMON /IO/ CMS,IN1,IN2,IN4
      DIMENSION G(IMAX,JMAX),D(IMAX,JMAX),XI(303),RR(303)
      DIMENSION XG(303),XD(303),YG(303),YD(303),XY(303)
      NAR=303
      PIE=3.141592653589793
      MPIE=-PIE
      TPIE=2.*PIE
      MPIT=MPIE/2.
      PIET=PIE/2.
C    ZERO THE ARRAYS:
      DO 1 J=1,JMAX
      DO 1 I=1,IMAX
      G(I,J)=0.
1     D(I,J)=0.
      DO 2 I=1,NAR
      XG(I)=0.
      XD(I)=0.
      YG(I)=0.
      YD(I)=0.
      XY(I)=0.
      XI(I)=0.
2     RR(I)=0.
C    READ THE BASIC DATA:
      IF (CMS.EQ.1.) REWIND 1
      READ (IN1,59) NOF,NCODE
      READ (IN1,58) Z,XO,YO,PHISYM,XMX,XMN,YMX,YMN
      READ (IN1,59) JM
      RIMX=IMAX
      DR=SIZE/RIMX
      R=(-DR-SIZE)/2.
      RZO=SQRT(XO**2+YO**2)
      GAM=ATANM(YO,XO)
      TP=3-ISYM
      BT=JSYM
      DAN=PIE*TP/BT
      HS=SIZE/2.
```

```
      MXY=1
      IF((XMX.NE.0.).OR.(XMN.NE.0.).OR.(YMX.NE.0.).OR.(YMN.NE.0.))MXY=0
      IF (MXY.EQ.1) GO TO 3
      XMX=XO+HS
      YMX=YO+HS
      XMN=XO-HS
      YMN=YO-HS
C    COMMENCE THE READ AND FILL LOOP:
3     DO 12 J=1,JM
      READ (IN1,59)IM
      MN=0
      XH=0.
      YH=0.
C    READ THE LINES, DETERMINE CODE, CALCULATE RADIUS & ANGLE FOR CODE 1:
      DO 5 I=1,IM
      IF(NCODE.LE.0)READ(IN1,58) XD(I),YD(I),XG(I),YG(I),D(I,J),RR(I),
     1XI(I),XY(I)
      IF(NCODE.GE.1)CALL SIM(XD(I),YD(I),XG(I),YG(I),RR(I),XI(I),
     1XY(I),XO,YO,PHISYM,XMX,XMN,YMX,YMN,NOF,I,IM)
      IF (XY(I).EQ.3.) GO TO 5
      IF((XD(I).NE.0.).OR.(YD(I).NE.0.)) XY(I)=1.
      IF((XG(I).NE.0.).OR.(YG(I).NE.0.)) XY(I)=1.
      IF(((RR(I).NE.0.).OR.(XI(I).NE.0.)).AND.(XY(I).EQ.0.)) XY(I)=2.
      IF((XY(I).EQ.0.).AND.(D(I,J).NE.0.)) XY(I)=2.
      IF (XY(I).NE.1.) GO TO 4
      DEN=SQRT((XG(I)-XD(I))**2+(YG(I)-YD(I))**2)
      IF (DEN.EQ.0.) XY(I)=4.
      IF (XY(I).EQ.4.) GO TO 4
      RR(I)=((XO-XD(I))*(YG(I)-YD(I))-(XG(I)-XD(I))*(YO-YD(I)))/DEN
      XI(I)=ATANM((YG(I)-YD(I)),(XG(I)-XD(I)))
      XIM=XI(I)
4     IF (XY(I).EQ.2.) XIM=XI(I)
      XIN=XIM
      IF(XY(I).EQ.2.) RR(I)=RR(I)+RZO*SIN(GAM-XI(I))
5     CONTINUE
C    COMPUTE MAX AND MIN ANGLE INDEXES FOR APERATURE POSITION LOCATION:
      DO 6 I=1,IM
      IF((XY(I).NE.1.).OR.(XY(I).NE.2.)) GO TO 6
      IF (XI(I).GT.XIM) XIM=XI(I)
      IF (XI(I).GT.XIM) IMT=I
      IF (XI(I).LT.XIN) XIN=XI(I)
      IF (XI(I).LE.XIN) INT=I
6     CONTINUE
C    DETERMINE APERATURE LOCATION:
      LPR=0
      XID=XI(IMT)-XI(INT)
      IF(ABS(XID).LT..00001) LPR=1
      XIH=(XI(IMT)+XI(INT))/2.
```

```
      RRH=10000.                                                        SUB03130
      XH=RRH*COS(XIH)                                                   SUB03140
      YH=RRH*SIN(XIH)                                                   SUB03150
      IF (LPR.EQ.1) GO TO 7                                             SUB03160
      YTX=-RR(IMT)*SIN(XI(IMT))-YO                                      SUB03170
      YTN=-RR(INT)*SIN(XI(INT))-YO                                      SUB03180
      XTX=RR(IMT)*COS(XI(IMT))-XO                                       SUB03190
      XTN=RR(INT)*COS(XI(INT))-XO                                       SUB03200
      UA=TAN(XI(IMT))                                                   SUB03210
      UC=TAN(XI(INT))                                                   SUB03220
      UB=YTX-UA*XTX                                                     SUB03230
      UD=YTN-UC*XTN                                                     SUB03240
      XH=(UD-UB)/(UA-UC)                                                SUB03250
      YH=XH*UA+UB                                                       SUB03260
      RRH=SQRT((XH-XO)**2+(YH-YO)**2)                                   SUB03270
      XIH=ATANM((YH-YO),(XH-XO))                                        SUB03280
7     CONTINUE                                                          SUB03290
C   FILL THE ANGLE AND RADIUS FOR ANY CODE 3 OR 4 LINES:                SUB03300
      DO 9 I=1,IM                                                       SUB03310
      IF(XY(I).NE.3.) GO TO 8                                           SUB03320
      BAS=SQRT(RRH**2-RR(I)**2)                                         SUB03330
      XI(I)=XIH-ATANM(RR(I),BAS)                                        SUB03340
      GO TO 9                                                           SUB03350
8     XI(I)= ATANM((YH-YD(I)),(XH-XD(I)))                               SUB03360
      RR(I)=RRH*SIN(XI(I)-XIH)                                          SUB03370
9     CONTINUE                                                          SUB03380
C   ANGLES AND RADII ARE NOW FILLED FOR ALL POINTS IN THIS LINE.        SUB03390
C   VACATE THE SET OF VECTORS TO BE USED AS TEMPORARY STORAGE:          SUB03400
      DO 10 I=1,IM                                                      SUB03410
      XD(I)=0.                                                          SUB03420
      YD(I)=0.                                                          SUB03430
      XG(I)=0.                                                          SUB03440
      YG(I)=RR(I)                                                       SUB03450
      XY(I)=D(I,J)                                                      SUB03460
      RR(I)=0.                                                          SUB03470
      D(I,J)=0.                                                         SUB03480
10    XI(I)=0.                                                          SUB03490
C   CONVERT THE LINE TO REGULAR RADII USING INTERPOLATION:              SUB03500
      RR(1)=R+DR                                                        SUB03510
      CALL SPLINE(YG,XY,IM,RR(1),D(1,J))                                SUB03520
      DO 11 I=2,IMAX                                                    SUB03530
      RI=I                                                              SUB03540
      RR(I)=R+DR*RI                                                     SUB03550
11    CALL SPLINN(YG,XY,IM,RR(I),D(I,J))                                SUB03560
C   GENERATE THE VECTOR OF ANGLES FOR THIS COLUMN AND STORE IN G ARRAY: SUB03570
      DO 12 I=1,IMAX                                                    SUB03580
      BAS=SQRT(RRH**2-RR(I)**2)                                         SUB03590
      G(I,J)=XIH-ATANM(RR(I),BAS)                                       SUB03600
```

