TECHNICAL REPORT CR-RD-AS-90-3

EXTENSION AND UPDATING OF THE COMPUTER SIMULATION OF RANGE RELATIVE DOPPLER PROCESSING AND INVARIANT MAPPING FOR MM WAVE SEEKERS

VOLUME I

Robert J. Polge
Bassem R. Mahafza
Jong G. Kim
College of Engineering
University of Alabama in Huntsville
Huntsville, AL 35899

OCTOBER 1990

Prepared for:
Advanced Sensors Directorate
Research, Development, and Engineering Center

Contract No. DAAH01-87-D-0021

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898-5000

Approved for public release; distribution is unlimited.

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) CR-RD-AS-90-3, Vol I
6a. NAME OF PERFORMING ORGANIZATION College of Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Advanced Sensors Directorate RD&E Center
6c. ADDRESS (City, State, and ZIP Code) University of Alabama in Huntsville Huntsville, AL 35899	7b. ADDRESS (City, State, and ZIP Code) Commander U.S. Army Missile Command ATTN: AMSMI-RD-AS Redstone Arsenal, AL 35898-5253
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAH01-87-D-0021
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Extension and Updating of the Computer Simulation of Range Relative Doppler Processing for MM Wave Seekers and Invariant Mapping for MM Wave Seekers, Volume I

12. PERSONAL AUTHOR(S)
Robert J. Polge, Bassem R. Mahafza, Jong G. Kim

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 04/03/88 TO 12/31/88	October 1990	110

16. SUPPLEMENTARY NOTATION
Report consists of 3 volumes.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Doppler MM Wave Seekers Invariant Mapping

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report extends and updates the computer simulation of Range Relative Doppler Processing (RRDP) and Invariant Mapping for MM wave seekers, which was introduced in prior reports. A major difference between standard Synthetic Aperture and RRDP is that the return signal is heterodyned with respect to a synthetic signal corresponding to a fictitious scatterer at the center of the range cell. It follows that the implementation of RRDP depends critically on an accurate simulation of the round trip delay. For this purpose the round trip delay is expressed as a four dimensional third order Taylor expansion. The footprint reflectivity versus azimuth and range is transferred to an absolute x-y map using the Invariant Mapping Technique proposed earlier, which has been greatly improved.

The major contributions of this report are: (1) a more general geometry where the trajectory is not restricted to the vertical plane, (2) an exact simulation of the dish antenna, (3) a more exact signal synthesis that includes the scatterer height, (4) the development of an interactive subroutine which allows online updating of the data file, (Continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL A. H. Green	22b. TELEPHONE (Include Area Code) (205) 987-1728 22c. OFFICE SYMBOL AMSMI-RD-AS

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

BLOCK 19 (Cont'd): (5) a much better compensation for antenna gain and range attenuation, (6) a reduction through frequency interpolation of the peak reflectivity variation due to FF quantization, (7) a new, better, and more efficient invariant mapping algorithm, and (8) online graphics displays for the reflectivity maps (3-D and contours). Two main programs were developed, FRRDP and CRRDP. The first generates the footprint reflectivity map versus azimuth and range. The second program uses our Invariant Mapping Technique to create an absolute x-y reflectivity map. Examples show that the scatters are detected at the proper locations and that impulse invariance has been achieved over one footprint within $\pm$5.5%.

This report is concerned with the theory, formulas, Fortran implementation, and performance of the simulation. A separate User Manual is also available in Volume II [8].

PREFACE

The report for the work performed under Contract No. D.O. 0064, DAAH01–87–D–0021 consists of three volumes: (1) "Extension and Updating of the Computer Simulation of Range Relative Doppler Processing and Invariant Mapping for MM Wave Seekers", (2) "User Manual of the Range Relative Doppler Processing Simulation for MM Wave Seekers", and (3) "Computer Simulation of a Doppler Beam Sharpening Radar for MM Wave Seekers". The period of performance is April 12 to December 31, 1988.

The main objective of Volume I is to extend and update the MM wave computer simulation developed for Contract No. DAAH01–87–D–0021, D.O. 18, documented in the final report dated February 1988, and entitled "Increasing Azimuth and Elevation Resolution of MM Wave Seeker Systems Using Coherent or Noncoherent Range Relative Doppler Processing (RRDP) with Constant or Linear Frequency Modulation and Invariant Mapping".

Volume II is a User Manual for the computer simulation documented in Volume I. With this manual, MICOM personnel should be able to: (1) install the Fortran software on an IBM PC/compatible, (2) duplicate the results presented in this report, and (3) run the simulation for different clutter maps and targets.

The main objective of Volume III is to develop Doppler Beam Sharpening for MM wave seekers. In this application the geometry is significantly different. That is, the seeker follows a straight horizontal trajectory while the antenna performs a forward near circular scan perpendicular to the trajectory.

Robert J. Polge

Huntsville, Alabama

January 1989

ACKNOWLEDGEMENT

The Principal Investigator wishes to take this opportunity to acknowledge the cooperation and guidance of Mr. Jim Mullins and Mr. A. H. Green both of the U. S. Army Missile Command. Special Thanks are due to Mrs. Janie Parent for her efforts in typing the manuscript.

Table of Contents

LIST OF ILLUSTRATIONS

LIST OF TABLES

I. Introduction and Background

One of the most serious deficiencies in the existing MM wave seeker systems is the lack of azimuth resolution. As a result, detection and discrimination of targets in a clutter background is quite difficult. Azimuth resolution cannot be increased using standard SAR techniques because the MM wave nonlinear geometry magnifies the problems of range walk and doppler bias.

Two new digital processing techniques were developed [1,2] to extend the application of SAR processing to nonlinear trajectories. They are called "Range Relative Doppler Processing" and "Invariant Mapping". SAR Doppler Processing of each range cell was extended to nonlinear trajectories, by using as Doppler reference a fictitious scatterer at the center of the range cell. This resolves the problems of range walk and doppler bias. In [1] and part I of [2] the approach was to perform a time Taylor expansion with respect to the fictitious scatterer; the effect of azimuth on doppler being introduced later as a parabolic fit at the center and extremities of the range cell. The simulated return of the fictitious scatterer is used for heterodyning. After deramping, windowing, and FFT processing, the range cell is divided into trapezoidal azimuth cells corresponding to a constant frequency increment Δf, where Δf is the inverse of the observation interval, D_{ob}.

Part II of [2] extended the application of Range Relative Doppler Processing (RRDP) to linear frequency modulation (discrete or continuous). Another important contribution was that the round trip delay model, and therefore the signal model, includes explicitly not only time but also azimuth and elevation increments. The azimuth and elevation angular increments (μ_i,ε_i) for an arbitrary scatterer $\vec{C}_i$ in the jth range cell are measured with respect to the reference fictitious scatterer $\vec{C}_j^*$. It is clear that both the signal return and the doppler due to $\vec{C}_i$ are functions of (μ_i,ε_i), where the effect of ε_i is small in the case of a narrow range bin. The round trip delay τ_i to $\vec{C}_i$ is a function of (t,μ_i,ε_i) where t is relative time. A good approximation for the round trip delay $\tau_i(t,\mu_i,\varepsilon_i)$ is obtained using a three–dimensional Taylor

expansion about the fictitious scatterer state. As before, the fictitious scatterer at the center of the range cell serves as zero doppler reference. This means that the return signal must be heterodyned with a synthetic signal simulating the return of the fictitious scatterer, including the effect of frequency modulation when applicable. The remainder of the processing is unchanged: (1) A/D conversion, (2) deramping, (3) windowing, and (4) FFT processing. It was also shown that the stability requirement during the observation interval can be expressed as a bound on frequency drift which is practically attainable.

This report extends and updates the Range Relative Doppler Processing (RRDP) presented in [2], and also replaces the invariant mapping presented in [1] by a new and a much more efficient algorithm. Frequency modulation is not included in the simulation at this time. Most of our efforts on the RRDP simulation were concentrated on: (1) a more general geometry where the trajectory is not restricted to a vertical plane, (2) extending the RRDP simulation to include the effect of scatterer height, (3) entering interactively the parameters and the clutter data which means that only the changes need to be typed online, (4) an exact simulation for the dish antenna instead of a sin x /x pattern, (5) a reduction of impulse invariance over the entire footprint through frequency interpolation and a nearly perfect compensation for range and antenna gain. All the subroutines in the RRDP simulation were thoroughly tested and most of them were significantly improved.

Much efforts were expanded to develop a better and much more efficient invariant mapping technique. The heart of this technique is an algorithm which computes the area of overlap between a rectangle of the absolute grid and a trapezoidal azimuth cell from the current range cell. The mapping is area invariant which means that if a trapezoid overlaps with three rectangles the sum of the three overlaps is equal to the area of the trapezoid.

Using SURFER from Golden Software Inc. [7], we can display three–dimensional (3–D) maps or contour maps before and after mapping on our Compaq 386. In the previous report we were using Univac DISSPLA which is a much slower procedure. For two–

dimensional plots we use GRAPHER which is also from Golden Software.

The organization is as follows. The geometry and dish antenna simulation are discussed. Then, an extended Taylor expansion for the round trip delay $\tau_i(\hbar,t,\mu,\varepsilon)$, which includes scatterer height, is presented. Two simulation programs are developed. The first program, FRRDP generates a data file FRRDPG.GRD which stores reflectivity versus azimuth and range. The second program CRRDP uses the new invariant technique to generate a data file CRRDPG.GRD which stores reflectivity versus absolute x–y. The graphics software package SURFER transforms the reflectivity data files, FRRDPG.GRD and CRRDPG.GRD, into 3–D displays or contour plots. The simulation is illustrated by examples. This report is concerned with the theory, implementation and performance of the simulation. An User Manual is also available [8].

2. Geometry and System Parameters

Figure 2.1 shows the geometry for a footprint at central time t_c, where $\vec{q}(t_c)$ defines the center of the footprint. In our previous work the seeker was assumed to remain in a vertical plane, this limitation is now removed. The only assumption left is that the vector velocity remains constant within the observation interval D_{ob}

$$\vec{a}(t_c) = \{v_x, v_y, v_z\} \quad . \tag{2.1}$$

The vector $\vec{m}$ defines the position of the antenna at central time, and its rectangular coordinates with respect to the absolute origin $\vec{0}$ are

$$\vec{m} = \vec{a}(t_c) = \{x(\vec{m}) = d_x,\ y(\vec{m}) = d_y,\ h\}. \tag{2.2}$$

Figure 2.1 shows an absolute system of reference $\{\vec{0x},\vec{0y},\vec{0z}\}$ where $\vec{0}$ is a fixed point on the ground. The positions of the seeker, the current footprint center, and the scatterers are normally defined in absolute coordinates. For example, the footprint center is given by

$$\vec{q}^{*} = \vec{q}(t_c) = \{x_q, y_q, 0\}. \tag{2.3}$$

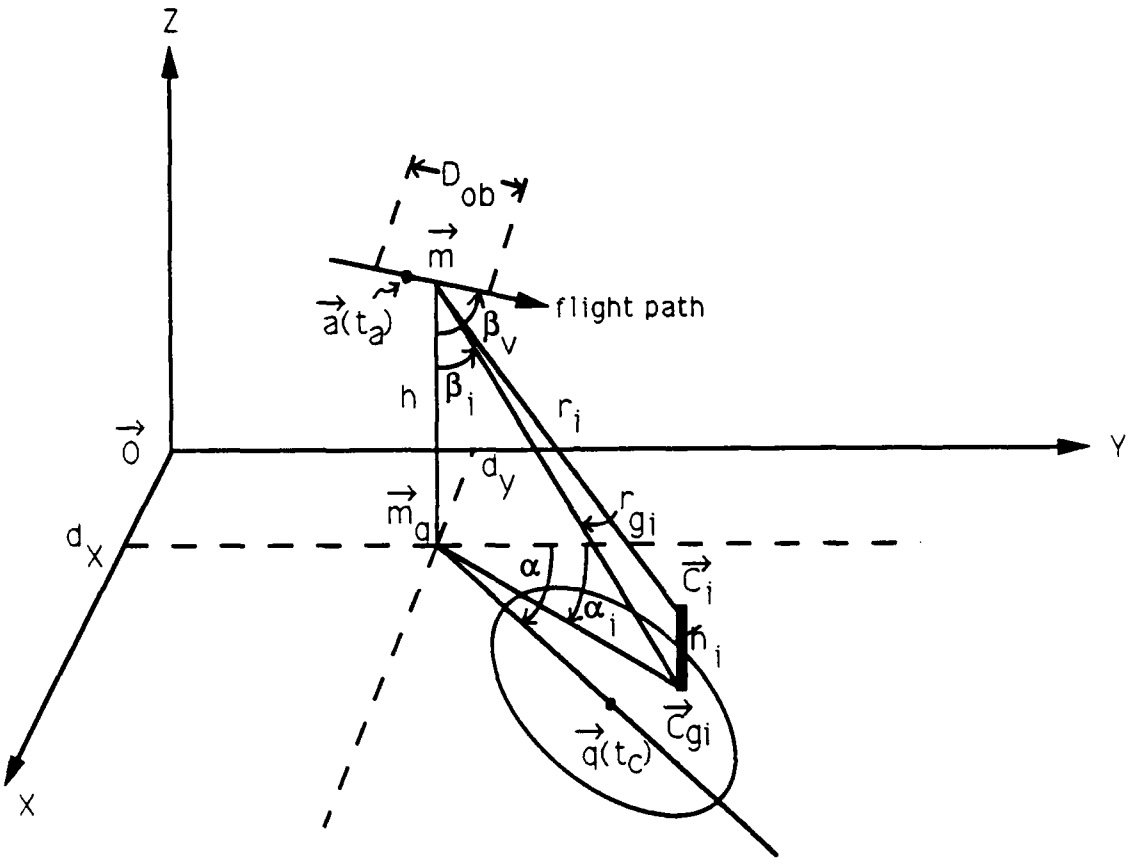

Figure 2.1 Geometry for a footprint at central time t_c:$t=t_a-t_c$.

However, within one observation interval, a spherical representation of a ground point with respect to the central seeker position $\vec{m}$ is more useful: {range, azimuth, elevation}, where azimuth is referred to $\overrightarrow{0y}$ and elevation is referred to the downward vertical. For example, the footprint center in the spherical representation is given by

$$\vec{q}^{*} - \vec{m} = \{r_q, \alpha_q = \alpha^{*} = \text{Angle}(\vec{q}^{*} - \vec{m}_g, \overrightarrow{0y}), \beta_q = \beta^{*} = \text{Angle}(\vec{m}_g - \vec{m}, \vec{q}^{*} - \vec{m})\}. \tag{2.4}$$

Similarly, let $\vec{C}_i$ be the ith scatterer and let $\vec{C}_{gi}$ be its ground projection, then

$$\vec{C}_{gi} - \vec{m} = \{r_{gi}, \alpha_i, \beta_i\} . \tag{2.5}$$

In the remainder of this report the azimuth and elevation angles of a ground point are always with reference to $\vec{m}$.