```
      YG(I)=0.
      D(I,J)=XY(I)
12    XY(I)=0.
C   COLUMNS ARE NOW ALL REGULARLY FILLED.
C   NEXT,INTERPOLATE EACH ROW REGULARLY OVER THE ANGLES.
      DO 23 I=1,IMAX
C   EXPAND THE DATA TO 2 SETS TO ESTABLISH SMOOTH INTERPOLATION.
      JM3=3*JM
      II=IMAX+1-I
      IF (JSYM.NE.0) GO TO 14
      DO 13 J=1,JMS
      J2=J+JMS
      J3=J2+JMS
      XD(J)=D(I,J)
      XD(J2)=D(II,J)
      XD(J3)=D(I,J)
      XG(J)=G(I,J)-TPIE-PHISYM
      XG(J2)=G(II,J)-PIE-PHISYM
13    XG(J3)=G(I,J)-PHISYM
      GO TO 16
14    DO 15 J=1,JMS
      J1=JMS+1-J
      J2=JMS+J
      J3=JM3+1-J
      XD(J1)=D(I,J)
      XD(J2)=D(II,J)
      XD(J3)=D(II,J)
      XG(J1)=G(I,J)-2.*(G(I,J)-PHISYM)-PIE-PHISYM
      XG(J2)=G(II,J)-PIE-PHISYM
15    XG(J3)=G(II,J)+2.*(DAN+PHISYM-G(II,J))-PIE-PHISYM
16    CONTINUE
      JM2=2*JMS
      JP=JMS/2
      DO 17 J=1,JM2
      XD(J)=XD(J+JP)
17    XG(J)=XG(J+JP)
      JJS=JM2+1
      DO 18 J=JJS,JM3
      XD(J)=0.
18    XG(J)=0.
C   FIND THE SMALLEST ANGLE
      XY(1)=1.
      SA=XG(1)
      DO 19 J=1,JM2
      IF (XG(J).GE.SA) GO TO 19
      SA=XG(J)
      XY(1)=J
19    CONTINUE
```

```
C     FIND THA MAX ANGLE IN THE ROW:
         XY(JM2)=JM2
         SB=XG(JM2)
         DO 20 J=1,JM2
         IF (XG(J).LE.SB) GO TO 20
         SB=XG(J)
         XY(JM2)=J
20       CONTINUE
C     DETERMINE THE ORDER OF INCREASING ANGLE IN THE ROW
         SB=XG(JM2)
         JJ=2
21       JSA=XY(JJ-1)
         SA=XG(JJ-1)
         JTS=0
         DO 22 J=1,JM2
         IF (XG(J).LE.SA) GO TO 22
         IF (XG(J).GT.SB) GO TO 22
         SB=XG(J)
         XY(JJ)=J
         JTS=1
22       CONTINUE
         IF (JTS.EQ.0) JM2=JJ
         JJ=JJ+1
         IF (JJ.LE.JM2) GO TO 21
         DO 23 J=1,JM2
         JX=XY(J)
23       YD(J)=XD(JX)
C     INTERPOLATE:
         DXI=2.*PIE/FCU
         XI(J)=DXI/2.-PIE-PHISYM
         CALL SPLINE (XG,YD,JM2,XI(J),G(I,J))
         DO 24 J=2,JMS
         XI(J)=XI(J-1)+DXI
24       CALL SPLINN (XG,YD,JM2,XI(J),G(I,J))
         DO 25 J=1,JMS
         XIJ=XI(J)
         XU=XMX
         IF ((XIJ.GE.0.).AND.(XIJ.LT.PIE)) XU=XMN
         YU=YMN
         IF ((XIJ.GE.MPIT).AND.(XIJ.LT.PIT)) YU=YMX
         XL=XMN
         IF ((XIJ.GE.0.).AND.(XIJ.LT.PIE)) XL=XMX
         YL=YMX
         IF ((XIJ.GE.MPIT).AND.(XIJ.LT.PIT)) YL=YMN
         SXIJ=SIN(XIJ)
         CXIJ=COS(XIJ)
         RMN=(XO-XL)*SXIJ-(YO-YL)*CXIJ
         RMX=(XO-XU)*SXIJ-(YO-YU)*CXIJ
```

```
      DO 25 I=1,IMAX
      IF (RR(I).LT.RMN) G(I,J)=0.
      IF (RR(I).GT.RMX) G(I,J)=0.
25    CONTINUE
C   EXPAND SYMMETRY SECTOR INTO AN ORTHOGONAL INTERVAL.
      IF (ISYM.EQ.2) GO TO 27
      DO 26 J=1,JMS
      J2=JMAX/2+1-J
      J3=JMAX/2+J
      J4=JMAX+1-J
      DO 26 I=1,IMAX
      II=IMAX+1-I
      G(I,J2)=G(I,J)
      G(II,J3)=G(I,J)
26    G(II,J4)=G(I,J)
      RETURN
C   FOR EVEN SYMMETRY, AVERAGE THE GARRAY COLUMNS.
27    IMS=(2*IMAX+1)/2
      DO 28 J=1,JMAX
      DO 28 I=1,IMS
      II=IMAX+1-I
      GST=(G(I,J)+G(II,J))/2.
      G(I,J)=GST
28    G(II,J)=GST
      RETURN
59    FORMAT (5I5)
58    FORMAT (10F7.3)
      END
C000010
C


      FUNCTION ATANM(Y,X)
C
C   COMPUTES THE ARCTAN OF Y/X BETWEEN -P1 AND +P1.
C
      PIE=3.141592653589793
      P12=PIE/2.
      ATANM=SIGN(PI2,Y)
      IF(X.NE.0.) ATANM=ATAN(Y/X)
      IF(X.GE.0.) RETURN
      IF(Y.GE.0.) ATANM=PIE+ATANM
      IF(Y.LT.0.) ATANM=-PIE+ATANM
      RETURN
      END
C000011
C
```

```
      SUBROUTINE SIM (XD,YD,XG,YG,D,R,XI,XY,XO,YO,PS,XM,XN,YM,YN,I,IM,   SUB05020
     1DG,NF)                                                              SUB05030
C    SIM SIMULATES THE FRINGE NUMBER DATA ONE WOULD OBTAIN FROM THE       SUB05040
C    HOLOGRAPHIC INTERFEROGRAM PROCESS FOR A KNOWN FUNCTION AS            SUB05050
C    CONTAINED IN SUBROUTINE FUNCT.  THE GRID BOX DIMENSIONS MUST         SUB05060
C    EXCEED THE INVERSION CIRCLE SIZE, AND APERATURE POINTS SPECIFIED     SUB05070
C    MUST FALL BETWEEN XI=-40 DEGREES, AND XI=+130 DEGREES.               SUB05080
C                                                                         SUB05090
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALF,SIZ,EPS,MOD,BOX,SD,IX,Z         SUB05100
      COMMON /IO/ CMS,IN1,IN2,IN4                                         SUB05110
      READ (IN1,29) XH,YH                                                 SUB05120
      ZER=0.                                                              SUB05130
      RIM=IM                                                              SUB05140
      RI=I                                                                SUB05150
      DX=(XM-XN)/(RIM-1.)                                                 SUB05160
      DY=(YM-YN)/(RIM-1.)                                                 SUB05170
      X1=XM-(RIM-1.)*DX                                                   SUB05180
      Y1=YM-(RIM-1.)*DY                                                   SUB05190
      XI1=ATANM (YH-Y1,XH-X1)-PS                                          SUB05200
      RRH=SQRT((XH-XO)**2+(YH-YO)**2)                                     SUB05210
      XIO=ATANM(YH-YO,XH-XO)-PS                                           SUB05220
      R1=RRH*SIN(XI1-XIO)                                                 SUB05230
      IF ((I.GT.1).AND.(I.LT.IM)) GO TO 1                                 SUB05240
      IF (ABS(R1).LT.SIZ/2.) R1=SIGN(SIZ/2.,R1)                           SUB05250
      XY=3.                                                               SUB05260
      GO TO 2                                                             SUB05270
1     CALL GOLF (R,XI,D,NR,ZER,ZER)                                       SUB05280
      XD=XN                                                               SUB05290
      YD=YN                                                               SUB05300
      IF (XH.NE.X1) YD=Y1-(X1-XN)*(YH-Y1)/(XH-X1)                         SUB05310
      IF (YS.GE.YN) GO TO 2                                               SUB05320
      YD=YN                                                               SUB05330
      IF (YH.NE.Y1) XD=X1-(Y1-YN)*(XH-X1)/(YH-Y1)                         SUB05340
2     RETURN                                                              SUB05350
29    FORMAT (10F7.3)                                                     SUB05360
      END                                                                 SUB05370
                                                                          SUB05380
C000012                                                                   SUB00010
C                                                                         SUB00020