The positions of the antenna within D_{ob}, and of a scatterer $\vec{C}_i$ located h_i above the ground are easily written in absolute coordinates,

$$\vec{a}(t_a) = \vec{a}(t + t_c) = \{d_x + v_x t, d_y + v_y t, h + v_z t\} \tag{2.6a}$$

$$-\frac{D_{ob}}{2} \le t < \frac{D_{ob}}{2} \tag{2.6b}$$

where t_a denotes absolute time, and t is the relative time. The absolute coordinates of the ith scatterer $\vec{C}_i$ are

$$\vec{C}_i = \{x_i, y_i, h_i\} . \tag{2.7}$$

It will be shown that the height of the scatterer h_i significantly affects detection. The ground projection of the scatterer is defined in absolute coordinates as

$$\vec{C}_{gi} = \{x_i, y_i, 0\} . \tag{2.8}$$

It is obvious that a scatterer above the ground can be read as the sum of its ground projection and a vertical vector

$$\vec{C}_i = \vec{C}_{gi} + (0, 0, \hbar_i) \tag{2.9a}$$

$$\vec{C}_i = (\vec{C}_{gi} - \vec{m}) + (0, 0, \hbar_i) + \vec{m} \tag{2.9b}$$

where (2.5) or (2.8) are used depending on the preferred system of reference. Assuming that the scatterer projection $\vec{C}_{gi}$ is somewhere within the jth range cell, then $\vec{C}_i$ may actually appear in the $(j-1)$th range cell, if the height $\hbar_i$ is significant.

Within one range cell the reference is a fictitious scatterer at the center of the range cell. Figure 2.2 shows the jth range cell where the reference $\vec{C}_j^*$ is the flat fictitious scatterer at the center of the cell. This scatterer is defined with respect to $\vec{m}$ by

$$\vec{C}_j^* - \vec{m} = \{r_j^*, \alpha_j^*, \beta_j^*\} \ , \tag{2.10}$$

where $\alpha_j^* = \alpha^*$ because $\vec{C}_j^*$ is on the ground line of sight. Since we want to use $\vec{C}_j^*$ as a reference for the jth range cell, we define the azimuth and elevation of $\vec{C}_{gi}$ in terms of small incremental variations (μ_i, ε_i) about (α_j^*, β_j^*)

$$\alpha_i = \alpha^* + \mu_i \tag{2.11a}$$

$$\beta_i = \beta_j^* + \varepsilon_i. \tag{2.11b}$$

Given the position of the jth fictitious scatterer, the position of the ground projection $\vec{C}_{gi}$ of an arbitrary scatterer within the jth range cell is entirely defined by the angles μ_i and ε_i, which will be denoted as relative azimuth and relative elevation. The position of $\vec{C}_i$ follows from that of the ground point $\vec{C}_{gi}$ by an upward vertical motion $\hbar_i$.

As before, the footprint is defined as the 3dB contour illuminated by the antenna at central time. While the footprint moves within the observation interval, most of this motion is compensated by Range Relative Doppler Processing (RRDP). Given the antenna position $\vec{m}$ and using (2.3) or (2.4), the footprint is entirely defined by its center, that is $\{x_q, y_q\}$ or $\{\alpha^*, \beta^*\}$. The 3dB footprint is divided into range cells of width Δr, and for each range cell, the median range and the interval for μ $\{-\mu mx < \mu < \mu mx\}$ must be computed. Within the jth

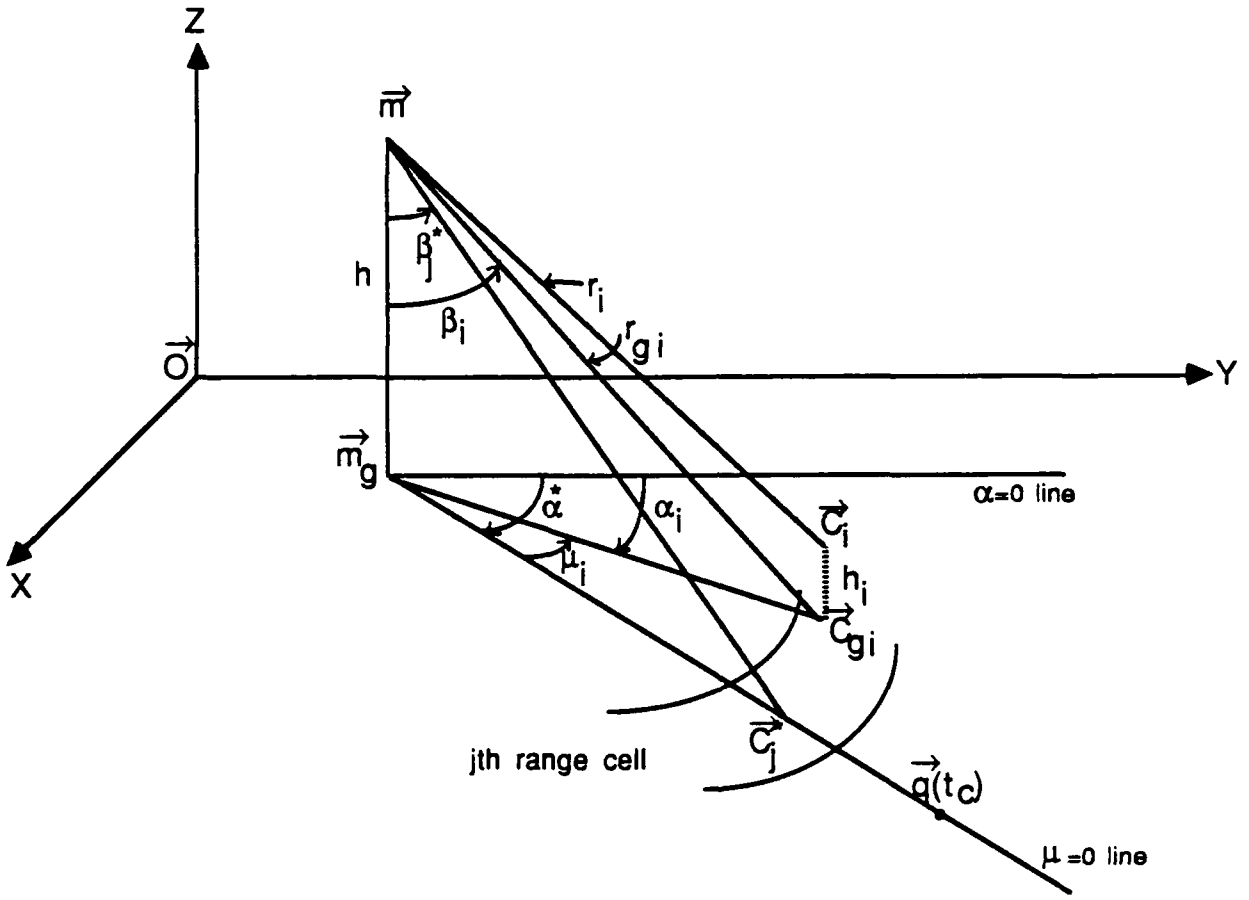

Figure 2.2 Fictitious scatterer and ith scatterer in the jth range cell: $\alpha_i = \alpha_j^* + \mu_i$.

range cell, ϵ is zero at medium range and maximum at minimum range $(r_j^* - \Delta r/2)$. The maximum values of (ϵ,μ), that is $(\epsilon mx, \mu mx)$ are used to determine which terms in the Taylor series expansion of round trip delay can be neglected. Appendix A contains all the formulas which define the relevant information in a footprint.

The ensemble of scatterers is denoted as clutter map. Details of the clutter map simulation are in section 7. There are two options to enter the ith scatterer in the simulation: $\{x_i,y_i,\hbar_i\}$ or $\{\mu_i,r_{gi},\hbar_i\}$. For each scatterer $\vec{C}_i$, defined in rectangular or spherical coordinates, one needs to compute the following: (1) r_i, range to the scatterer, (2) the index of the range cell containing $\vec{C}_i$, (3) the elevation, azimuth and their relative counterparts $(\alpha_i, \beta_i, \mu_i, \epsilon_i)$, and (4) the antenna gain in the direction $(\vec{C}_i - \vec{a}(t_a))$ at the start, middle, and end of the observation interval. The antenna gain at an arbitrary time within the observation interval follows by parabolic interpolation. This information together with the Taylor coefficients computed in Appendix D is all that is needed to simulate the signal. Appendix B contains all the formulas which define the relevant information for a scatterer $\vec{C}_i$.

In this report the seeker transmits simple RF pulses without linear frequency modulation. The system parameters are given in Table 2.1. The system operates in the forward looking SAR strip mode.

3. Dish Antenna Simulation

In our previous work the dish gain pattern was approximated as a sin x /x function. An exact dish gain pattern is now implemented. The gain pattern for a circular dish with diameter D at direction sine s = sinβ is given by [3,4],

$$G(sD/\lambda) = 2A \frac{J_1(\pi s D/\lambda)}{(\pi s D/\lambda)} \tag{3.1}$$

where J_1 is the Bessel function of the first kind of the first order, and A is the area of the circular dish. Figure 3.1 shows the normalized two–way dish pattern $G_n(w)$ in dB where w = $512sD/\lambda$.

Table 2.1: List of System Parameters

Parameter	Symbol	Value
Antenna Beamwidth	θ	43 mrad
Frequency	f	94 GHz
Range Bin	Δr	1 m
Total Transmitter Bandwidth	B	500 MHz
PRF or FM sweep Repetition Rate		50 KHz
CFAR 13 Cell Parametric Cell-averaging Line CFAR	Not implemented yet	
Log Receiver	Assumed ideal	
Seeker Velocity (input)	$\vec{a}\cdot(t_c)=\vec{q}^{\cdot}=\{v_x,v_y,v_z\}$ w.r.t. $\vec{0}$ $=\{\alpha_v,\beta_v,v\}$ w.r.t. $\vec{m}$	
Seeker Depression angle or Slant Angle (inputs)	δ_v β_v	20° 70°
Seeker Azimuth (input)	α_v	35°
Dwell time About Central Footprint (input)	D_{ob}	20 ms
Position of Seeker at Central time (input)	$\vec{a}(t_c)=\vec{m}=$ $\{d_x,d_y,h\}$	h ≈ 900 m
Center of Footprint (input)	$\vec{q}(t_c)=\vec{q}^{\cdot}=\{x_q,y_q,0\}$ w.r.t. $\vec{0}$ $=\{r_q,\alpha^*,\beta^*\}$ w.r.t. $\vec{m}$	

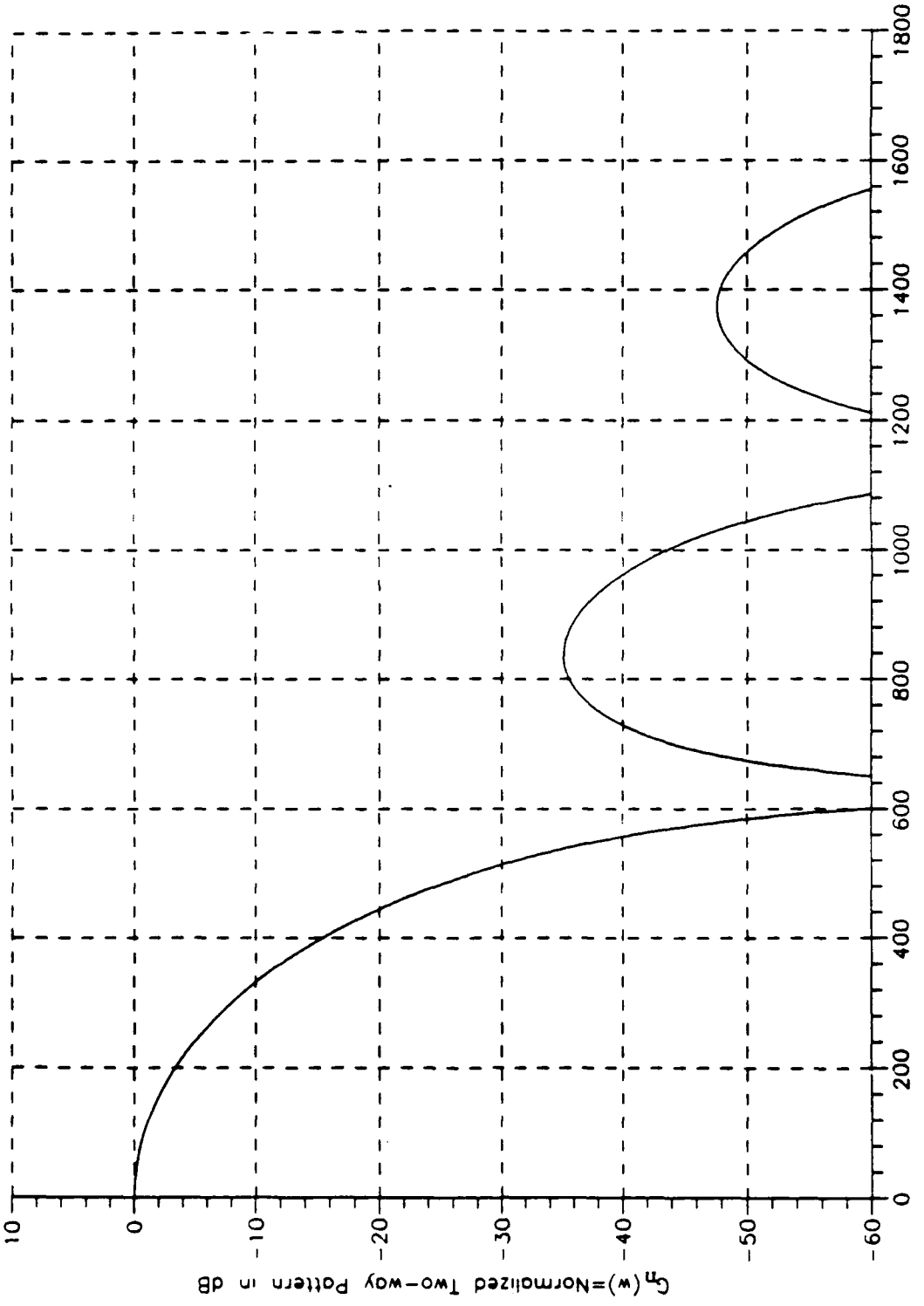

Figure 3.1. Normalized two-way dish pattern in dB versus w=512 sD/λ.

The one–way dish gain is computed using the function DISH(SINB, IDDISH, DDISH) where SINB = sinβ = the off boresight angle, IDDISH = {1,2,3} = {rectangular, circular Dish, sin x /x}, and DDISH = D/λ = dish diameter in λ units. For a beamwidth θ = 43 mrad, the two–way 3dB footprint corresponds to θ_H = 21.5 mrad. One can compute the equivalent diameter for IDDISH = 2 and 3 by solving

$$40 \log_{10}[\text{DISH}(\theta_H, 2, D_2)] = -3 \tag{3.3a}$$

$$40 \log_{10}[\text{DISH}(\theta_H, 3, D_3)] = -3. \tag{3.3b}$$

The results are: D_2 = 17.15094 and D_3 = 14.80884. Note that the one–way power gain is proportional to the modulus square of the electric field. The listings for the Function DISH and the associated Bessel function BESSJ1 are in Appendix C.