      SUBROUTINE FREAD (NO,RO,NF,ZZ)                                      SUB00030
C                                                                         SUB00040
C    FREAD READS THE NUMERIC ARRAY WHICH IS USED FOR EQUATION 8 OF        SUB00050
C    SUBROUTINE FUNCT.  FIRST CARD IS NUMBER OF POINTS (N.GE.1),          SUB00060
C    FOLLOWED BY ONE POINT PER CARD.                                      SUB00070
C                                                                         SUB00080
      DIMENSION RO(101)                                                   SUB00090
```

```
      READ (NF,89) NO,ZZ
      WRITE(6,90) NO,ZZ
      DO  10  I=1,NO
      READ(NF,88)  RO(I)
      WRITE(6,88)  RO(I)
   10 CONTINUE
89    FORMAT (I5,F9.3)
   88 FORMAT(F8.5)
   90 FORMAT(1X,I5,F9.3)
      RETURN
      END
C000013
C


      SUBROUTINE GPRINT (G,NUMB)
C
C   GPRINT PRINTS THE DATA ARRAY 'G'  WHICH WAS INPUT TO
C   THE PROGRAM IN SUBROUTINE GARRAY.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      DIMENSION G(IJMX)
      DIMENSION X(15)
      DATA HYP,VERT/1H-,1H|/
      IF (NUMB.EQ.1) WRITE (6,99) MODE,Z
      IF (NUMB.EQ.2) WRITE (6,92) Z
      JM2=JMAX/2
      RIMAX=IMAX
      RJMAX=JMAX
      DX=SIZE/RIMAX
      DXI=360./RJMAX
C   INTRVL SETS THE NUMBER OF TERMS PRINTED PER LINE. IF IT IS ALTERED,
C   ONE MUST ALSO REDIMENSION X AND ALTER FORMATS 98, 97, AND 95.
      INTRVL=15
      IB=1
      IT=IB+INTRVL-1
1     IF (IT.GT.IMAX) IT=IMAX
      IBT=IT-IB+1
      WRITE (6,98) (II,II=IB,IT)
      DO 2 I=1,IBT
      RI=IB-1+I
2     X(I)=-SIZE/2.+(RI-.5)*DX
      LM=7*IBT+1
      WRITE (6,97) (X(I),I=1,IBT)
      WRITE (6,96) (HYP,L=1,LM),VERT
      JMH=JM2+1
      DO 3 J=JMH,JMAX
      RJ=J
```

```
      XI=-180.+DXI*(RJ-.5)                                              SUB00510
      IGB=(J-1)*IMAX+IB                                                 SUB00520
      IGT=IGB-IB+IT                                                     SUB00530
3     WRITE (6,95) J,XI,(G(L),L=IGB,IGT)                                SUB00540
      WRITE (6,94) (HYP,L=1,LM),VERT                                    SUB00550
      IB=IB+INTRVL                                                      SUB00560
      ITOLD=IT                                                          SUB00570
      IT=IT+INTRVL                                                      SUB00580
      IF (ITOLD.LT.IMAX) GO TO 1                                        SUB00590
      WRITE (6,93)                                                      SUB00600
99    FORMAT (1H1//'   THE ARRAY OF INPUT DATA (G), OBTAINED BY GARRAY, SUB00610
     1MODE ',I1,', FOR Z='F7.3,' CM:')                                  SUB00620
98    FORMAT (///11X,'I= ',15I7)                                        SUB00630
97    FORMAT (11X,'X=   ',15F7.3)                                       SUB00640
96    FORMAT ('    J=      XI= |',115A1)                                SUB00650
95    FORMAT (2X,I3,F9.2,'|',15F7.3)                                    SUB00660
94    FORMAT (14X,'|',115A1)                                            SUB00670
93    FORMAT (///5X)                                                    SUB00680
92    FORMAT (1H1//'   THE ADD-ON FUNCTION GARRAY FOR Z=',F7.3,' CM:')  SUB00690
      RETURN                                                            SUB00700
      END                                                               SUB00710
C000014                                                                 SUB00720
C                                                                       SUB00730


      SUBROUTINE GPUNCH (Z,XO,YO,PHS,NOF,IMX,JMX,G)                     SUB00740
C                                                                       SUB00750
C   GPUNCH PUNCHES OUT THE FIRST NON-SYMMETRIC PORTION OF GARRAY        SUB00760
C   (OR WRITES IT ON FILE 7 IN CMS VERSION)                             SUB00770
C                                                                       SUB00780
      COMMON /SYM/ ISM,JSM,MSM,FCU,IMS,JMS,QSM                          SUB00790
      DIMENSION G(IMX,JMX)                                              SUB00800
      WRITE (7,39) NOF,IMX,JMX,ISM,JSM,IMS,JMS                          SUB00810
      WRITE (7,38) ((G(I,J),I=1,IMS),J=1,JMS)                           SUB00820
39    FORMAT(10I5)                                                      SUB00830
38    FORMAT(10F7.3)                                                    SUB00840
      RETURN                                                            SUB00850
      END                                                               SUB00860
C000015                                                                 SUB00870
C                                                                       SUB00880


      SUBROUTINE READ (Z,XO,YO,PHISYM,NOF,IMAX,JMAX,G)                  SUB00890
C                                                                       SUB00900
C   READS THE NON-SYMMETRIC PORTION OF THE GARRAY AND EXPANDS IT TO AN  SUB00910
C   ORTHOGONAL SET. NOTE, INSURE SUFFICIENT DIMENSIONS IN MAIN PROGRAM. SUB00920
C                                                                       SUB00930
      COMMON /SYM/ ISYM,JSYM,MSYM,FCU,IMS,JMS,QSYM                      SUB00940
```

```
      COMMON /IO/ CMS,IN1,IN2,IN4
      DIMENSION G(IMAX,JMAX)
      READ (IN1,39) NOF,IMAX,JMAX,ISYM,JSYM,IMS,JMS
      READ (IN1,38) Z,XO,YO,PHISYM
      READ (IN1,38) ((G(I,J),I=1,IMS),J=1,JMS)
      WRITE(6,37) NOF,Z,XO,YO,PHISYM,IMAX,JMAX,JSYM
      RJMX=JMAX
      MSYM=JSYM
      IF ((MSYM.EQ.0).OR.(MSYM.GT.JMAX)) MSYM=1
      FCU=ISYM*JSYM*JMAX
      IF (JSYM.GT.JMAX) FCU=JMAX
      QSYM=FCU/RJMX
      DO 4 J=1,JMS
      IF(ISYM.EQ.1) GO TO 2
      DO 1 I=1,IMS
      II=IMAX+1-I
1     G(II,J)=G(I,J)
      GO TO 4
2     J2=JMAX/2+1-J
      J3=JMAX/2+J
      J4=JMAX+1-J
      DO 3 I=1,IMAX
      II=IMAX+1-I
      G(I,J2)=G(I,J)
      G(II,J3)=G(I,J)
3     G(II,J4)=G(I,J)
4     CONTINUE
39     FORMAT(10I5)
38    FORMAT (10F7.3)
37    FORMAT(///'  MODE 3 READS GARRAY DIRECTLY:   NOF='I4/'      Z='F7.3
     1,',   XO=',F7.3,',   YO=',F7.3,',   PHISYM=',F7.3/'     IMAX=',I4,
     2',   JMAX=',I4,',   JSYM=',I4///)
      RETURN
      END
C000016
C

      SUBROUTINE MAP (IM,JM,A,N,Z,BAND)
C
C   MAP CALLS SUBROUTINE MIMPII AND PLOTS A CONTOUR MAP OF THE ARRAY
C
      DIMENSION A(IM,JM),T(24)
      DATA BL/1H /
      DO 1 I=1,24
1     T(I)=BL
      ICON=1
      IF(BAND.LT.0.) ICON=0
```

```
      IF(BAND.LT.0.) BAND=-BAND
      AMIN=0.
      IJT=0
      AZ=1.
      BZ=0.
      WRITE(6,49) N,Z
      CALL MTMPII (A,IM,JM,T,BAND,AZ,BZ,AMIN,IJT,ICON)
49    FORMAT (1H1//'  THE FUNCTION SURFACE, TEST NO.'I3,'  Z='F5.3//)
      RETURN
      END
C000017
C

      SUBROUTINE GPLOT (G,GA,JMS)
C
C   GPLOT PRINTS A ROUGH PLOT OF THE LINE INTEGRAL FUNCTIONS IN GARRAY.
C
      COMMON IMAX,JMAX,IIMX,JJMX,IJMX,ALPHA,SIZE,EPS,MODE,BOX,SD,IX,Z
      CCMMON /TAB/ INDEX(7),JSYM,ISYM
      COMMON /TAB2/ IPT,KPT,LPT,MPT,REST(5)
      DIMENSION G(IMAX,JMAX),GA(IMAX,JMAX),ROW(101)
      DIMENSION A(201),B(101),C(201),D(101)
      JM=101
      DATA BL,PL,ST,DH,EX/1H ,1H+,1H*,1H-,1HX/
      JMS2=JMAX/2+1
      IF(ISYM.EQ.2)JMS2=1
      JMS3=JMS2+JMS-1
      DO 8 J=JMS2,JMS3
      WRITE(6,67) (ST,I=1,120)
      DO 1 I=1,IMAX
      A(I)=G(I,J)
1     C(I)=GA(I,J)
      AS=.5
      BS=.0
      CALL INTERP (A,IMAX,AS,B,JM,BS)
      CALL INTERP (C,IMAX,AS,D,JM,BS)
      WRITE (6,69) J
      BIG=0.
      SMALL=0.
      DO 2 I=1,IMAX
      IF(A(I).GT.BIG) BIG=A(I)
      IF(C(I).GT.BIG) BIG=C(I)
      IF(A(I).LT.SMALL) SMALL=A(I)
2     IF(C(I).LT.SMALL) SMALL=C(I)
      RANGE=BIG-SMALL
      RINK=RANGE/80.
      TOP=BIG+RINK
```

```
      CEN=BIG                                                            SUB01870
      BOT=BIG-RINK                                                       SUB01880
      KC=0                                                               SUB01890
      DO 7 K=1,41                                                        SUB01900
      IC=0                                                               SUB01910
      DO 6 I=1,101                                                       SUB01920
      ROW(I)=BL                                                          SUB01930
      IF((I.EQ.1).OR.(I.EQ.51).OR.(I.EQ.101))ROW(I)=PL                   SUB01940
      IF((K.EQ.1).OR.(K.EQ.41)) ROW(I)=PL                                SUB01950
      IF((TOP.GE.0.).AND.(BOT.LE.0.)) ROW(I)=DH                          SUB01960
      IF(IC.EQ.5) GO TO 3                                                SUB01970
      GO TO 4                                                            SUB01980
3     IC=0                                                               SUB01990
      IF(KC.EQ.10) ROW(I)=PL                                             SUB02000
4     IF(KPT.LE.2) GO TO 5                                               SUB02010
      IF((D(I).LE.TOP).AND.(D(I).GE.BOT)) ROW(I)=ST                      SUB02020
5     IF((B(I).LE.TOP).AND.(B(I).GE.BOT)) ROW(I)=EX                      SUB02030
6     IC=IC+1                                                            SUB02040
      IF(KC.EQ.5) KC=0                                                   SUB02050
      IF(KC.NE.0) WRITE (6,65) (ROW(I),I=1,101)                          SUB02060
      IF(KC.EQ.0) WRITE (6,68) CEN,(ROW(I),I=1,101)                      SUB02070
      TOP = TOP-2.*RINK                                                  SUB02080
      CEN=CEN-2.*RINK                                                    SUB02090
      BOT=BOT-2.*RINK                                                    SUB02100
7     KC=KC+1                                                            SUB02110
8     WRITE (6,66) (ST,I=1,120)                                          SUB02120
69    FORMAT (//'       J=',I3///)                                       SUB02130
68    FORMAT (1X,F8.3,1X,101A1)                                          SUB02140
67    FORMAT (1H1,//121A1//)                                             SUB02150
66    FORMAT (///121A1///)                                               SUB02160
65    FORMAT (10X,101A1)                                                 SUB02170
      RETURN                                                             SUB02180
      END                                                                SUB02190
C000018                                                                  SUB02200
C                                                                        SUB02210

      SUBROUTINE INTERP (A,IM,AS,B,JM,BS)                                SUB02220
C                                                                        SUB02230
C    INTERP CONVERTS A REGULAR VECTOR A OF IM POINTS TO A REGULAR VECTOR SUB02240
C    B OF JM POINTS.  OS=.5 FOR A VECTOR WITH POINTS DEFINED IN THE      SUB02250
C    CENTER OF THE INTERVAL, AS AND BS ARE THE % OF AN INTERVAL FROM THE SUB02260
C    EDGE OF THE FIELD TO THE FIRST POINT (.0 OR .5 FOR EDGE OR CENTER   SUB02270
C    DEFINED POINTS)                                                     SUB02280
C                                                                        SUB02290
      DIMENSION A(IM),B(JM)                                              SUB02300
      RIM=IM                                                             SUB02310
      RJM=JM                                                             SUB02320
```

```
      RAT=(RIM-1.+2.*AS)/(RJM-1.+2.*BS)
      DO 2 I=1,JM
      BI=I
      AI=RAT*(BI+BS)-AS
      IA=AI
      F=AI-FLOAT(IA)
      IF((IA.EQ.0).OR.(IA.EQ.JM)) GO TO 1
      B(I)=A(IA)+F*(A(IA+1)-A(IA))
      GO TO 2
1     IF(IA.EQ.0) B(I)=A(1)*(F-AS)/(1.-AS)
      IF(IA.EQ.JM) B(I)=A(JM)*F/(1.-AS)
2     CONTINUE
      RETURN
      END
C000019
C     ---------------------------------------------------------------------
C
C         SUBROUTINE SPLINE
C
C         PURPOSE
C             PROVIDES INTERPOLATED VALUE USING "CUBIC SPLINE FITTING"
C
C         USAGE
C             FIRST CALL TO SUBROUTINE:
C             CALL SPLINE(X,Y,M,XINT,YINT)
C
C             SUBSEQUENT CALLS:
C             CALL SPLINN(X,Y,M,XINT,YINT)
C
C         DESCRIPTION OF PARAMETERS
C             X:      MONOTONICALLY INCREASING ABSCISSA ARRAY
C             Y:      ONE-FOR-ONE CORRESPONDING ORDINATE ARRAY
C             M:      NUMBER OF X AND Y VALUES SUPPLIED < OR = 300
C             XINT:   VALUE OF ABSCISSA FOR WHICH CORRESPONDING ORDINATE
C                     IS TO BE INTERPOLATED (OR EXTRAPOLATED)
C             YINT:   INTERPOLATED (OR EXTRAPOLATED) ORDINATE VALUE
C
C
C         REMARKS
C             IF SPECIFIED X FALLS OUTSIDE OF RANGE, AN EXTRAPOLATED
C             VALUE WILL BE SUPPLIED
C
C         SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C             SUBROUTINE SPLICO IS INCLUDED IN SUBROUTINE SPLIN PACKAGE
C
C         MATHEMATICAL METHOD
C             UPON FIRST ENTRY TO SPLIN, A CALL TO SPLICO IS MADE TO
```