4. Taylor Expansion of Round Trip Delay τ_i

Assume that the ground projection of the ith scatterer within the jth cell is specified by

$$\vec{C}_{gi} - \vec{m} = \{r_{gi}, \alpha_i = \alpha^* + \mu_i, \beta_i = \beta_j^* + \varepsilon_i\} . \tag{4.1}$$

Appendix B shows how to compute all the relevant information for the scatterer $\vec{C}_i$, and in particular r_i, ε_i, and μ_i. From Fig. 2.1, the elevation of $\vec{C}_{gi}$ with respect to $\vec{m}$ is

$$\beta_i = \cos^{-1}(h/r_{gi}) . \tag{4.2}$$

It follows that the absolute coordinates are computed by combining (2.9b) and (4.1).

Table 4.1 gives the rectangular coordinates of the vectors $\vec{a}(t_a) = \vec{a}(t_c)$, $\vec{a}(t_a)$, $\vec{C}_i$, $\vec{C}_i - \vec{a}(t_a)$. One can now compute r_i, the slant range between the seeker at time $t_a = t_c + t$ and the scatterer $\vec{C}_i$,

$$r_i = \sqrt{|\vec{C}_i - \vec{a}(t_a)|^2} . \tag{4.3}$$

After few manipulations one obtains

$$r_i = |\vec{C}_i - \vec{a}(t_a)| = [h^2\tan^2\beta_i + v^2t^2 + H^2$$

$$-2\, h\, \tan\beta_i\, t\, \{v_x \sin\alpha_i + v_y \cos\alpha_i\} + 2H\, v_z\, t]^{1/2}, \tag{4.4}$$

where $H = h - \hbar_i$, $\alpha_i = \alpha^* + \mu_i$, $\beta_i = \beta_j^* + \varepsilon_i$.

During RRDP of the jth range cell, the reference state is defined by

$$\{\vec{m} = \vec{a}(t_c), \vec{q}^*, \vec{C}_j^*\} \quad , \tag{4.5}$$

and the reference slant range is

$$r_j^* = |\vec{C}_j^* - \vec{m}| \quad . \tag{4.6}$$

The slant range r_i can be expressed in terms of incremental variations about the reference range

$$r_i = r_i(\hbar_i, t, \mu_i, \varepsilon_i \quad ; \quad \vec{m}, \vec{q}^*, \vec{C}_j^*) \tag{4.7a}$$

$$r_j^* = r_i(0,0,0,0 \; ; \; \vec{m}, \vec{q}^*, \vec{C}_j^*) \quad . \tag{4.7b}$$

Table 4.1. Rectangular coordinates for relevant vectors

Vector	x–coordinate	y–coordinate	z–coordinate
$\dot{\vec{a}}(t_a)$	$v_x = v\sin\beta_v \sin\alpha_v$	$v_y = v\sin\beta_v \cos\alpha_v$	$v_z = -v\cos\beta_v$
$\vec{a}(t_a)$	$d_x + v_x t$	$d_y + v_y t$	$h + v_z t$
$\vec{C}_i$	$h \tan\beta_i \sin\alpha_i + d_x$	$h \tan\beta_i \cos\alpha_i + d_y$	$\hbar_i$
$\vec{C}_i - \vec{a}(t_a)$	$h \tan\beta_i \sin\alpha_i - v_x t$	$h \tan\beta_i \cos\alpha_i - v_y t$	$-(h - \hbar_i + v_z t)$

The slant range r_i can be expressed as a Taylor expansion in $\{\hbar_i, t, \mu_i, \varepsilon_i\}$ about the reference slant range r_j^*. Equivalently one can expand the round trip delay $\tau_i = \frac{2r_i}{c}$, where c is the speed of light.

Since the formula for r_i is complicated, we need to define a simple general notation. Assume that $\tau = \tau_i$ is a function of four variables (v_1, v_2, v_3, v_4). The partial derivative of τ with respect to v_k evaluated at the reference state is denoted as $\bar{\tau}_k$. This

notation is extended to multiple partial derivatives. For example

$$\bar{\tau}_{123} = \frac{\partial^3}{\partial v_1 \partial v_2 \partial v_3} \tau \Big|_{v_1 = v_2 = v_3 = v_4 = 0} \quad (4.8)$$

Using this notation, the round trip delay $\tau_i = 2\,\frac{|\vec{C}_i - \vec{m}|}{c}$ becomes

$$\tau(1,2,3,4) = \bar{\tau} + (\bar{\tau}_1 \hbar + \bar{\tau}_2 t + \bar{\tau}_4 \varepsilon) + (\bar{\tau}_{11} \frac{\hbar^2}{2} + \bar{\tau}_{12} \hbar t + \bar{\tau}_{14} \hbar\varepsilon + \bar{\tau}_{22} \frac{t^2}{2} + \bar{\tau}_{23} t\mu$$

$$+ \bar{\tau}_{24} t\varepsilon + \bar{\tau}_{44} \frac{\varepsilon^2}{2}) + (\bar{\tau}_{111} \frac{\hbar^3}{6} + \bar{\tau}_{112} \frac{\hbar^2 t}{2} + \bar{\tau}_{114} \frac{\hbar^2 \varepsilon}{2} + \bar{\tau}_{122} \frac{\hbar t^2}{2} + \bar{\tau}_{123} \hbar t\mu + \bar{\tau}_{124} \hbar t\varepsilon$$

$$+ \bar{\tau}_{144} \frac{\hbar\varepsilon^2}{2} + \bar{\tau}_{222} \frac{t^3}{6} + \bar{\tau}_{223} \frac{t^2\mu}{2} + \bar{\tau}_{224} \frac{t^2\varepsilon}{2} + \bar{\tau}_{233} \frac{t\mu^2}{2} + \bar{\tau}_{234} t\mu\varepsilon + \bar{\tau}_{244} \frac{t\varepsilon^2}{2}$$

$$+ \bar{\tau}_{444} \frac{\varepsilon^3}{6}) \quad (4.9)$$

where the subscript i has been dropped to further simplify the notation. The coefficients of the Taylor series are computed in Appendix D. The Fortran implementation, Subroutine TAYLOR, is in Appendix E.

Numerical evaluation of the coefficients of (4.9) shows that many of them are completely negligible. Therefore, the final practical formula is

$$\tau(\hbar,t,\mu,\varepsilon) = (\bar{\tau} + \bar{\tau}_1 \hbar + \bar{\tau}_4 \varepsilon) + t(\bar{\tau}_2 + \bar{\tau}_{23} \mu + \bar{\tau}_{24} \varepsilon$$

$$+ \bar{\tau}_{12} \hbar + \bar{\tau}_{233} \frac{\mu^2}{2}) + \frac{t^2}{2} (\bar{\tau}_{22} + \bar{\tau}_{223} \mu + \bar{\tau}_{224} \varepsilon). \quad (4.10)$$

The accuracy of (4.10) was validated by comparing the simulated round trip delay and doppler to the exact values. For example, the simulated doppler is (–6994.76 rad/sec) while the actual value is (–6992.87 rad/sec). It follows that (4.10) can be used for an accurate signal synthesis.

5. Range Relative Doppler Processing

The concept of Range Relative Doppler Processing (RRDP) is presented in Part I of [2] and the implementation for linear frequency modulation is part II of [2]. For completeness a concise derivation of RRDP follows for the RF case.

The return corresponding to the scatterer $\vec{C}_i$ located within the jth range cell is

$$s_i(t) = B_i(t) \cos(\omega_o(t - \tau_i) + \xi) \ , \tag{5.1}$$

where ξ is the transmitter phase, τ_i the round trip delay is given in (4.10), and $B_i(t)$ incorporates antenna gain, range attenuation and scatterer strength. Using the Taylor expansion (4.10) one can write

$$\omega(t - \tau_i) + \xi = \omega_o(1 - \bar{\tau}_2)t + \psi_i^* \tag{5.2a}$$

$$\psi_i^* = -\omega_o[t(\bar{\tau}_{23}\mu + \bar{\tau}_{24}\varepsilon + \bar{\tau}_{12}\hbar + \bar{\tau}_{233}\frac{\mu^2}{2}) + \frac{t^2}{2}(\bar{\tau}_{22} + \bar{\tau}_{223}\mu + \bar{\tau}_{224}\varepsilon) + (\bar{\tau} + \bar{\tau}_1\hbar + \bar{\tau}_4\varepsilon)] + \xi. \tag{5.2b}$$

As illustrated in Fig. 5.1 the first step of the processing is mixing at the frequency of the fictitious scatterer,

$$\omega_{oj} = \omega_o(1 - \bar{\tau}_2) \ , \tag{5.3}$$

where ω_{oj} is a function (nearly linear) of j through $\bar{\tau}_2$. The synthetic signal $\exp(j\omega_{oj}t)$ is the estimated return from $\vec{C}_j^*$, a fictitious scatterer at the center of the jth range cell. This means that the analog quadrature components are given by

$$x_{di}^* = B_i(t) \cos(\psi_i^*) \tag{5.4a}$$

$$x_{qi}^* = B_i(t) \sin(\psi_i^*). \tag{5.4b}$$

After filtering and A/D conversion, deramping is performed on the quadrature components as a matrix rotation to cancel the quadratic phase $\omega_o t^2/2$. The deramped quadrature components are

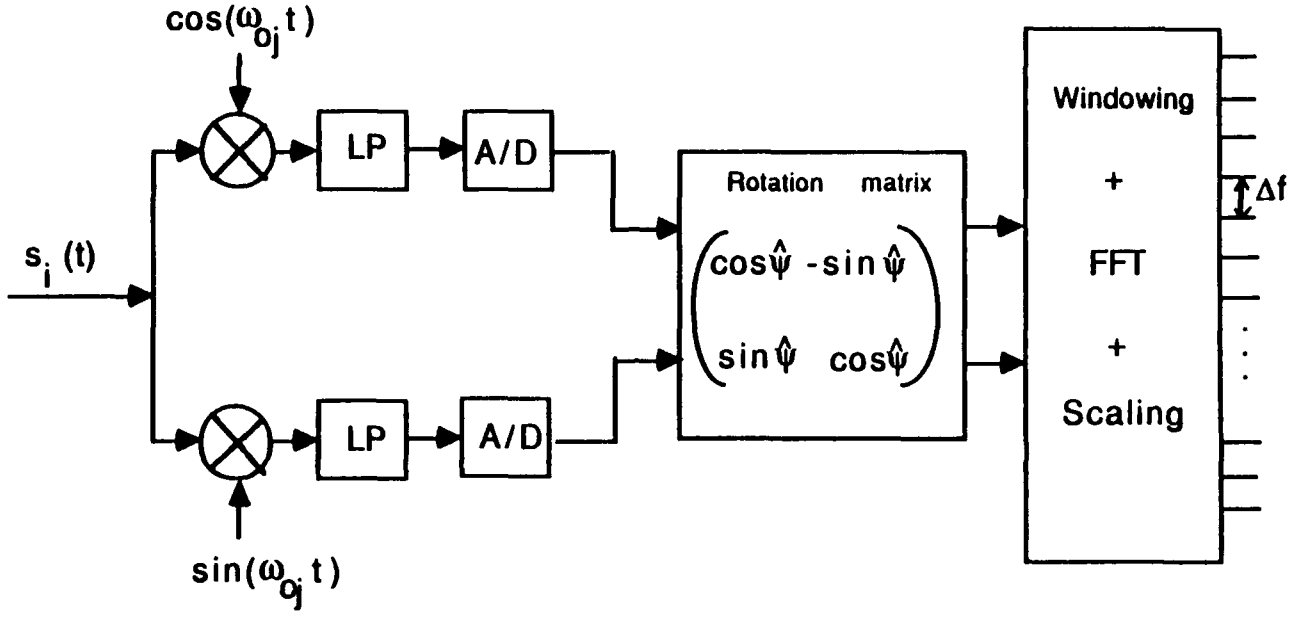

Figure 5.1 Block diagram for Range Relative Doppler Processing for jth range cell.

$$\hat{x}_{di} = B_i(t) \cos(\hat{\psi}(t)) \tag{5.5a}$$

$$\hat{x}_{qi} = B_i(t) \sin(\hat{\psi}(t)) \tag{5.5b}$$

where

$$\begin{aligned}\hat{\psi}(t) &= \psi_i^* + \omega_o \bar{\tau}_{22} \frac{t^2}{2} \\ &= -\omega_o[t(\bar{\tau}_{23}\mu + \bar{\tau}_{24}\varepsilon + \bar{\tau}_{12}\hbar + \bar{\tau}_{233}\frac{\mu^2}{2}) \\ &+ \frac{t^2}{2}(\bar{\tau}_{223}\mu + \bar{\tau}_{224}\varepsilon) + (\bar{\tau} + \bar{\tau}_1\hbar + \bar{\tau}_4\varepsilon)] + \xi\end{aligned} \tag{5.5c}$$

and the residual quadratic phase term is quite small.

It follows from (5.5c) that the relative doppler frequency corresponding to the $\vec{C}_i$ scatterer is (neglecting a small linear frequency variation)

$$\omega_{di} = \frac{d\hat{\psi}(t)}{dt} = -\omega_o[\bar{\tau}_{23}\mu + \bar{\tau}_{24}\varepsilon + \bar{\tau}_{12}\hbar + \bar{\tau}_{233}\frac{\mu^2}{2}]. \tag{5.6}$$

When ($\hbar = 0, \varepsilon = 0$), i.e., on the median of the range cell, (5.6) becomes

$$\omega_{di}(\hbar = 0, \varepsilon = 0) = -\omega_o[\bar{\tau}_{23}\mu + \bar{\tau}_{233}\frac{\mu^2}{2}], \tag{5.7}$$

where the μ^2 term is relatively small.

Figure 5.1 shows that the processing of the array of deramped quadrature components consists of three steps: windowing, FFT, and scaling. The Kaiser window is recommended [5,6] because it allows a continuous tradeoff between resolution and leakage. The output of the FFT should be in the modulus/phase format. The modulus must be scaled to compensate for antenna pattern and range attenuation. The FFT cells are numbered (k=0, N–1) and the kth cell corresponds to a relative doppler interval:

$$(k-0.5)\Delta\omega < \omega_{rd} < (k+0.5)\Delta\omega \tag{5.8a}$$

$$\Delta\omega = 2\pi\,\Delta f = 2\pi/D_{ob} \tag{5.8b}$$

where D_{ob} is the duration of the observation interval in seconds.

A point scatterer will affect three adjacent FFT cells as illustrated in Fig. 5.2 for a rectangular window and for a Kaiser window with parameter equal to π. Assume for example that the relative doppler for a point scatterer is $k\Delta\omega$, this means that the position of the scatterer in the jth range cell is defined by the angular increments (ε,μ) such that $\omega_{di} = k\Delta\omega$ in (5.6). As a first approximation one could use the scaled modulus A_k of the kth cell (on–target) as an estimate of the strength of the scatterer. However, A_k varies with the relative location of the scatterer within the range cell, being maximum when the scatterer is located at the center. The normalized values of the (early, on–target, late) components are shown in Fig. 5.3 for two cases: (a) scatterer at the center of the cell, and (b) scatterer on the border between on–target and late. Theoretically, one could use an interpolation algorithm to define the center of the scatterer more precisely from the (early, on–target, late) components; very much as for range interpolation. However, this is not practical in the case of multiple scatterers. Instead we will detect a single point scatterer in a range cell as three scatterers located at the centers of adjacent cells. Continuing our example the three scatterers will be detected at

$$\{(\varepsilon=0, \mu_{k-1}) \ , \ (\varepsilon=0, \mu_k) \ , \ (\varepsilon=0, \mu_{k+1})\} \tag{5.9}$$

where μ_ℓ is the solution of (5.7) for $\{\omega_{di} = \ell\ \Delta\omega, \ \ell = k-1, k+1\}$. The value of μ which defines the border line between the kth and the (k+1)th FFT cells is obtained by solving (5.7) for $\omega_{di} = (k+0.5)\Delta\omega$.

The maximum values of (ε,μ) are denoted as εmx and μmx, respectively. Appendix D explains the computation of $(\varepsilon mx, \mu mx)$.

Since a point scatterer will also affect the adjacent cells it will be detected as nine samples altogether. In a typical application εmx is much smaller than μmx and formula (5.7) can be used. It indicates a nearly linear correspondence between doppler

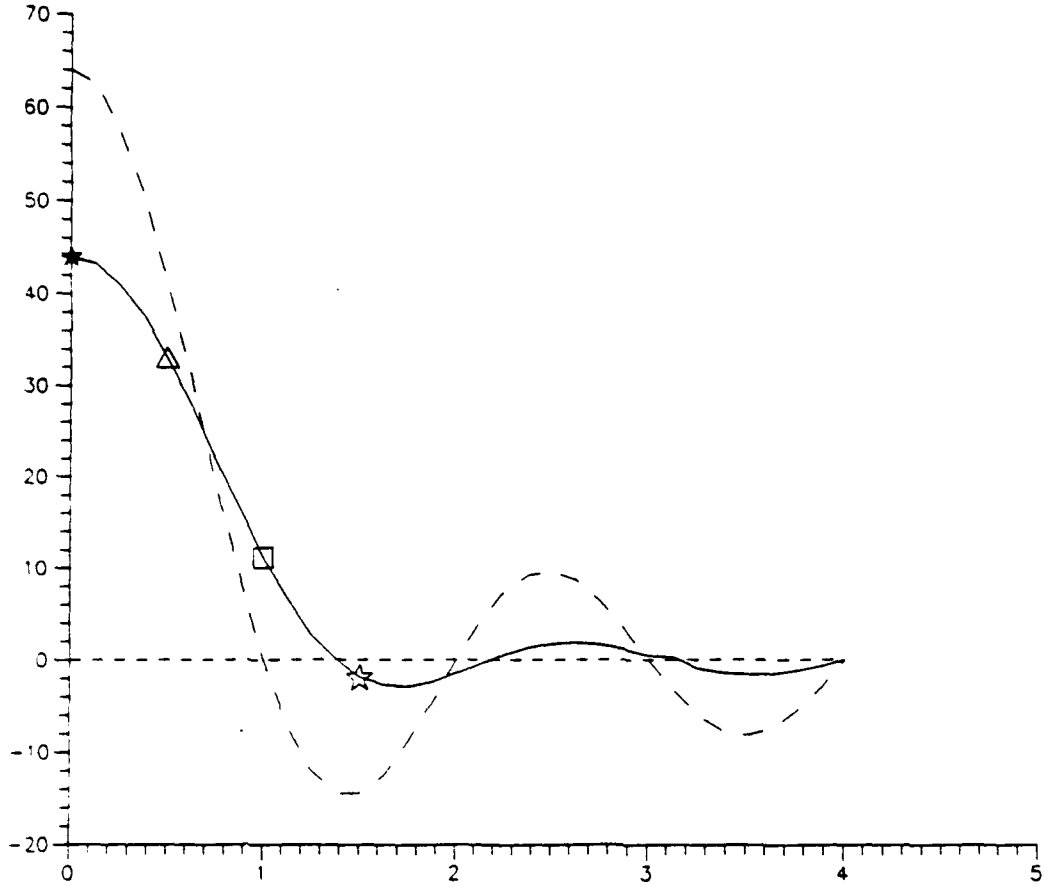

Figure 5.2. Normalized response for both rectangular and Kaiser windows, N=64. Kaiser (early,on,late) samples are shown. Central target(☆ □ □). Border-line target (☆ ☆ ☆).

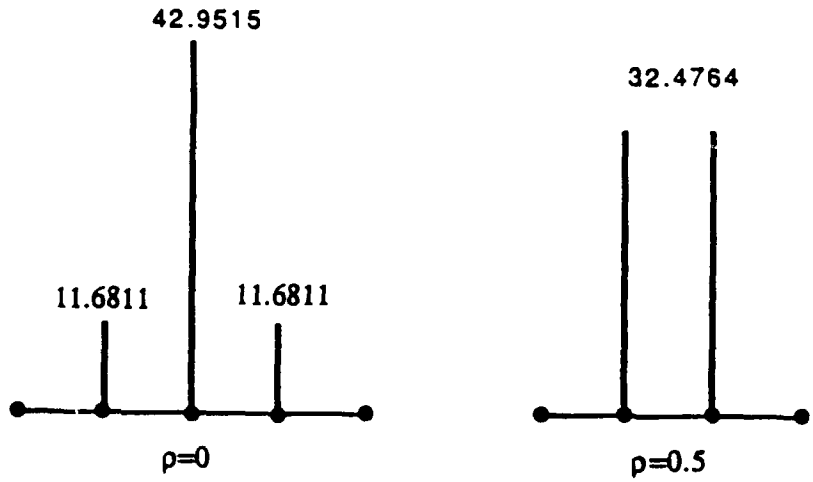

Figure 5.3 FFT line spectrum for a unit sinewave and Kaiser window($\beta=\pi$). $N=64, f_o=(m+\rho)\Delta f$ ($\rho=0$:central target,$\rho=0.5$:borderline target).

boundaries and angular boundaries because $(\bar{\tau}_{\mu\mu}\mu^2/2)$ is much smaller than $(\bar{\tau}_{t\mu}\mu)$ for the useful range of μ. Yet including the quadratic term increases the accuracy by about five percent.

6. Reduction of Impulse Invariance Over a Footprint

A unit scatterer is detected as a narrow three–dimensional Gaussian shaped pulse [1,2]. The impulse corresponding to a unit scatterer located at $\vec{q}^*$ (the center of the footprint) will be defined as the impulse response of the system. Impulse invariance means that a unit scatterer anywhere within the footprint will produce the same impulse response. It is difficult to achieve impulse invariance over the entire footprint because of range attenuation, antenna gain variation, and FFT sampling effects. The results in [2] show the impulse response varying by as much as 31%, and it was recommended to correct this problem. One of the major accomplishments of the present effort is to reduce impulse invariance over an entire footprint to within ±4%. The causes of impulse variation and the proposed solutions are discussed below.

The range to a scatterer varies very little during the observation interval, but it depends significantly on the location of the scatterer within the footprint. The range variation is even greater from one footprint to the other. Since the radar equation involves the fourth power of range, the range attenuation must be compensated with respect to a reference range, r_{ref}, on the ground map. More precisely, the range compensation of the jth cell is performed by multiplying the amplitude of the FFT cells by $(r_j^*/r_{ref})^4$, where r_j^* is the median range. This compensation works very well because range is precisely known at central time and changes very little during the observation interval.

The antenna gain varies by as much as 3dB within the footprint. It follows that a scatterer on the edge of the footprint will produce an impulse response half as small as a scatterer at the center of the footprint. For each FFT cell the antenna gain can be computed at central time. Therefore, most of the variation due to antenna gain can be

removed by dividing each cell return by the estimated antenna gain for that FFT cell.

We have explained that the detection of scatterers was transformed into a problem of power spectrum analysis [2]. It is well known that for sinewaves of the same strength the peak depends on the frequency f: highest peak for $f = m\Delta f$, lowest for $f = (m+0.5)\Delta f$, where m is an integer. This means that a scatterer will appear larger if it is in the center of the FFT cell than if it is on the border between two FFT cells. Figure 5.3 shows that in the case of a Kaiser window, the peak variation is about 25% which is an unacceptable ratio. The variation in peak amplitude due to the FFT sampling error is reduced from 25% to 5% through frequency interpolation. This is accomplished by adding N zeros to the N return samples prior to FFT transformation. The effect is to halve Δf. The number of significant samples without interpolation is 3 and with interpolation it becomes 6. We have shown that the sum of the significant samples remains practically invariant, with or without interpolation.

As a result of the combined proposed corrections, the impulse response within one footprint becomes invariant within ±4%. This is observed in Fig. 12.1 which displays the reflectivity of a footprint versus range and azimuth for eleven scatterers distributed over the footprint. Therefore, the detection of scatterers within one single footprint can now be considered as impulse invariant.

Invariant mapping is used to transfer the footprint information onto an absolute x–y ground reflectivity map. In this manner one can accumulate the information for a sequence of footprints which should significantly improve the detection performance. The mapping technique is invariant in terms of volume. In other words, an impulse is mapped onto an impulse of the same volume. However, the impulse obtained after invariant mapping is slightly wider and shorter, which cannot be avoided. The peaks after invariant mapping vary within ±5.5% depending on the relative position of the FFT cell representing the scatterer with respect to the rectangular grid. Figure 13.1

shows that the x–y reflectivity map obtained using invariant mapping is nearly impulse invariant.

7. Other Useful Changes

Several major improvements to the RRDP simulation were discussed in the previous sections. They include: (1) extension to 3–D trajectories, (2) an accurate circular dish simulation, (3) an expanded Taylor expansion which includes the effect of scatterer height, (4) reduction of impulse invariance through FFT interpolation and a better compensation for range attenuation and antenna gain. This section discusses other useful changes, except for those related to graphics displays and invariant mapping which are documented in separate sections.

Two main programs were developed: (1) FRRDP, where the output is a reflectivity map versus range and azimuth and (2) CRRDP, where the output is a reflectivity map versus absolute x–y coordinates. Both programs have the same input files: CRRDPI.INP which defines the system parameters, and CLUTINFI.INP which defines the scatterers. Also subroutine RECPOL has been deleted. The overviews of FRRDP and CRRDP are in sections 9 and 11, respectively.

The subroutine CLUTINFO which is used to enter the scatterer data was also rewritten. Major changes include: (1) the position of $\vec{C}_i$ can be entered as $\{x_i, y_i, h_i\}$ or $\{\mu_i, r_{gi}, h_i\}$, where h_i is the scatterer height, (2) the output array SMOMAP has been modified to account for h_i, (3) the subroutines XYTOMU and MUTOXY were eliminated. As the ith scatterer is read in the file, the information $\{x_i, y_i, \mu_i, r_{gi}, h_i\}$ is displayed on the screen. CLUTINFO provides information to the main program, on the scatterers that are both within the selected window display and the footprint, through four arrays: SMOMAP, WADRLO, WADRUP, and WADSMO. Range smoothing performed by RANGSMOO, transforms one real scatterer into three smoothed scatterers, and this information is stored in SMOMAP as 15 components:

{amp(1), sfas(1), mu(1), jranon−1, zsca(1), amp(2), sfas(2),

mu(2), jranon, zsca(2), amp(3), sfas(3), mu(3), jranon+1, zsca(3)} (7.1)

where the indexes {1,2,3} refers to {early, on, late}, and jranon is the range index for the on−component. The elements of the arrays WADSMO and WJRAN for the kth smoothed scatterer are: {WADSMO(k) = address of the scatterer range index in SMOMAP} and {WJRAN(k) = SMOMAP(k) = range index for scatterer}. Once completed, the two arrays {WADSMO, WJRAN} are sorted with respect to increasing range index. The arrays WADRLO (JRAN) and WADRUP(JRAN), where JRAN varies over all range cells, contain the lower and upper addresses of the sorted scatterers which are within the JRAN cell. If WADRLO (JRAN) equals zero this means that no useful scatterer falls within this cell. On the other hand {WADRLO (JRAN) = k1, WADRUP (JRAN) = k2} means that the sorted scatterers whose index satisfies $k_1 \le k \le k_2$ fall within the JRAN cell. The information for the kth sorted scatterer can be found in SMOMAP which is addressed using WADSMO(k).

In both main programs (FRRDP and CRRDP) and in CLUTINFO the inputs are entered interactively. More precisely, the steps are as follows: (1) the current input data file is read, (2) current values are printed on the screen, (3) a prompt K = ? allows the user to specify the index of the data to be changed, (4) if the 3rd data input need to be changed to 200.0 we enter K = 3, then 200.0 at the X = ? prompt, and (5) when all the necessary changes have been made, enter K = 0 and the updated data is rewritten on the input file. This approach is very practical because usually no more than two parameters are changed in a new run. Subroutines WINPUT and WCHANGE have been developed for this purpose, they are listed in Appendix E.

Both signal synthesis and deramping involve very large phases. These phases have to be evaluated modulo 2π. We found that it is critical to use double precision for

phases and phase residues. This affects subroutines TSIGNAL and DERAMP which are listed in Appendix E.

8. Display of FRRDPG.GRD and CRRDPG.GRD Using SURFER

The main programs FRRDP and CRRDP generate, respectively, the files FRRDPG.GRD and CRRDPG.GRD. The data in FRRDPG.GRD describes the reflectivity versus azimuth and range. The data in CRRDPG.GRD describes the reflectivity versus absolute x–y coordinates. The purpose of this section is to explain how the format of FRRDPG.GRD and CRRDPG.GRD was selected to allow direct display by SURFER [7].

The information for a 3–D display or a contour map could be written as a three column file, where the number of rows is equal to the number of data points and one row contains the x–y–z coordinates for an output point. The disadvantage of such a format is that SURFER will require gridding prior to display. Therefore, it is much better to write the files FRRDPG.GRD and CRRDPG.GRD in the format specified by SURFER. This output format is as follows: (1) five rows which specify the file as indicated by the graphics manual, and (2) the data in blocks of length N_x, one block for each constant y. There are a total of N_y blocks, and each block is written in rows of 10 real numbers. Note that (N_x,N_y) are the dimension of the x and y coordinates.

SURFER from Golden Software, Inc., [7] is used to transform CRRDPG.GRD into a three dimensional display. In a standard application the first step is to compute a grid starting from the x–y–z data. However, gridding is a very time consuming process and it can cause errors if the smoothing parameters are not selected properly. Fortunately, when the increments for x and y are constant with no data missing, the gridding process can be entirely avoided through proper formatting. This is the case in

our application. It follows that CRRDPG.GRD is written in the block format specified in the previous paragraph.

The file CRRDPG.GRD is in the correct format for a 3–D display using subroutine SURF, or for a contour map using subroutine TOPO [7]. Both SURF and TOPO allow: (1) selection of color, (2) discrete or continuous lines, (3) scaling, (4) rotation, (5) labeling, (6) entering a title, and (7) screen gridding. TOPO also provides control of the minimum and maximum contour levels, and of the contour increments. In both cases we can obtain a hard copy by using one of four devices: (1) an IBM graphics Proprinter, (2) an HP 7475A pen plotter, (3) an HP laserjet series II printer, and (4) an HP paint jet printer which allows the use of colors. In this report all the hard copy displays were obtained using the HP laserjet printer. Examples of footprint reflectivity maps are in section 13.