```
C           DETERMINE THE COEFFICIENTS TO BE USED IN PERFORMING THE    SPL00340
C           INTERPOLATIONS. SEARCH FOR BRACKETING ABSCISSA VALUES IS   SPL00350
C           ALWAYS MADE FROM THE REFERENCE LAST USED IN INTERPOLATING. SPL00360
C                                                                      SPL00370
C        REFERENCE                                                     SPL00380
C           PENNINGTON, RALPH H.,"INTRODUCTORY COMPUTER METHODS AND    SPL00390
C           NUMERICAL ANALYSIS", THE MACMILLAN COMPANY, NEW YORK, 1965 SPL00400
C                                                                      SPL00410
C     -----------------------------------------------------------------SPL00420
C                                                                      SPL00430

      SUBROUTINE SPLINE(X,Y,M,XINT,YINT)                               SPL00440
      DIMENSION X(M),Y(M),C(4,300)                                     SPL00460
      CALL SPLICO(X,Y,M,C)                                             SPL00470
      K=1                                                              SPL00480
      ENTRY SPLINN(X,Y,M,XINT,YINT)                                    SPL00490
    3 IF(XINT-X(1)) 70,1,2                                             SPL00500
   70 K=1                                                              SPL00510
      GO TO 7                                                          SPL00520
    1 YINT=Y(1)                                                        SPL00530
      RETURN                                                           SPL00540
    2 IF(XINT-X(K+1))6,4,5                                             SPL00550
    4 YINT=Y(K+1)                                                      SPL00560
      RETURN                                                           SPL00570
    5 K=K+1                                                            SPL00580
      IF(M-K) 71,71,3                                                  SPL00590
   71 K=M-1                                                            SPL00600
      GO TO 7                                                          SPL00610
    6 IF(XINT-X(K))13,12,11                                            SPL00620
   12 YINT=Y(K)                                                        SPL00630
      RETURN                                                           SPL00640
   13 K=K-1                                                            SPL00650
      GO TO 6                                                          SPL00660
    7 PRINT 101,XINT                                                   SPL00670
  101 FORMAT(8H0XINT = E18.9,32H, OUT OF RANGE FOR INTERPOLATION)      SPL00680
   11 YINT=(X(K+1)-XINT)*(C(1,K)*(X(K+1)-XINT)**2+C(3,K))              SPL00690
      YINT=YINT+(XINT-X(K))*(C(2,K)*(XINT-X(K))**2+C(4,K))             SPL00700
      RETURN                                                           SPL00710
      END                                                              SPL00720

      SUBROUTINE SPLICO(X,Y,M,C)                                       SPL00730
      DIMENSION X(M),Y(M),C(4,300),D(300),P(300),E(300),A(300,3),B(300),SPL00750
     1Z(300)                                                           SPL00760
      MM=M-1                                                           SPL00770
      DO 2 K=1,MM                                                      SPL00780
      D(K)=X(K+1)-X(K)                                                 SPL00790
```

```
      P(K)=D(K)/6.
    2 E(K)=(Y(K+1)-Y(K))/D(K)
      DO 3 K=2,MM
    3 B(K)=E(K)-E(K-1)
      A(1,2)=-1.-D(1)/D(2)
      A(1,3)=D(1)/D(2)
      A(2,3)=P(2)-P(1)*A(1,3)
      A(2,2)=2.*(P(1)+P(2))-P(1)*A(1,2)
      A(2,3)=A(2,3)/A(2,2)
      B(2)=B(2)/A(2,2)
      DO 4 K=3,MM
      A(K,2)=2.*(P(K-1)+P(K))-P(K-1)*A(K-1,3)
      B(K)=B(K)-P(K-1)*B(K-1)
      A(K,3)=P(K)/A(K,2)
    4 B(K)=B(K)/A(K,2)
      Q=D(M-2)/D(M-1)
      A(M,1)=1.+Q+A(M-2,3)
      A(M,2)=-Q-A(M,1)*A(M-1,3)
      B(M)=B(M-2)-A(M,1)*B(M-1)
      Z(M)=B(M)/A(M,2)
      MN=M-2
      DO 6 I=1,MN
      K=M-I
    6 Z(K)=B(K)-A(K,3)*Z(K+1)
      Z(1)=-A(1,2)*Z(2)-A(1,3)*Z(3)
      DO 7 K=1,MM
      Q=1./(6.*D(K))
      C(1,K)=Z(K)*Q
      C(2,K)=Z(K+1)*Q
      C(3,K)=Y(K)/D(K)-Z(K)*P(K)
    7 C(4,K)=Y(K+1)/D(K)-Z(K+1)*P(K)
      RETURN
      END

C000020
C
C        ******************************************************************
C
C        SUBROUTINE MTMPII
C
C        PURPOSE
C                 MTMPII WILL PRODUCE,ON THE PRINTER, A CONTOUR MAP
C                 OF ANY SINGLE PRECISION TWO DIMENSIONAL ARRAY.
C
C        USAGE
C                 CALL MTMPII(Y,N,M,T,BND,AZ,BZ,AMIN,IJT,ICON)
C
```

```
C         DESCRIPTION OF PARAMETERS                                         MET00130
C                                                                           MET00140
C                   Y    - THE ARRAY TO BE CONTOURED. DIMENSIONED Y(N,M)MET00150
C                   N    - NUMBER OF ROWS IN Y.                             MET00160
C                   M    - NUMBER OF COLUMNS IN Y.                          MET00170
C                   T    - ARRAY FOR PLOT TITLE. REAL*4 T(24).              MET00180
C                   BND  - BANDWIDTH FOR THE CONTOURING. IF BND IS ZERO MET00190
C                          A BANDWIDTH WILL BE CALCULATED AS FOLLOWS        MET00200
C                          BND=(MAX(Y)-MIN(Y))/15.                          MET00210
C                   AZ   - A LINEAR TRANSFORMATION MAYBE PERFORMED ON       MET00220
C                          THE ARRAY Y OF THE FOLLOWING FORM AZ*Y+BZ.       MET00230
C                          IF AZ=0 THEN AZ WILL BE COMPUTED SUCH THAT       MET00240
C                          MAX(|MAX(Y)|,|MIM(Y)|) WILL BE LESS THAN 1,      MET00250
C                          AND BZ WILL BE LEFT AS INPUT.                    MET00260
C                   BZ   - SEE UNDER AZ                                     MET00270
C                   AMIN - THE LEVEL AT WHICH COUTOURING WILL BEGIN. IF MET00280
C                          AMIN > MIN(Y) THEN AMIN WILL BE CALCULATED       MET00290
C                          TO BE ,WITH REFERENCE AT ZERO, THE NEXT LOWERMET00300
C                          CONTOUR LEVEL FROM MIN(Y) AS DETERMINED BY       MET00310
C                          BND.                                             MET00320
C                   IJT  - IF IJT=0 AMIN WILL BE CALCULATED AS DESCRIBEDMET00330
C                          ABOVE.                                           MET00340
C                   ICON - IF ICON=0 NO CONTOURING WILL BE DONE BUT THE MET00350
C                          ARRAY Y WILL BE PRINTED IN THE PLOT FORMAT.      MET00360
C                                                                           MET00370
C         REMARKS                                                           MET00380
C                   MTMPII REQUIRES A PRINTER WITH 132 PRINT POSITIONS.     MET00390
C                   IF NECESSARY THE MAP WILL BE SEGMENTED COLUMNWISE.      MET00400
C                   THE ROWS AND COLUMNS ARE NUMBERED ALONG THE EDGES SOMET00410
C                   THAT A SEGMENTED MAP MAYBE EASILY JOINED TOGETHER.      MET00420
C                   ONLY THREE SIGNIFICANT FIGURES WILL BE PRINTED AT       MET00430
C                   EACH POINT. THE POSITION OF THE FIRST SIGNIFICANT       MET00440
C                   DIGIT WILL BE DETERMINED BY MAX(|MAX(Y)|,|MIN(Y)|).     MET00450
C                   THE PLOT WILL BE PRODUCED ON A INCH INCH GRID. IT       MET00460
C                   WILL BE ASSUMED THAT THE SPACING BETWEEN POINTS IN      MET00470
C                   BOTH DIRECTIONS IS THE SAME AND EQUAL FOR ALL POINTSMET00480
C                                                                           MET00490
C         SUBROUTINES REQUIRED                                              MET00500
C                   NONE                                                    MET00510
C                                                                           MET00520
C         METHOD                                                            MET00530
C                   THE CONTOUR LEVELS ARE DETERMINED BY SIMPLE LINEAR      MET00540
C                   INTERPOLATION FROM THE FOUR SURROUNDING PIONTS.         MET00550
C                                                                           MET00560
C         ******************************************************************MET00570
C                                                                           MET00580
C                                                                           MET00590
C    MTMPII SUBROUTINE FOR ONE-INCH GRID SPACING                            MET00600
```