9. Overview of Computer Simulation for Detection Within One Footprint: FRRDP

The overview of FRRDP is best explained using the functional block diagram of Fig. 9.1. The goal of this computer simulation is to write the file FRRDPG.GRD which SURFER uses to display the reflectivity map versus range and azimuth. A second output file, FRRDPP.PRT, is only used for debugging. Note that SURFER includes both SURF and TOPO. The block diagram is an implementation of the techniques and formulas defined in the previous sections. The programs are written in Fortran 77, in a modular structure that allows easy modifications. They are listed in Appendix E. A functional description follows.

The initialization block includes two steps. In the first step the input parameters are entered interactively by reading, displaying on the screen, and updating the input parameter file CRRDPI.INP. A printout of CRRDPI.INP is in section 13. Note that the input data includes the definition of the display window through the array WPAR. Only the scatterers within the specified window are involved in the simulation and mapping. The subroutine WINPUT is used to enter the parameters interactively as

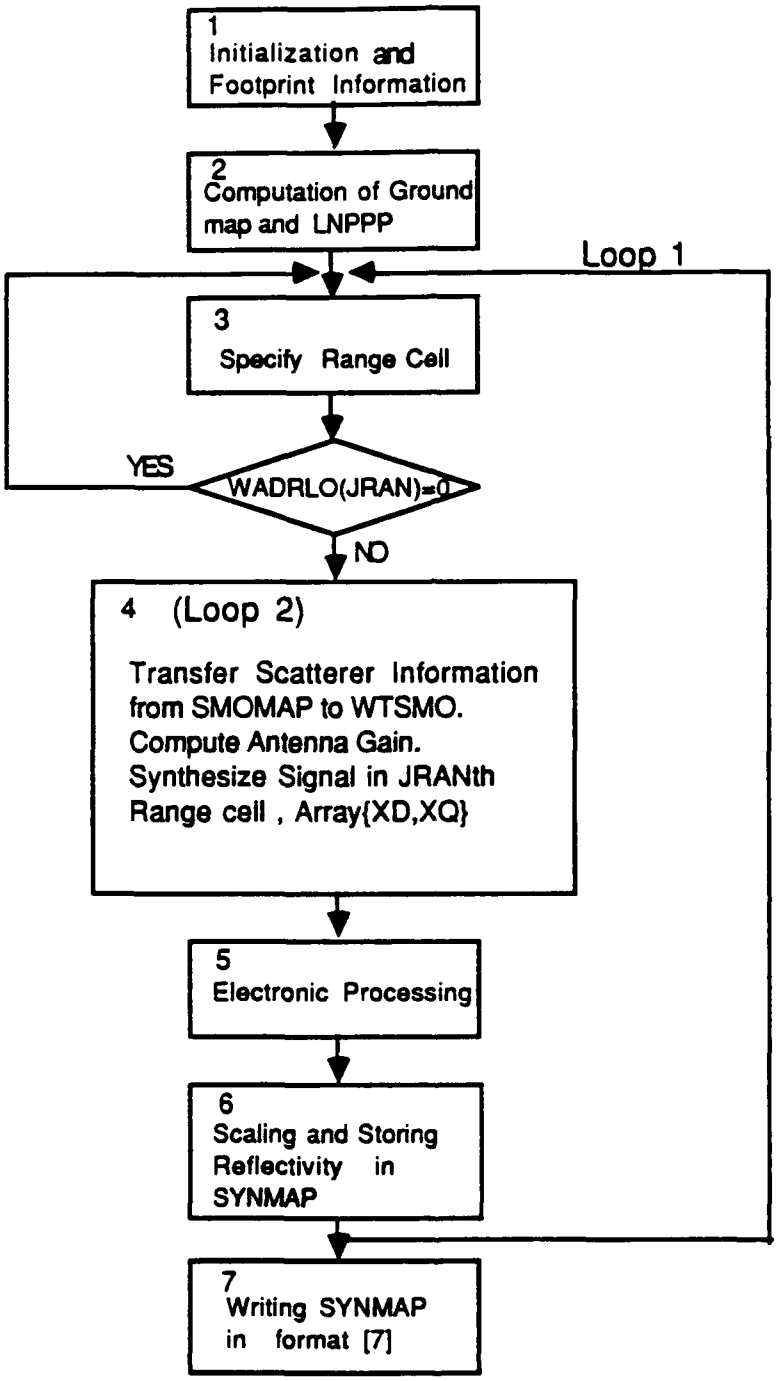

Figure 9.1 Block diagram of the FRRDP simulation:
SYNMAP contains reflectivity versus azimuth and range.

explained in section 7, and PRTXY displays the current parameters on the screen. Subroutine WINPUT calls WCHANGE which uses KINPUT and XINPUT to enter integer or real numbers on line. The second step is the precomputation of the trajectory and footprint information at central dwell time. Also the output reflectivity array SYNMAP is initialized. The array WMU3DB stores the maximum azimuth increments μmx, one for each range cell.

The ground map is computed using subroutine CLUTINFO. The scatterer data is in the file CLUTINFI.INP. The format is as follows: (1) the first line is the signed number of scatterers, where a positive sign indicates x–y coordinates for the scatterer, while a negative sign means relative spherical coordinates, and (2) each scatterer is entered as three lines of data, either (x_i, y_i, h_i) or (μ_i, r_{gi}, h_i). The scatterer data is read, displayed on the screen, modified on line, and rewritten on the file CLUTINFI.INP, using subroutine WINPUT which was discussed earlier. Only the scatterers that fall within the selected footprint and display map (including range smoothing effect) are processed further. As well known, a point scatterer affects three consecutive range cells because of the finite bandwidth of the system. This effect which is denoted as range smoothing is simulated using subroutine RANGSMOO, which splits each scatterer into three equivalent scatterers. All the necessary scatterer information is passed to the main program through the arrays SMOMAP, WADRLO, WADRUP, and WADSMO. More details on CLUTINFO are in section 7. It is convenient to use the index KNPPP which varies from 1 to NFFT from minimum to maximum doppler, rather than the FFT index KFF which does not correspond to a continuous doppler variation. The correspondence is: {KFF = $<$KNPPP + NFFT/2 – 1 $>_{NFFT}$ + 1, KNPPP = 1, NFFT}, where $<\cdot>_{NFFT}$ means modulo NFFT. Furthermore, the useful range of KNPPP is limited to within the 3dB footprint, that is {KPPPMN < KNPPP < NPPPMX}. The change from KNPPP to KFF is effected by the array LNPPP.

The heart of FRRDP is loop 1 which corresponds to "DO 35" in the program. In this loop the range cells are processed sequentially: JRAN = 1, JRANMX.

In Block 3 a check for the presence of smoothed scatterers within the current JRANth cell is performed. If WADRLO(JRAN) = 0 there are no scatterers and loop1 has been completed.

Block 4 represents loop 2 which corresponds to "DO 30" in the program. Its purpose is to synthesize the signal {XD(KDQ), XQ(KDQ), KDQ = 1, NFFT} in the JRANth range cell. The contributing scatterers are defined by {WADRLO(JRAN) < KTHSCA < WADRUP(JRAN)}. The information for the KTHSCA scatterer is transferred from SMOMAP into the array WTSMO. The antenna gain and the contribution to the signal due to the KTHSCA scatterer are computed using subroutines SCAGAN and TSIGNAL which are listed in Appendix E.

The electronic processing of the JRANth cell is in block 5. It consists of the following steps: (1) deramping, (2) windowing, and (3) fast Fourier transformation (FFT). The subroutines DERAMP, WINDOWS and FFT which are listed in Appendix E are used for this purpose. Note that the FFT is performed on the extended arrays {XD(k), XQ(k), k = 1, 2*NFFT} which are padded with NFFT zeros. This corresponds to the implementation of frequency interpolation which was discussed in section 6.

Block 6 is concerned with scaling of the FFT cells and the synthesis of the reflectivity array SYNMAP. First, the array WAGNCL which contains the antenna gains at the center of the FFT cells is computed, ("DO 39" in the program). Then, both the scaling and the computation of SYNMAP are performed in "DO 41". The scaling which compensates for antenna gain and range variation is the implementation of section 6.

Block 7 corresponds to the two "DO 60" in the program. The purpose of this block is to write the file FRRDPG.GRD which contains the reflectivity map in the format suggested by [7] (see section 8).

10. New Invariant Mapping Technique

Application of SURF on the file FRRDPG.GRD written by FRRDP produces a 3–D reflectivity map, where {x = azimuth, y = range, z = reflectivity}. In many applications it is desirable to represent the information as an absolute reflectivity map {x= absolute–x, y = absolute–y, z = reflectivity}. This approach allows combining the information of a sequence of footprints into an overall absolute map. An invariant mapping technique to transfer the information from the footprint to an absolute map was developed in [1,2]. The reflectivity of one trapezoidal FFT cell is shared between the overlapping rectangular grid cells on the basis of overlap area. It follows that the efficiency of the invariant mapping algorithm depends mostly on the computation of the area of overlap. In [1,2] the trapezoid was split into two triangles and various overlap configurations were considered. In this report we are presenting a new algorithm which computes directly the overlap of the trapezoid cells with the absolute grid rectangles. The new algorithm is much more general, compact, and efficient than that of [1,2]. It is discussed below.

Prior to writing the algorithm we need to define: (1) the trapezoidal FFT cells being mapped and their reflectivity, (2) the absolute rectangular cells and their estimated reflectivity, and (3) the area of overlap between the mth trapezoidal cell and the kth rectangle;

TRP(m) $\Leftrightarrow$ mth trapezoidal cell

RFT(m) $\Leftrightarrow$ reflectivity of the mth trapezoidal cell

REC(k) $\Leftrightarrow$ kth absolute rectangular cell

RFR(k) $\Leftrightarrow$ estimated reflectivity of the kth rectangular cell

OVL(m,k) $\Leftrightarrow$ overlap area between mth trapezoid and kth rectangle.

Using the above definitions, the invariant mapping algorithm implements the following formula

$$RFR(k) = \frac{\sum_{m} OVL(m,k)\ RFT(m)}{\sum_{m} OVL(m,k)} \qquad (10.1)$$

where m defines all the trapezoids which overlap the kth rectangle.

The invariant mapping algorithm is implemented in CRRDP which is discussed in section 11. It involves the following subroutines: APEXTRAP and MAPRGM. The subroutine MAPRGM calls MAPAZM which calls POLONFRA. The only complicated subroutine is POLONFRA which is discussed next.

Figure 10.1 shows the mth trapezoid being mapped on its frame prior to transfer to the absolute window defined by WPAR = {DELMAP,THRESH,XMN,YMN,XMX, YMX}. Only the frame that contains the trapezoid is involved in the overlap. The purpose of POLONFRA is to compute all the nonzero OVL(m,k). The inputs of POLONFRA are as follows:

NBA ⇔ # trapezoid apexes = 4

WAX ⇔ x–coordinates of trapezoid apexes

WAY ⇔ y–coordinates of trapezoid apexes

WPAR ⇔ array defining the absolute window being mapped.

The outputs of POLONFRA are:

(IIO,JJO) ⇔ absolute coordinates of frame upper left corner

(IMX,JMX) ⇔ frame dimensions

TAREA ⇔ area of trapezoid

WWAREA(I,J) ⇔ overlapping area for the (I,J)th rectangle within the frame.

IOKRO = 0 if apexes are entered in counterclockwise order, and 1 otherwise.

IFAIL ⇔ test for failure, it should be equal to zero.

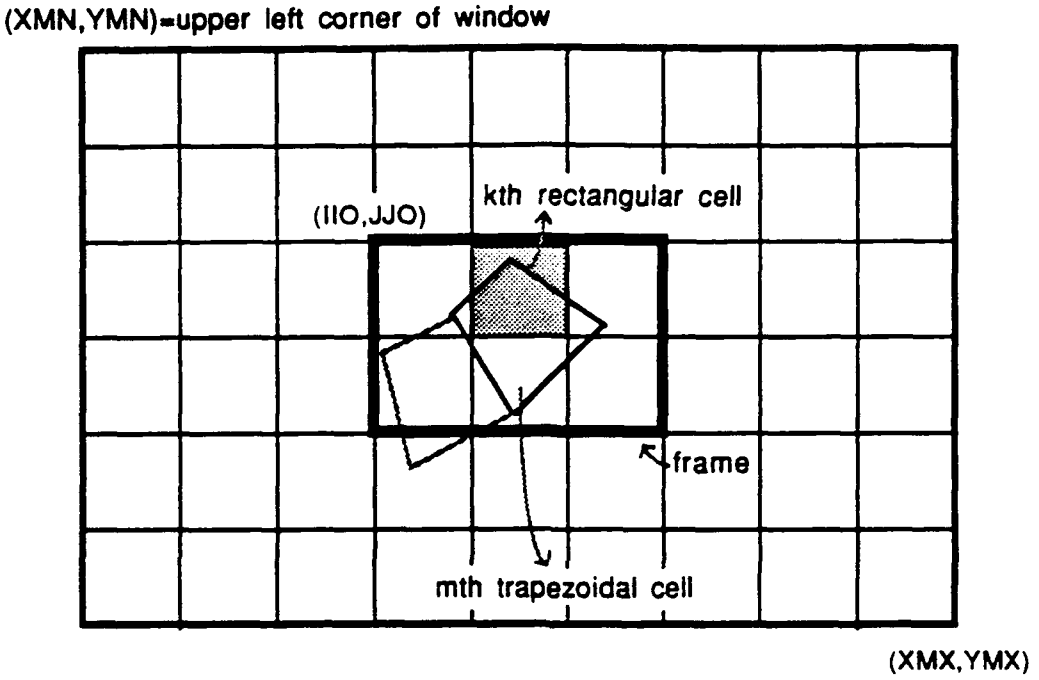

Figure 10.1 Mapping the mth trapezoidal cell on its frame prior to transfer to absolute grid map.

Conceptually POLONFRA operates as follows: (1) following the trapezoid counterclockwise contour (CCWC), one computes all the intersects with the rectangular grid, (2) the intersects are ordered in order of CCWC, (3) duplicate intersects are eliminated, (4) the corners of the rectangles within the frame are defined as exterior or interior to the trapezoid, and (5) the overlap areas of the rectangles which have interior and/or intersects with the trapezoid are computed. The overlap contour for the rectangle is defined by ordering the interior corners and/or intersect points in CCWC. In this manner we define a convex contour whose area is easy to compute using the function AREPOL. Listings for POLONFRA and associated subprograms are in Appendix F.

11. Overview of Computer Simulation with Invariant Mapping: CRRDP

The overview of CRRDP is best explained using the functional block diagram of Fig. 11.1. The goal of this computer simulation is to write the file CRRDPG.GRD which SURFER uses to display the reflectivity map versus the absolute x–y coordinates. A second output file, CRRDPP.PRT, is only used for debugging. The main difference between FRRDP and CRRDP is that FRRDP generates the footprint information directly, while CRRDP transfer the information onto a absolute x–y grid using the invariant mapping technique discussed in section 10.

It follows that the first block of CRRDP is the same as the first block of FRRDP except for the initialization of SYNMAP and SUMARA ("DO 13" is the program). These two arrays are used in subroutine MAPRGM to accumulate both the numerator and denominator of (10.1). Blocks {2,3,4,5} in CRRDP are the same as in FRRDP and were discussed in section 9.

Block 6 performs antenna and range attenuation compensation as explained in section 9. Block 7 implements Invariant Mapping. It performs two tasks: (1) computation of the trapezoid FFT cells apexes coordinates in the current range cell

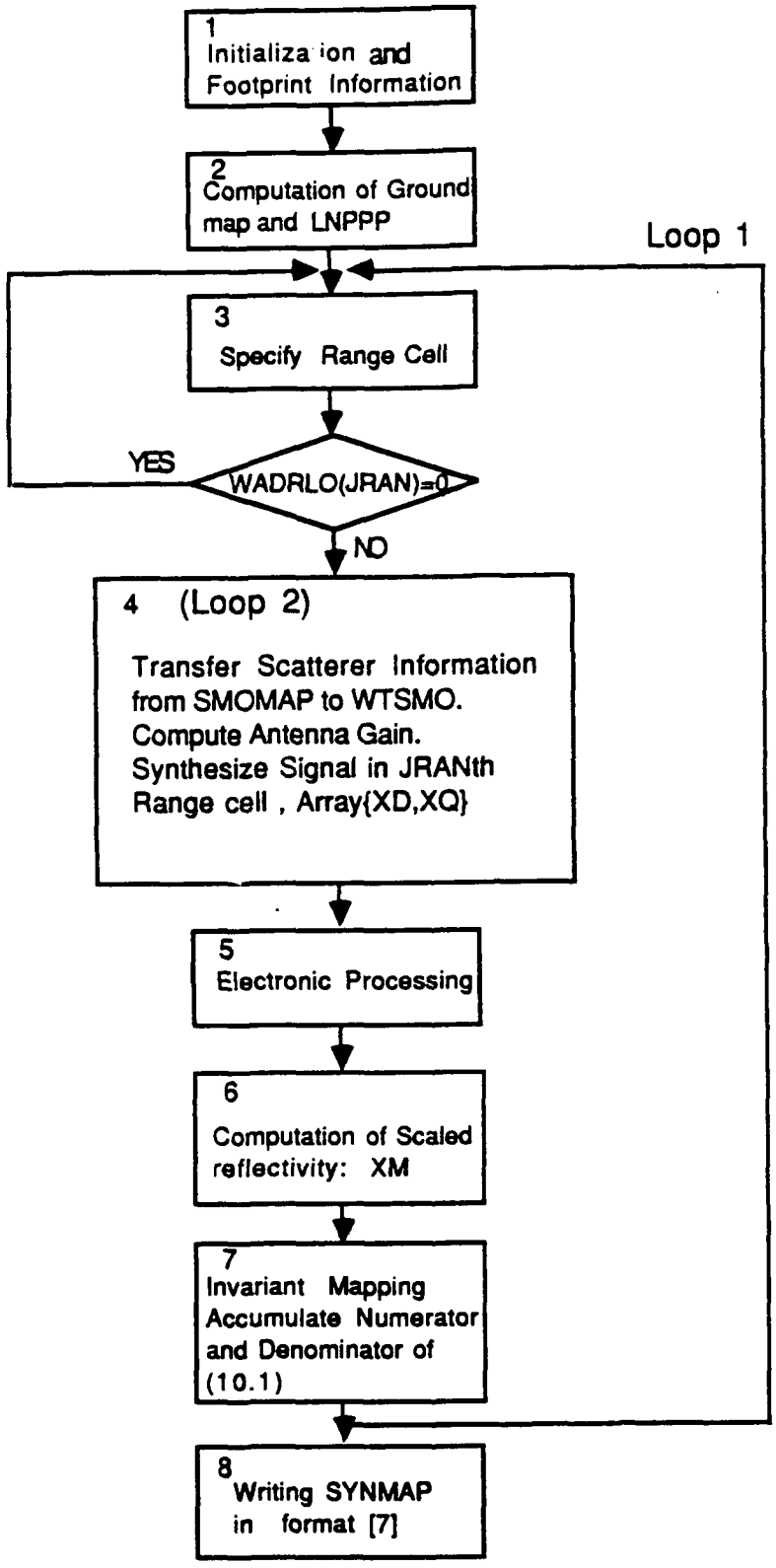

Figure 11.1 Block Diagram for the CRRDP Simulation:
SYNMAP contains the absolute x-y reflectivity map.

(subroutine APEXTRAP), and (2) computation of the arrays SYNMAP and SUMARA which represent respectively, the numerator and denominator of (10.1). The accumulation in SYNMAP and SUMARA is performed by subroutine MAPAZM which is called by MAPRGM.

Block 8 corresponds to the two "DO 60" in the program. First, the reflectivity map SYNMAP is computed according to (10.1). Then, the reflectivity map versus absolute x–y coordinates is written on the file CRRDPG.GRD using the format recommended by [7]. Appendix G contains the listings for CRRDP and its subroutines, except for those listed earlier in Appendixes C, D, E, and F.

12. Demonstration of the FRRDP Simulation

In order to run FRRDP or CRRDP one must first define the following: (1) the seeker antenna information, (2) the footprint center, (3) the selected frame window, and (4) the electronic processing parameters. This is accomplished interactively using the subroutine WINPUT on the data file CRRDPI.INP. Table 12.1 shows the file CRRDPI.INP with data set #1, which is used for most of our examples. This file is self explanatory since each parameter is followed by a comment. A demonstration diskette (DEMO #1) is included in [8].

The clutter configuration is also entered interactively through subroutine CLUTINFO which uses WINPUT to read, and update as needed, the clutter file CLUTINFI.INP. This file contains the signed number of scatterers followed by $\{\mu_i, r_{gi}, \hbar_i\}$ for negative or $\{x_i, y_i, \hbar_i\}$ for positive sign (see section 7). Table 12.2a contains the data for eleven scatterers with zero height which are distributed over the footprint specified by Table 12.1. When FRRDP is executed using Table 12.2a the scatterer information appears on the screen as shown in Table 12.2b.

All the 3–D displays in this section are for a 312° rotation about the z–axis (entered in SURF). Figure 12.1 shows the 3–D reflectivity map versus azimuth and range, obtained by using SURF [7] on the file FRRDPG.GRD created by FRRDP for

Table 12.1. Data set #1 for file CRRDPI.INP

.0100000	TIMCNT=central time
.0200000	TIMDWL=observation interval
150.0000000	VELOM=seeker velocity
.0000000	ALFM=seeker azimuth angle
1.2217300	BETM=seeker elevation angle
.0000000	WMPOSC(1)=seeker x-coordinate
7.0000000	WMPOSC(2)=seeker y-coordinate
900.0000000	WMPOSC(3)=seeker z-coordinate
516.2463000	WQ9(1)=footprint center x-coordinate
744.2216000	WQ9(2)=footprint center y-coordinate
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
1.0000000	DELRAN=range bin
6.0000000	FFNBIT=# FFT bits
.7500000	WPAR(1)=rectangular grid increment
.1000000	WPAR(2)=mapping threshold
480.0000000	WPAR(3)=x-min display window
710.0000000	WPAR(4)=y-min display window
550.0000000	WPAR(5)=x-max display window
780.0000000	WPAR(6)=y-max display window
.0000000	XIFXMT=transmitter phase
4.0000000	SELWIN=define FFT window
3.1415900	BETKAI=Kaiser window parameter
2.0000000	SELDIS=define dish simulation
1.0000000	BITINT=define FFT interpolation

Table 12.2a. Data set #1 for input file CLUTINFI.INP.

```
-34.00000000  W1=3* #scatt.+1,pos. if (xi,yi),neg. if (mui,rgi)
    .0000000  X1=x(1st scat), or mu1
1272.7910000  Y1=y(1st scat), or rg1
   0.0000000  Z1=z(1st scat) = hbar1
   -.0304062  X1=x(2nd scat), or mu2
1272.7910000  Y1=y(2nd scat), or rg2
    .0000000  Z1=z(2nd scat) = hbar2
    .0304062  X1=x(3rd scat), or mu3
1272.7910000  Y1=y(3rd scat), or rg3
    .0000000  Z1=z(3rd scat) = hbar3
    .0000000  X1=x(4th scat), or mu4
1285.7910000  Y1=y(4th scat), or rg4
    .0000000  Z1=z(4th scat) = hbar4
   -.0267718  X1=x(5th scat), or mu5
1285.7910000  Y1=y(5th scat), or rg5
    .0000000  Z1=z(5th scat) =hbar5
    .0267718  X1=x(6th scat), or mu6
1285.7910000  Y1=y(6th scat), or rg6
    .0000000  Z1=z(6th scat) = hbar6
    .0000000  X1=x(7th scat), or mu7
1260.7910000  Y1=y(7th scat), or rg7
    .0000000  Z1=z(7th scat) = hbar7
   -.0273642  X1=x(8th scat), or mu8
1260.7910000  Y1=y(8th scat), or rg8
    .0000000  Z1=z(8th scat) = hbar8
    .0273642  X1=x(9th scat), or mu9
1260.7910000  Y1=y(9th scat), or rg9
    .0000000  Z1=z(9th scat) = hbar9
    .0000000  X1=x(10th scat), or mu10
1295.7910000  Y1=y(10th scat), or rg10
    .0000000  Z1=z(10th scat) = hbar10
    .0000000  X1=x(11th scat), or mu11
1250.7910000  Y1=y(11th scat), or rg11
    .0000000  Z1=z(11th scat) = hbar11
```

Table 12.2b. Screen printout for scatterer input in Table 12.2a

```
XI=.51624E+03 YI=.74422E+03 MUI=  .00000E+00 RGI=.12728E+04 HI=.00E+00
XI=.49359E+03 YI=.75957E+03 MUI= -.30406E-01 RGI=.12728E+04 HI=.00E+00
XI=.53842E+03 YI=.72818E+03 MUI=  .30406E-01 RGI=.12728E+04 HI=.00E+00
XI=.52674E+03 YI=.75920E+03 MUI=  .00000E+00 RGI=.12858E+04 HI=.00E+00
XI=.50641E+03 YI=.77303E+03 MUI= -.26772E-01 RGI=.12858E+04 HI=.00E+00
XI=.54668E+03 YI=.74483E+03 MUI=  .26772E-01 RGI=.12858E+04 HI=.00E+00
XI=.50646E+03 YI=.73025E+03 MUI=  .00000E+00 RGI=.12608E+04 HI=.00E+00
XI=.48648E+03 YI=.74384E+03 MUI= -.27364E-01 RGI=.12608E+04 HI=.00E+00
XI=.52606E+03 YI=.71612E+03 MUI=  .27364E-01 RGI=.12608E+04 HI=.00E+00
XI=.53474E+03 YI=.77063E+03 MUI=  .00000E+00 RGI=.12958E+04 HI=.00E+00
XI=.49824E+03 YI=.71850E+03 MUI=  .00000E+00 RGI=.12508E+04 HI=.00E+00
```

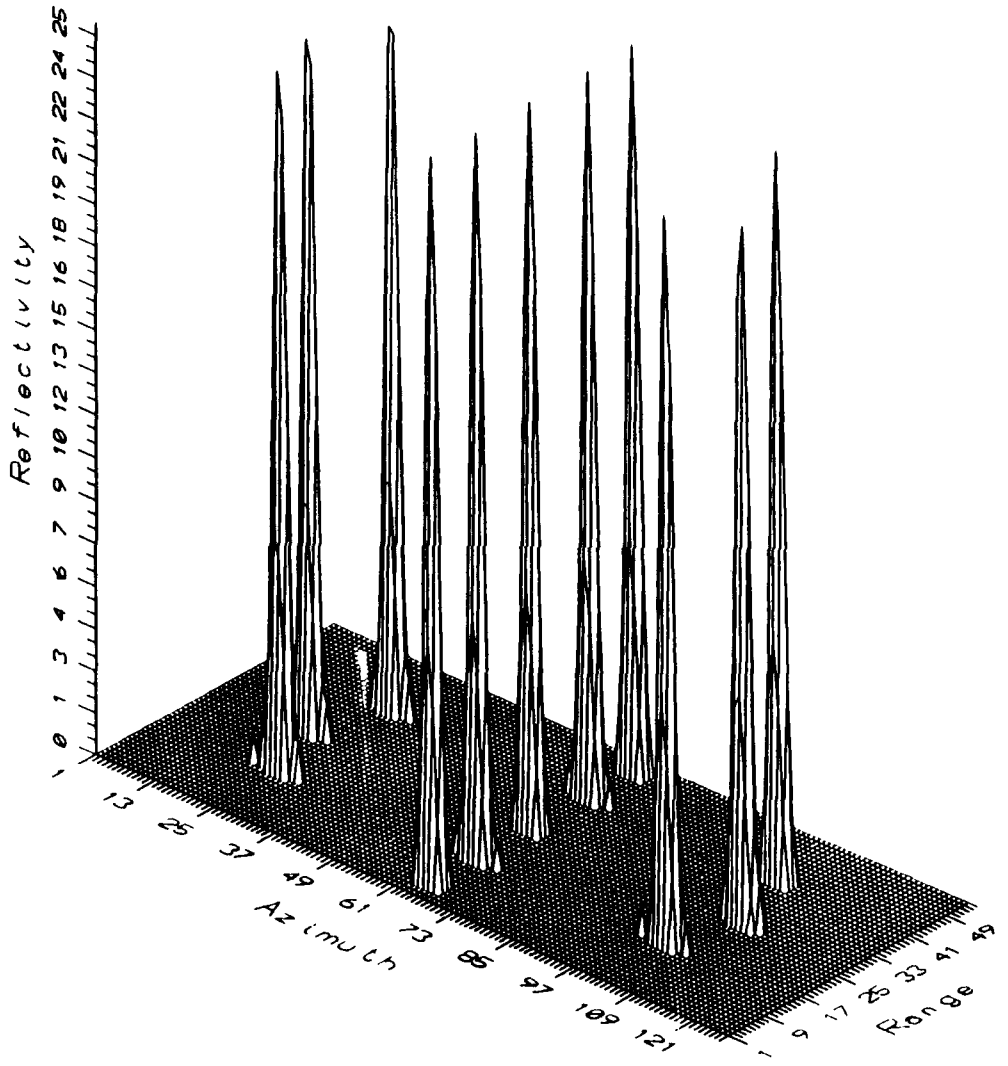

Figure 12.1 Footprint reflectivity vs azimuth & range (Tables 12.1 & 12.2).

the inputs defined in Tables 12.1 and 12.2a. Figure 12.2 shows a side view of the 3–D display in Fig. 12.1. This demonstrates that impulse invariance has been achieved within ±4% over the entire footprint, and it indicates that antenna and range compensation work very well. Figure 12.3 is a reflectivity contour plot obtained by using TOPO [7] on the same file FRRDPG.GRD. This figure shows that: (1) the scatterers are detected at the correct locations, and (2) the azimuth resolution is now comparable to range resolution.