```
C    OAKES CODE 5105    15 JAN 69
C
      SUBROUTINE MTMPII(Y,N,M,T,BND,AZ,BZ,AMIN,IJT,ICON)
      REAL*4 IH,KG,ITJZ
      DIMENSION A(140),B(140),C(140),D(140),IH(20),Y(N,M),TP(10),TPX(10)
     Z  ,TPM(10),XMT(10),BTM(10),BTX(10),BT(10),KG(10),T(24)
      DIMENSION E(140),F(140),G(140),H(140)
C
C
      DATA DUE/4H    /,EPL/4H+   /,EMI/4H-   /,IH/1H0,1H ,1H1,1H ,1H2,
     11H ,1H3,1H ,1H4,1H ,1H5,1H ,1H6,1H ,1H7,1H ,1H8,1H ,1H9,1H /,KG/
     21H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/,BLK/4H    /
C
      YMIN=Y(1,1)
      YMAX=Y(1,1)
      DO 20 I=1,M
      DO 10 J=1,N
      YMIN=AMIN1(YMIN,Y(J,I))
      YMAX=AMAX1(YMAX,Y(J,I))
   10 CONTINUE
   20 CONTINUE
      DELY=YMAX-YMIN
      IF(BND) 25,25,30
   25 BND=DELY/15.0
   30 IF (AMIN-YMIN) 31,31,32
   31 IF (IJT) 33,32,33
   32 PD=YMIN/BND
      PF=ABS(PD-INT(PD))
      IF (YMIN) 2,1,1
    1 AMIN=YMIN-PF*BND
      GO TO 33
    2 AMIN=YMIN-(1.0-PF)*BND
   33 AHLD=AZ
      IF(AZ) 55,35,55
   35 SM=AMAX1(ABS(YMIN),ABS(YMAX))
      NS=0
   40 NS=NS+1
      SM=10.0*SM
      IF(SM-1.0)40,50,45
   45 NS=NS-1
      SM=SM/10.0
      IF(SM-1.0)50,50,45
   50 AHLD=10.0**NS
   55 HBND=BND/2.0
      PRINT 70
      PRINT 6,T
```

```
    6 FORMAT(5X,24A4,//)
      PRINT 57,AHLD,BZ
   57 FORMAT(1H0,65HTHE FOLLOWING TRANSFORMATION WAS PERFORMED ON THE IN
     1PUT MATRIX    /5X,1H(,E12.5,8H*Y(I,J)+,E12.5,1H) //2X,73HAND THREE
     2 DIGITS TO THE RIGHT OF THE DECIMAL POINT ARE PRINTED IN THE MAP )
C
      PRINT 54,YMAX,YMIN
   54 FORMAT(/4X,5HYMAX=,E15.7,5X,5HYMIN=,E15.7)
      IF (ICON)5,58,5
    5 PRINT 11,BND
   11 FORMAT(2X,17HTHE BAND WIDTH IS,E12.5,6H UNITS //4X,14HCONTOUR LEVE
     1LS        )
      I=0
      YTOP=AMIN
      IF(ABS(YMIN-YMAX)-100.0*BND)53,53,58
   53 YB=YTOP
      YTOP=YTOP+BND
      I=I+1
      J=MOD(I,20)
      ITJZ=IH(J)
      IF(YB-YMAX)59,58,58
   59 PRINT 61,YB,YTOP,ITJZ
   61 FORMAT(/4X,E10.3,4H TO ,E10.3,2H =,1X,A1)
      GO TO 53
   58 NCCP=0
      NCP=0
   60 PRINT 70
   70 FORMAT(1H1)
      PRINT 6,T
      NLINE=0
      NCCP=NCP+1
      NCP = NCP + 13
   73 IF(NCP-M)80,80,75
   75 NCP=M
   80 CONTINUE
      J=-2
      NLINE=NLINE+1
      LLINE=N-NLINE+1
C SET UP HEADING
      IF(NCCP-1) 85,85,90
   85 J = - 1
   90 DO 100 I = 1,135
      A(I)=BLK
      B(I)=BLK
      H(I) = BLK
  100 CONTINUE
  110 DO 160 L=NCCP,NCP
      J = J+8
```

```
      KI=L
      IF(KI-100) 130,120,120
  120 LL=KI/100
      A(J)=KG(LL+1)
      KI=KI-100*LL
      GO TO 135
  130 A(J)=KG(1)
  135 J=J+1
      IF(KI-10) 150,140,140
  140 LL=KI/10
      A(J)=KG(LL+1)
      KI=KI-10*LL
      GO TO 155
  150 A(J)=KG(1)
  155 J=J+1
      A(J)=KG(KI+1)
  160 CONTINUE
C SETUP FIRST ROW OF ARRAY
      GO TO 260
  170 NLINE=NLINE+1
      LLINE=N-NLINE+1
      IF(NLINE-N) 180,180,380
  180 DO 190 I=1,135
      A(I)=BLK
      B(I)=BLK
      C(I)=BLK
      D(I)=BLK
      E(I) = BLK
      F(I) = BLK
      G(I) = BLK
      H(I) = BLK
  190 CONTINUE
      IF (ICON)195,260,195
  195 NCY=NCCP-1
      J=4
      IF(NCY)200,200,210
  200 J=5
  210 NCY=NCY+1
      IF(NCY-NCP) 220,220,260
  220 IF(NCY-M) 230,260,260
  230 NLINE = NLINE - 1
      YD1 = Y(NLINE,NCY) - Y(NLINE+1,NCY)
      YD2=Y(NLINE,NCY+1)-Y(NLINE+1,NCY+1)
      TP(1) = Y(NLINE,NCY)-0.125*YD1
      TPX(1)=Y(NLINE,NCY)-0.250*YD1
      TPM(1)=Y(NLINE,NCY)-0.375*YD1
      XMT(1)=Y(NLINE,NCY)-0.500*YD1
      BTM(1)=Y(NLINE,NCY)-0.625*YD1
```

```
      BTX(1)=Y(NLINE,NCY)-0.750*YD1
      BT (1)=Y(NLINE,NCY)-0.875*YD1
      TP (10)=Y(NLINE,NCY+1)-0.125*YD2
      TPX(10)=Y(NLINE,NCY+1)-0.250*YD2
      TPM(10)=Y(NLINE,NCY+1)-0.375*YD2
      XMT(10)=Y(NLINE,NCY+1)-0.500*YD2
      BTM(10)=Y(NLINE,NCY+1)-0.625*YD2
      BTX(10)=Y(NLINE,NCY+1)-0.750*YD2
      BT (10)=Y(NLINE,NCY+1)-0.875*YD2
      NLINE = NLINE + 1
      D1=0.1 * (TP (10)-TP (1))
      D2=0.1 * (TPX(10)-TPX(1))
      D3=0.1 * (TPM(10)-TPM(1))
      D4=0.1 * (XMT(10)-XMT(1))
      D5=0.1 * (BTM(10)-BTM(1))
      D6=0.1 * (BTX(10)-BTX(1))
      D7=0.1 * (BT (10)-BT (1))
      DO 240 I = 2,9
      TP (I)=TP (I-1)+D1
      TPX(I)=TPX(I-1)+D2
      TPM(I)=TPM(I-1)+D3
      XMT(I)=XMT(I-1)+D4
      BTM(I)=BTM(I-1)+D5
      BTX(I)=BTX(I-1)+D6
      BT (I)=BT (I-1)+D7
  240 CONTINUE
      DO 250 I=1,10
      J=J+1
      I1=MOD(IFIX((TP (I)-AMIN)/BND),20)+1
      I2=MOD(IFIX((TPX(I)-AMIN)/BND),20)+1
      I3=MOD(IFIX((TPM(I)-AMIN)/BND),20)+1
      I4=MOD(IFIX((XMT(I)-AMIN)/BND),20)+1
      I5=MOD(IFIX((BTM(I)-AMIN)/BND),20)+1
      I6=MOD(IFIX((BTX(I)-AMIN)/BND),20)+1
      I7=MOD(IFIX((BT (I)-AMIN)/BND),20)+1
      A(J)=IH(I1)
      B(J)=IH(I2)
      C(J)=IH(I3)
      D(J)=IH(I4)
      E(J)=IH(I5)
      F(J)=IH(I6)
      G(J)=IH(I7)
  250 CONTINUE
      GO TO 210
  260 NCY=NCCP-1
      J=-2
      IF(NCY) 265,265,270
  265 J=-1
```