Even though the footprint defined by Table 12.1 contains eleven scatterers defined by Tables 12.2a and 12.2b, the reflectivity display can be restricted to a single scatterer by shrinking the size of the display window (WPAR) in CRRDPI.INP. Table 12.3 is identical to Table 12.1 except for a smaller window about the center of the footprint. The output reflectivity for the run that uses Tables 12.3 and 12.2a is shown in Fig. 12.4. To demonstrate the effect of scatterer height, h_i = 3 meters is entered for the central scatterer as shown in Table 12.4. The reflectivity displays for the run using Tables 12.3 and 12.4 are in Figs. 12.5a and b. This experiment demonstrates: (1) the utilization of the window display, and (2) the effect of scatterer height. Indeed a scatterer height of 3m has the effect of reducing range and increasing doppler by about one cell.

13. Demonstration of CRRDP Simulation

The examples in this section use the same inputs as those defined for FRRDP in the previous section. As before, the system and the clutter parameters are in files CRRDPI.INP and CLUTINFI.INP, respectively, and are entered interactively using WINPUT. The examples in this section are similar to those in the previous section except that the outputs are absolute reflectivity maps versus absolute x– y coordinates. A demonstration diskette (DEMO #2) is included in [8]

All the 3–D displays in this section are for a 312° rotation about the z–axis

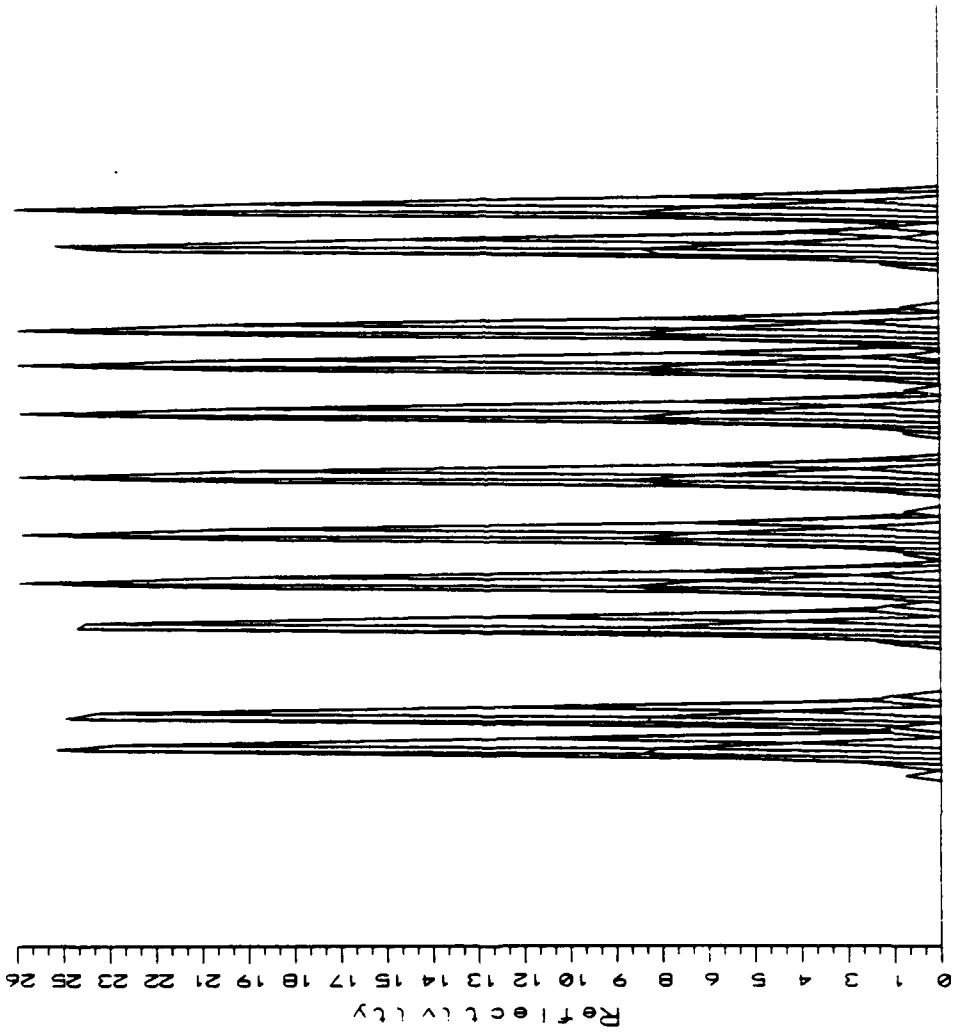

Figure 12.2. Side view of the 3-D plot in Fig. 12.1.

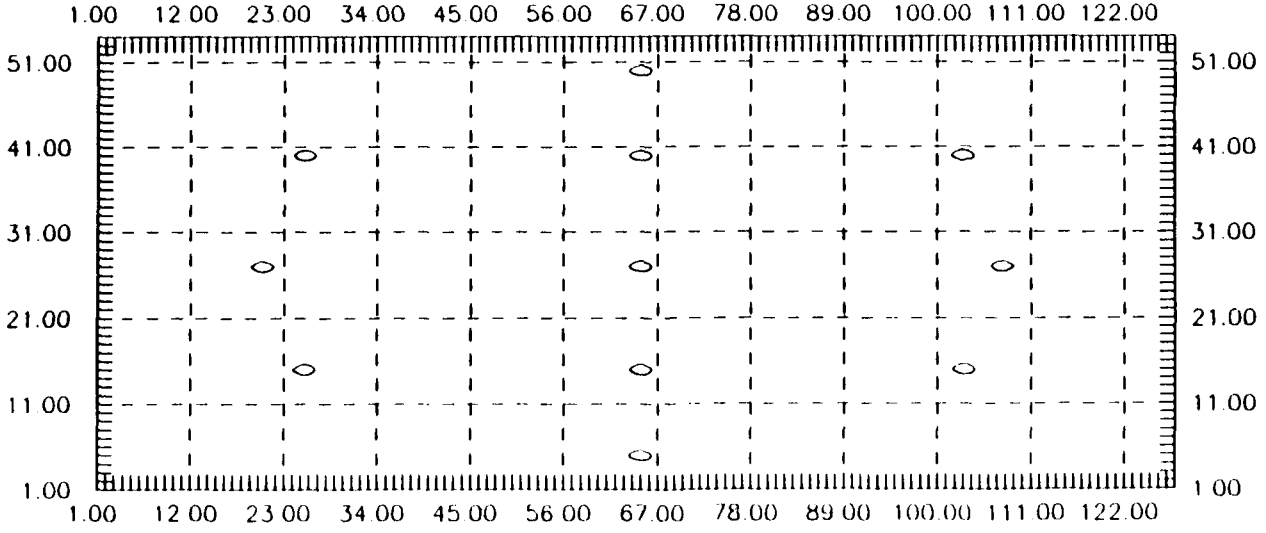

Figure 12.3 Footprint reflectivity contour map from FRRDP (Tables 12.1 & 12.2)

Table 12.3. Data set #2 for file CRRDPI.INP

.0100000	TIMCNT=central time
.0200000	TIMDWL=observation interval
150.0000000	VELOM=seeker velocity
.0000000	ALFM=seeker azimuth angle
1.2217300	BETM=seeker elevation angle
.0000000	WMPOSC(1)=seeker x-coordinate
7.0000000	WMPOSC(2)=seeker y-coordinate
900.0000000	WMPOSC(3)=seeker z-coordinate
516.2463000	WQ9(1)=footprint center x-coordinate
744.2216000	WQ9(2)=footprint center y-coordinate
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
1.0000000	DELRAN=range bin
6.0000000	FFNBIT=# FFT bits
.7500000	WPAR(1)=rectangular grid increment
.1000000	WPAR(2)=mapping threshold
506.0000000	WPAR(3)=x-min display window
734.0000000	WPAR(4)=y-min display window
526.0000000	WPAR(5)=x-max display window
754.0000000	WPAR(6)=y-max display window
.0000000	XIFXMT=transmitter phase
4.0000000	SELWIN=define FFT window
3.1415900	BETKAI=Kaiser window parameter
2.0000000	SELDIS=define dish simulation
1.0000000	BITINT=define FFT interpolation

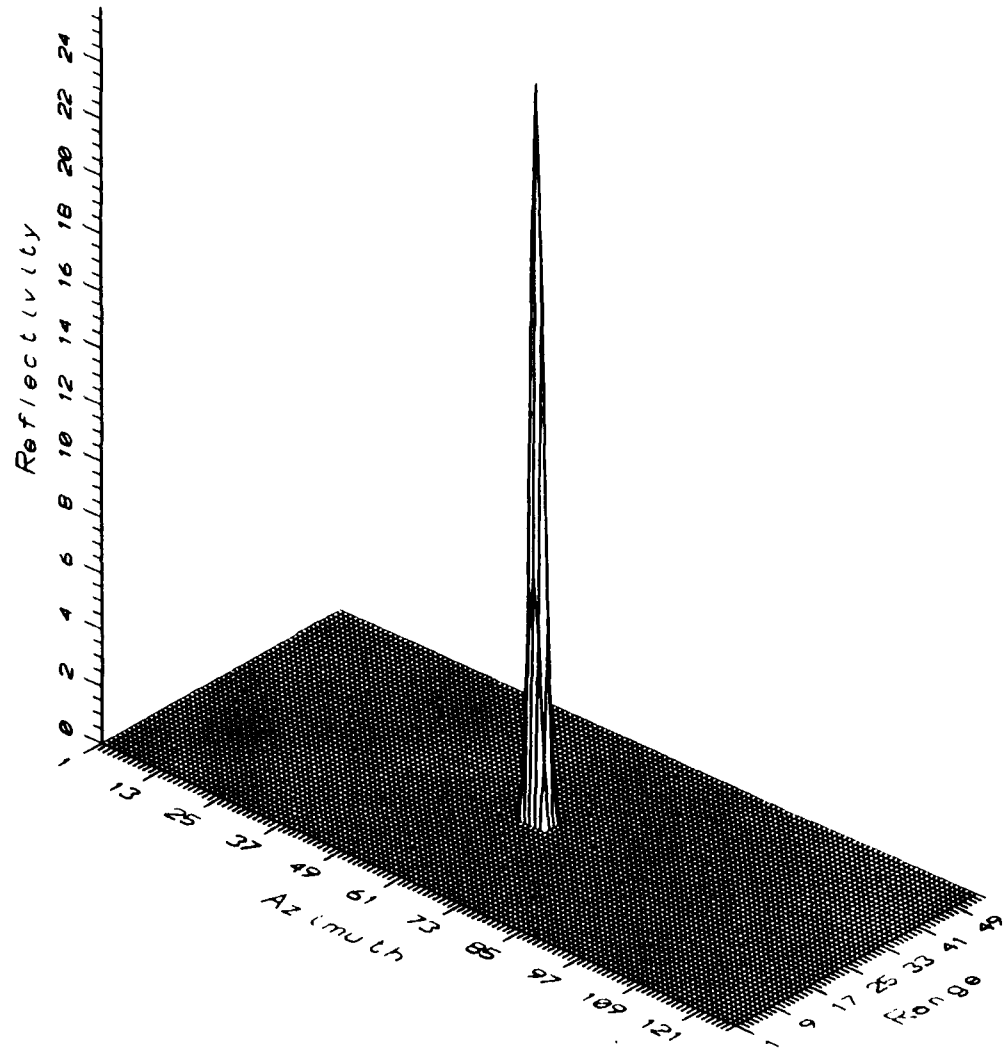

Figure 12.4 Footprint reflectivity for a central scatterer (Tables 12.3 & 12.2)

Table 12.4. Data set #2 for input file CLUTINFI.INP.

```
-34.00000000  W1=3* #scatt.+1,pos. if (xi,yi),neg. if (mui,rgi)
    .0000000  X1=x(1st scat), or mu1
1272.7910000  Y1=y(1st scat), or rg1
   3.0000000  Z1=z(1st scat) = hbar1
   -.0304062  X1=x(2nd scat), or mu2
1272.7910000  Y1=y(2nd scat), or rg2
    .0000000  Z1=z(2nd scat) = hbar2
    .0304062  X1=x(3rd scat), or mu3
1272.7910000  Y1=y(3rd scat), or rg3
    .0000000  Z1=z(3rd scat) = hbar3
    .0000000  X1=x(4th scat), or mu4
1285.7910000  Y1=y(4th scat), or rg4
    .0000000  Z1=z(4th scat) = hbar4
   -.0267718  X1=x(5th scat), or mu5
1285.7910000  Y1=y(5th scat), or rg5
    .0000000  Z1=z(5th scat) =hbar5
    .0267718  X1=x(6th scat), or mu6
1285.7910000  Y1=y(6th scat), or rg6
    .0000000  Z1=z(6th scat) = hbar6
    .0000000  X1=x(7th scat), or mu7
1260.7910000  Y1=y(7th scat), or rg7
    .0000000  Z1=z(7th scat) = hbar7
   -.0273642  X1=x(8th scat), or mu8
1260.7910000  Y1=y(8th scat), or rg8
    .0000000  Z1=z(8th scat) = hbar8
    .0273642  X1=x(9th scat), or mu9
1260.7910000  Y1=y(9th scat), or rg9
    .0000000  Z1=z(9th scat) = hbar9
    .0000000  X1=x(10th scat), or mu10
1295.7910000  Y1=y(10th scat), or rg10
    .0000000  Z1=z(10th scat) = hbar10
    .0000000  X1=x(11th scat), or mu11
1250.7910000  Y1=y(11th scat), or rg11
    .0000000  Z1=z(11th scat) = hbar11
```

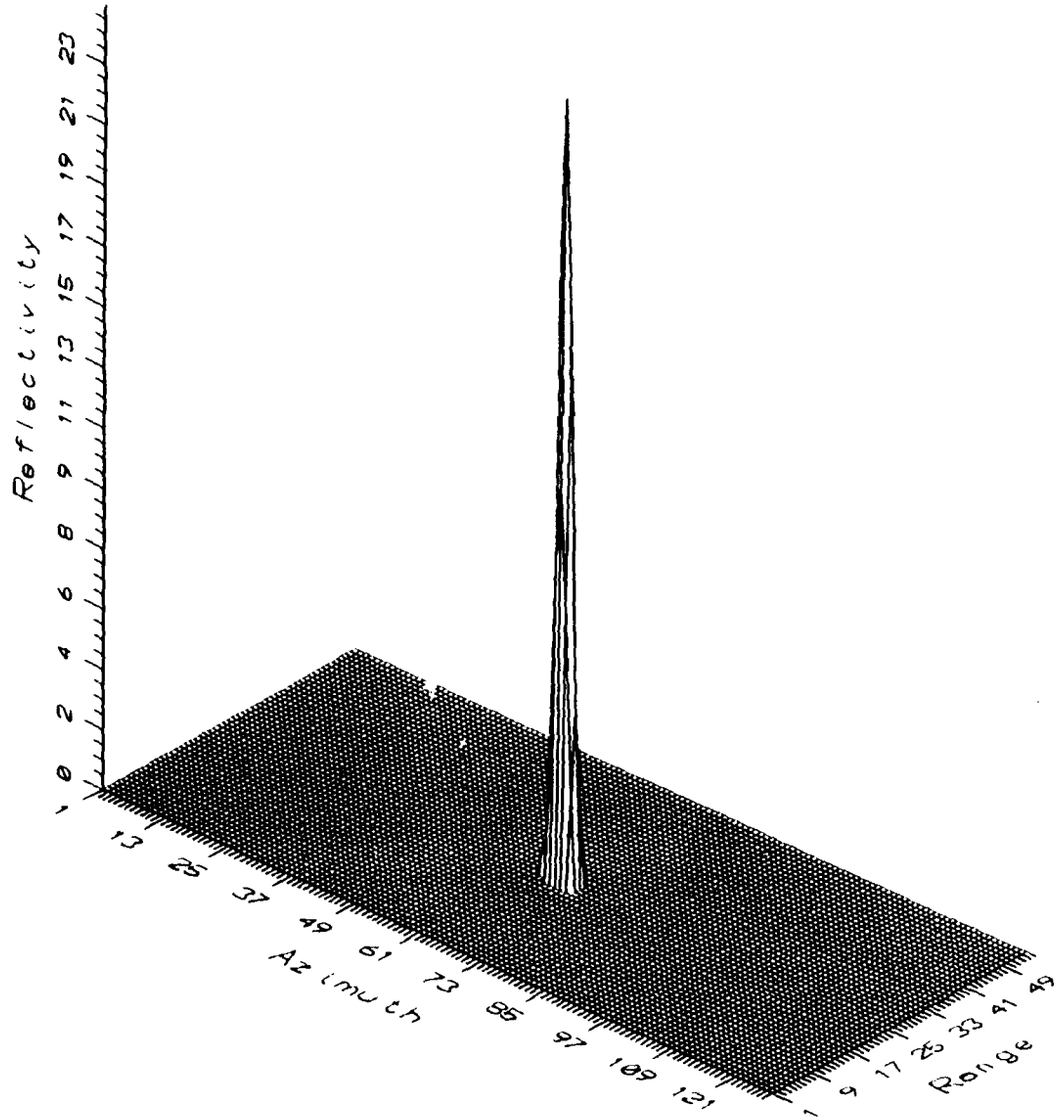

Figure 12.5a Footprint reflectivity for a central scatterer of height=3m.