```
      GO TO 330
  270 NCY=NCY+1
      IF(NCY-NCP) 280,280,310
  280 J=J+7
      THLD=AHLD*Y(NLINE,NCY)+BZ
      IF(THLD) 285,290,290
  285 H(J)=EMI
      GO TO 295
  290 H(J)=EPL
  295 NUM=INT(ABS(THLD-INT(THLD))*1000.0+0.5)
      NDS=100
      DO 300 KK=1,3
      J=J+1
      KI=NUM/NDS
      H(J)=KG(KI+1)
      NUM=NUM-KI*NDS
      NDS=NDS/10
  300 CONTINUE
      GO TO 270
  310 IF(NCP-M) 360,320,320
  320 IF(J-127)330,330,360
  330 J=J+3
      KI=NLINE
      IF(KI-100) 340,335,335
  335 LL=KI/100
      H(J)=KG(LL+1)
      KI=KI-100*LL
      GO TO 343
  340 H(J)=KG(1)
  343 J=J+1
      IF(KI-10) 350,345,345
  345 LL=KI/10
      H(J)=KG(LL+1)
      KI=KI-10*LL
      GO TO 355
  350 H(J)=KG(1)
  355 J=J+1
      H(J)=KG(KI+1)
      J=J-5
      IF(NCY-1) 270,270,360
  360 IF(NLINE-1)362,362,368
  362 PRINT 370,(A(I),I=1,132),(B(IP1),IP1=1,132),(H(IP2),IP2=1,132)
      GO TO 170
  368 PRINT 370,(A(I),I=1,132),(B(IP1),IP1=1,132),(C(IP2),IP2=1,132),
     1(D(IP3),IP3=1,132),(E(IP4),IP4=1,132),(F(IP5),IP5=1,132),
     2(G(IP6),IP6=1,132),(H(IP7),IP7=1,132)
  370 FORMAT(132A1)
      GO TO 170
```

```
  380 DO 390 I=1,135                                                MET02990
      A(I)=BLK                                                      MET03000
      B(I)=BLK                                                      MET03010
      C(I)=BLK                                                      MET03020
      D(I)=BLK                                                      MET03030
  390 CONTINUE                                                      MET03040
      J=-2                                                          MET03050
      IF(NCCP-1) 395,395,400                                        MET03060
  395 J=-1                                                          MET03070
  400 DO 430 L=NCCP,NCP                                             MET03080
      J=J+8                                                         MET03090
      KI=L                                                          MET03100
      IF(KI-100) 410,405,405                                        MET03110
  405 LL=KI/100                                                     MET03120
      C(J)=KG(LL+1)                                                 MET03130
      KI=KI-100*LL                                                  MET03140
      GO TO 412                                                     MET03150
  410 C(J)=KG(1)                                                    MET03160
  412 J=J+1                                                         MET03170
      IF(KI-10) 420,415,415                                         MET03180
  415 LL=KI/10                                                      MET03200
      C(J)=KG(LL+1)                                                 MET03210
      KI=KI-10*LL                                                   MET03220
      GO TO 422                                                     MET03230
  420 C(J)=KG(1)                                                    MET03240
  422 J=J+1                                                         MET03250
      C(J)=KG(KI+1)                                                 MET03260
  430 CONTINUE                                                      MET03270
      PRINT 370, (B(IP1),IP1=1,132),(C(IP2),IP2=1,132)              MET03280
      IF(NCP-M)60,500,500                                           MET03290
  500 RETURN                                                        MET03300
      END                                                           MET03300

C000021
C                                                                   GAUS  10
C     ..............................................................GAUS  20
C                                                                   GAUS  30
C        SUBROUTINE GAUSS                                           GAUS  40
C                                                                   GAUS  50
C        PURPOSE                                                    GAUS  60
C           COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A GIVEN GAUS  70
C           MEAN AND STANDARD DEVIATION                             GAUS  80
C                                                                   GAUS  90
C        USAGE                                                      GAUS 100
C           CALL GAUSS(IX,S,AM,V)                                   GAUS 110
C                                                                   GAUS 120
C        DESCRIPTION OF PARAMETERS                                  GAUS 130
```

```
C          IX -IX MUST CONTAIN AN ODD INTEGER NUMBER WITH NINE OR
C              LESS DIGITS ON THE FIRST ENTRY TO GAUSS. THEREAFTER
C              IT WILL CONTAIN A UNIFORMLY DISTRIBUTED INTEGER RANDOM
C              NUMBER GENERATED BY THE SUBROUTINE FOR USE ON THE NEXT
C              ENTRY TO THE SUBROUTINE.
C          S  -THE DESIRED STANDARD DEVIATION OF THE NORMAL
C              DISTRIBUTION.
C          AM -THE DESIRED MEAN OF THE NORMAL DISTRIBUTION
C          V  -THE VALUE OF THE COMPUTED NORMAL RANDOM VARIABLE
C
C       REMARKS
C          THIS SUBROUTINE USES RANDU WHICH IS MACHINE SPECIFIC
C
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C          RANDU
C
C       METHOD
C          USES 12 UNIFORM RANDOM NUMBERS TO COMPUTE NORMAL RANDOM
C          NUMBERS BY CENTRAL LIMIT THEOREM. THE RESULT IS THEN
C          ADJUSTED TO MATCH THE GIVEN MEAN AND STANDARD DEVIATION.
C          THE UNIFORM RANDOM NUMBERS COMPUTED WITHIN THE SUBROUTINE
C          ARE FOUND BY THE POWER RESIDUE METHOD.
C
C     ..................................................................
C

      SUBROUTINE GAJSS(IX,S,AM,V)
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END

C000022
C
C     ..................................................................
C
C       SUBROUTINE RANDU
C
C       PURPOSE
C          COMPUTES UNIFORMLY DISTRIBUTED RANDOM REAL NUMBERS BETWEEN
C          0 AND 1.0 AND RANDOM INTEGERS BETWEEN ZERO AND
C          2**31. EACH ENTRY USES AS INPUT AN INTEGER RANDOM NUMBER
```

```
C           AND PRODUCES A NEW INTEGER AND REAL RANDOM NUMBER.
C
C        USAGE
C           CALL RANDU(IX,IY,YFL)
C
C        DESCRIPTION OF PARAMETERS
C           IX - FOR THE FIRST ENTRY THIS MUST CONTAIN ANY ODD INTEGER
C                NUMBER WITH NINE OR LESS DIGITS. AFTER THE FIRST ENTRY,
C                IX SHOULD BE THE PREVIOUS VALUE OF IY COMPUTED BY THIS
C                SUBROUTINE.
C           IY - A RESULTANT INTEGER RANDOM NUMBER REQUIRED FOR THE NEXT
C                ENTRY TO THIS SUBROUTINE. THE RANGE OF THIS NUMBER IS
C                BETWEEN ZERO AND 2**31
C           YFL- THE RESULTANT UNIFORMLY DISTRIBUTED, FLOATING POINT,
C                RANDOM NUMBER IN THE RANGE 0 TO 1.0
C
C        REMARKS
C           THIS SUBROUTINE IS SPECIFIC TO SYSTEM/360 AND WILL PRODUCE
C           2**29 TERMS BEFORE REPEATING.  THE REFERENCE BELOW DISCUSSES
C           SEEDS (65539 HERE), RUN PROBLEMS, AND PROBLEMS CONCERNING
C           RANDOM DIGITS USING THIS GENERATION SCHEME.  MACLAREN AND
C           MARSAGLIA, JACM 12, P. 83-89, DISCUSS CONGRUENTIAL
C           GENERATION METHODS AND TESTS.  THE USE OF TWO GENERATORS OF
C           THE RANDU TYPE, ONE FILLING A TABLE AND ONE PICKING FROM THE
C           TABLE, IS OF BENEFIT IN SOME CASES.  65549 HAS BEEN
C           SUGGESTED AS A SEED WHICH HAS BETTER STATISTICAL PROPERTIES
C           FOR HIGH ORDER BITS OF THE GENERATED DEVIATE.
C           SEEDS SHOULD BE CHOSEN IN ACCORDANCE WITH THE DISCUSSION
C           GIVEN IN THE REFERENCE BELOW.  ALSO, IT SHOULD BE NOTED THAT
C           IF FLOATING POINT RANDOM NUMBERS ARE DESIRED,AS ARE
C           AVAILABLE FROM RANDU, THE RANDOM CHARACTERISTICS OF THE
C           FLOATING POINT DEVIATES ARE MODIFIED AND IN FACT THESE
C           DEVIATES HAVE HIGH PROBABILITY OF HAVING A TRAILING LOW
C           ORDER ZERO BIT IN THEIR FRACTIONAL PART.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           POWER RESIDUE METHOD DISCUSSED IN IBM MANUAL C20-8011,
C           RANDOM NUMBER GENERATION AND TESTING
C
C     ..................................................................
C
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY)5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*.4656613E-9
      RETURN
      END
```

LIST OF REFERENCES

1. Heflinger, L. O., Wuerker, R. F., and Brooks, R. E., "Holographic Interferometry," Journal of Applied Physics, Vol. 37, No. 2, pp. 642-649, February 1966.

2. Brooks R. E., Heflinger, L. O., and Wuerker, R. F., "9A9 - Pulsed Laser Holograms," IEEE Journal of Quantum Electronics, Vol. EQ-2, No. 8, pp. 275-299, August 1966.

3. Matulka, R. D., The Application of Holographic Interferometry to The Determination of Asymmetric Three-Dimensional Density Fields in Free Jet Flow, PhD Thesis, Naval Postgraduate School, June 1970.

4. Matulka, R. D., and Collins, D. J., "Determination of Three-Dimensional Density Fields From Holographic Interferograms," Journal of Applied Physics, Vol. 42, No. 3, pp. 1109-1119, March 1971.

5. Maldonado, C. D., Caron, A. P., and Olsen, H. N., "New Method for Obtaining Emission Coefficients from Emitted Spectral Intensities. Part I - Circularly Symmetric Light Sources," Journal of The Optical Society of America, Vol. 55, No. 10, pp. 1247-1254, October 1965.

6. Maldonado, C. D., and Olsen, H. N., "New Method for Obtaining Emission Coefficients from Emitted Spectral Intensities. Part II - Asymmetric Sources," Journal of The Optical Society of America, Vol. 56, No. 10, pp. 1305-1312, October 1966.

7. Jagota, R. C., The Application of Holographic Interferometry to The Determination of The Flow Field Around a Right Circular Cone At Angle of Attack, Aeronautical Engineer's Thesis, Naval Postgraduate School, December 1970.

8. Jagota, R. C., and Collins, D. J., "Finite Fringe Holographic Interferometry Applied to A Right Circular Cone At Angle of Attack," ASME Paper No. 72-APM-PP, to appear in the Journal of Applied Mechanics.

9. Heyer, R. W., Holographic Interferometry of The Flow Field Between A Fin and Flat Plate, Master's Thesis, Naval Postgraduate School, March 1972.

10. Liepmann, H. W., and Roshko, A., Elements of Gas Dynamics, p. 165, John Wiley and Sons, Inc., 1957.

11. Collier, R. J., and others, Optical Holography, sect. 10.8.4, Academic Press, 1971.

12. Sweeney, D. W., and Vest, C. M., "Reconstruction of Three-Dimensional Refractive Index Fields by Holographic Interferometry," Applied Optics, Vol. 11, No. 1, pp. 205-207, January 1972.

13. Van Houten, P. E., Analysis of Discontinuous Three-Dimensional Density Fields Using Holographic Interferometry, Aeronautical Engineer's Thesis, Naval Postgraduate School, December 1972.

14. Junginger, H. G., and van Haeringen, W., "Calculation of Three-Dimensional Refractive Index Fields Using Phase Integrals," to appear in Optics Communications.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor D. J. Collins, Code 57Co Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	LT Robert A. Kosakoski, USN The Arches, Apt #77 1235 Wildwood Avenue Sunnyvale, California 94086	1
5.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
6.	Dean of Research Naval Postgraduate School Monterey, California 93940	1

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP

3 REPORT TITLE

Application of Holographic Interferometry to Density Field Determination in Transonic Corner Flow

4 DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1972

5 AUTHOR(S) (First name, middle initial, last name)

Robert A. Kosakoski

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1972	120	14

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The successful application of holographic interferometry to the study of density fields around opaque bodies in wind tunnel experiments has been reported in the literature. The present report extends this technique to the study of the three-dimensional asymmetric flow fields encountered near the wing-fuselage junction of an aerodynamic model in the transonic flow regime. Finite fringe interferometry has been used to obtain fringe information about a partially transparent wind-body structure. A FORTRAN computer program was utilized to invert the fringe information and produce a plot of the density field around the model. The resulting asymmetric density field and shock wave structure are shown to be an accurate representation of the phenomena encountered in aerodynamic corner flow.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Holography						
Interferometry						
Transonic Flow						
Corner Flow						
Transparent Phase Object						
Aerodynamic Model						
Wing-Fuselage Junction						

DISTRIBUTION

1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library (Code 0212) Naval Postgraduate School Monterey, California 93940	2
3.	Commander Naval Air Systems Command Department of the Navy Washington, D. C. 20360	
	Attn: AIR 310	2
	AIR 604	2
4.	Chairman Department of Aeronautics Code 57 Naval Postgraduate School Monterey, California 93940	1
5.	Professor D. J. Collins Department of Aeronautics Code 57CO Naval Postgraduate School Monterey, California 93940	2
6.	Dean of Research Naval Postgraduate School Monterey, California 93940	2
7.	Dr. H. J. Mueller Naval Air Systems Command AIR 310 Department of the Navy Washington, D. C. 20360	2

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3 REPORT TITLE

Application of Holographic Interferometry to Density Field Determination in Transonic Corner Flow

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report NPS-57C072121A

5 AUTHOR(S) (First name, middle initial, last name)

D. J. Collins and R. A. Kosakoski

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1972	127	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. Dr. H. Mueller ONR Req. 00014-2-000033, 15 Sep 71	
c. NASC W. R. 2-6059, 12 Jul 71 d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Office of Naval Research and Naval Air Systems Command

13. ABSTRACT

The successful application of holographic interferometry to the study of density fields around opaque bodies in wind tunnel experiments has been reported in the literature. The present report extends this technique to the study of the asymmetric flow fields encountered near the wing-fuselage junction of an aerodynamic model in the transonic flow regime. Finite fringe interferometry has been used to investigate the three dimensional density field about a partially transparent wing-body structure. The resulting asymmetric density field and shock wave structure are shown to be an accurate representation of the phenomena encountered in aerodynamic corner flow.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Holography						
Interferometry						
Transonic Flow						
Corner Flow						
Transparent Phase OBject						
Aerodynamic Model						
Wing-Fuselage Junction						