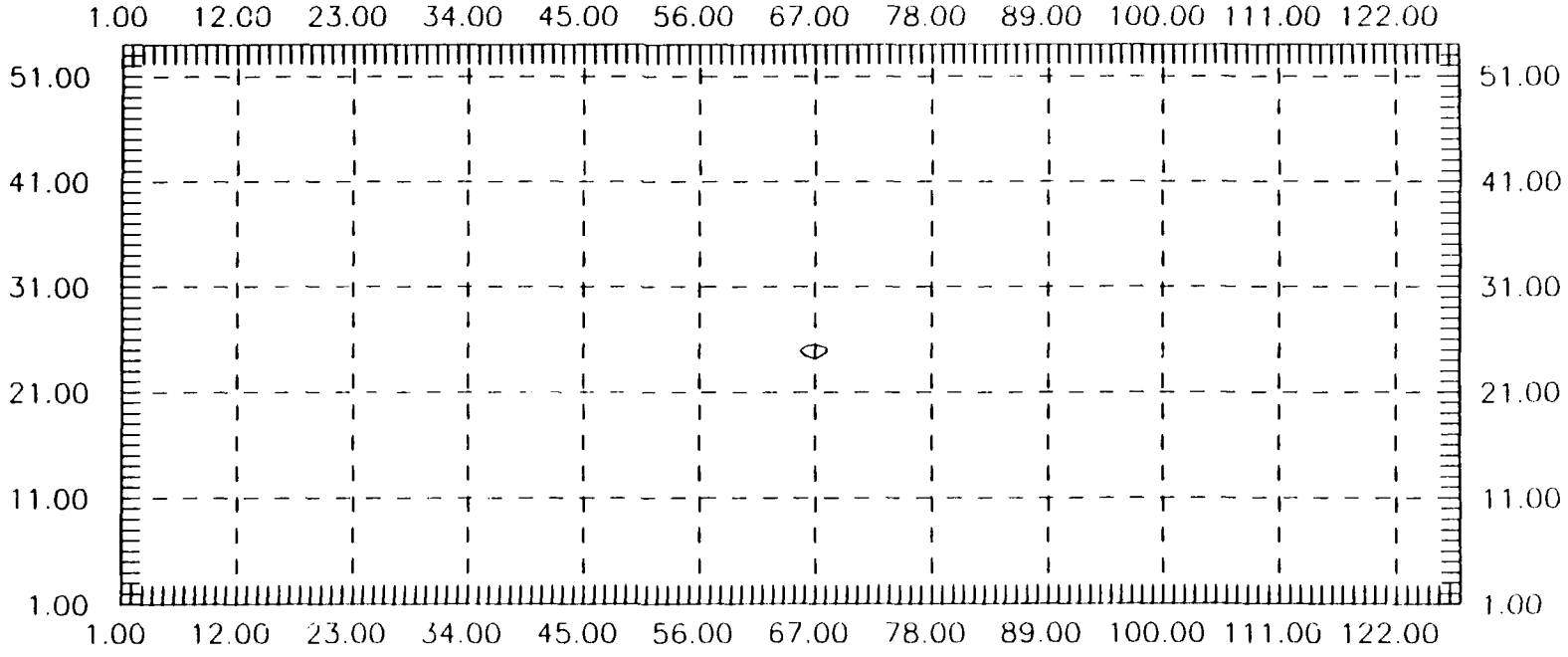

Figure 12.5b Footprint reflectivity contour for a central scatterer, height=3m

(entered in SURF). Figures 13.1a and b shows the 3–D reflectivity and the contour map versus absolute x–y coordinates obtained by using SURFER [7] on the file CRRDPG.GRD created by CRRDP for the inputs defined in Tables 12.1 and 12.2a. Figure 13.2 shows a side view of the 3–D plot in Fig. 13.1a. As expected the invariant mapping technique produces: (1) a small widening of the pulse, and (2) a slight increase in the variation over the footprint; this is due to different configurations of the FFT trapezoidal cells relative to the rectangular grid. However, one can clearly observe that: (1) the peak reflectivity is nearly invariant, (2) the maximum variation over the x–y map is within ±5.5%, which should be acceptable for most applications, and (3) the volume invariance of the impulse response is nearly perfect; this provides some compensation during detection because the shorter impulses are wider than the taller impulses.

As earlier one can reduce the selected window display to a single scatterer output by shrinking the display window (WPAR). Figure 13.3 is for the inputs specified by Tables 12.2a and 12.3. The effect of scatterer height is demonstrated by running CRRDP with the inputs defined by Tables 12.2a and 12.4.

All the results so far were for rectangular grids with grid size ($\Delta x = \Delta y = 0.75$m). The performance of the invariant mapping technique becomes even better when the grid increment is reduced to 0.6m. Figure 13.4 shows the reflectivity for the input files given in Tables 13.1 and 12.2a. Only eight scatterers are showing because we were forced to decrease the size of the of the display window due to the restriction of SURFER to a maximum of 10000 points. The impulses are smoother and the peak variation over the display is now only ±4%. Conversely, if the size of the grid is increased to 1m, the impulse variation will increase to about ±10%. Note that in all cases the average peak value remain constant, which confirms the invariance of the mapping technique.

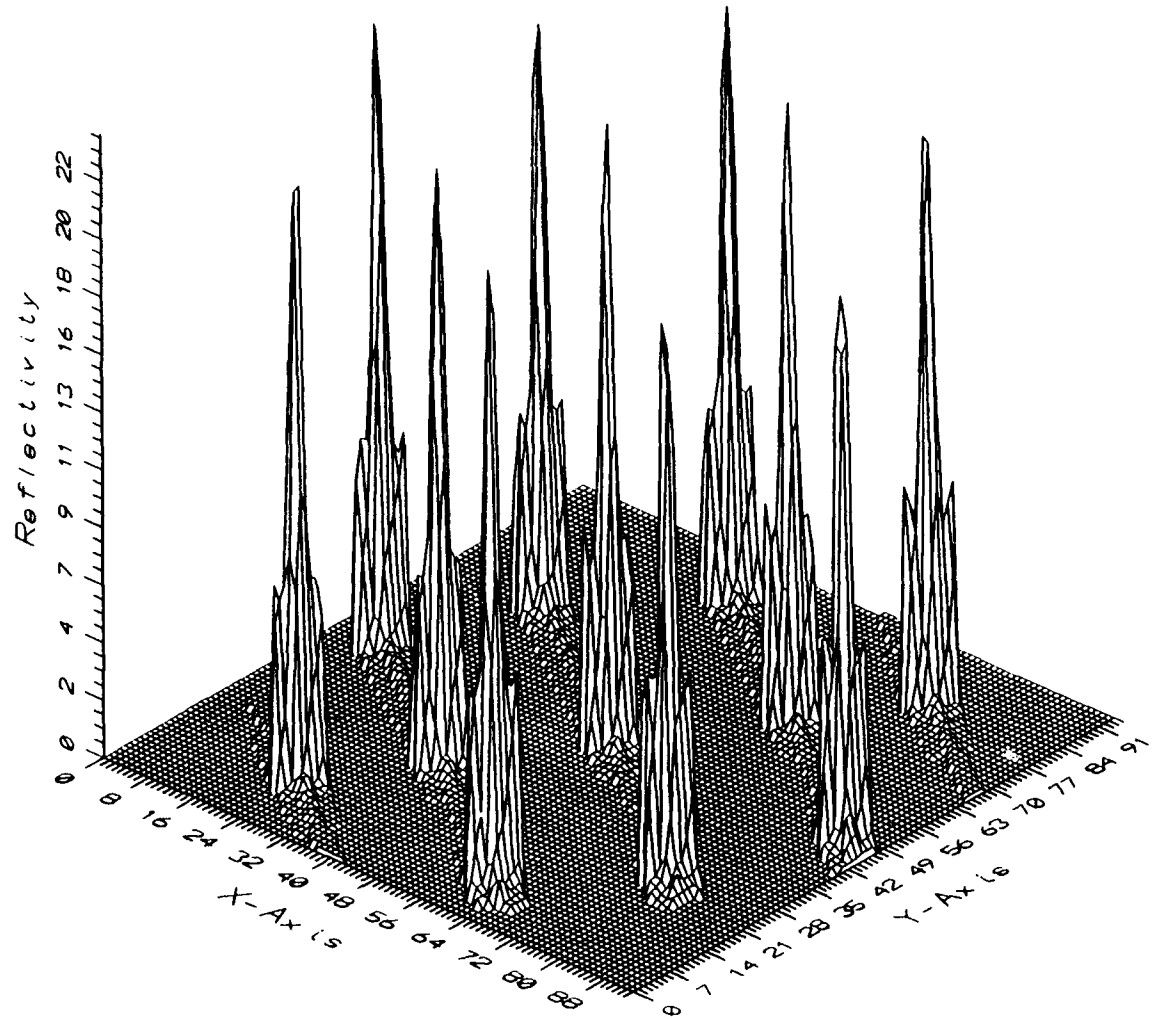

Figure 13.1a Absolute reflectivity map from CRRDP (Tables 12.1 & 12.2).

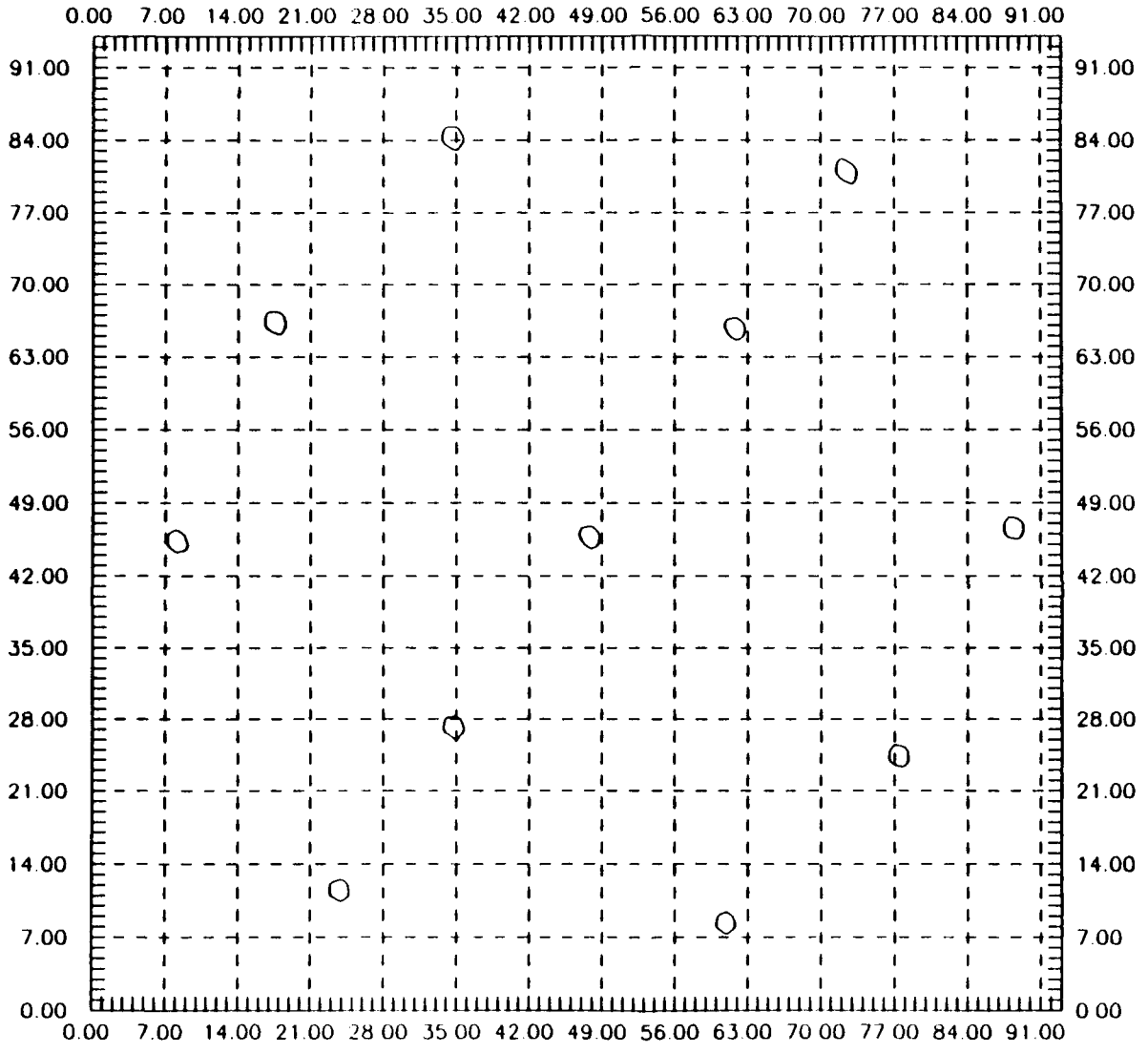

Figure 13.1b Absolute contour map from CRRDP (Tables 12.1 & 12.2).

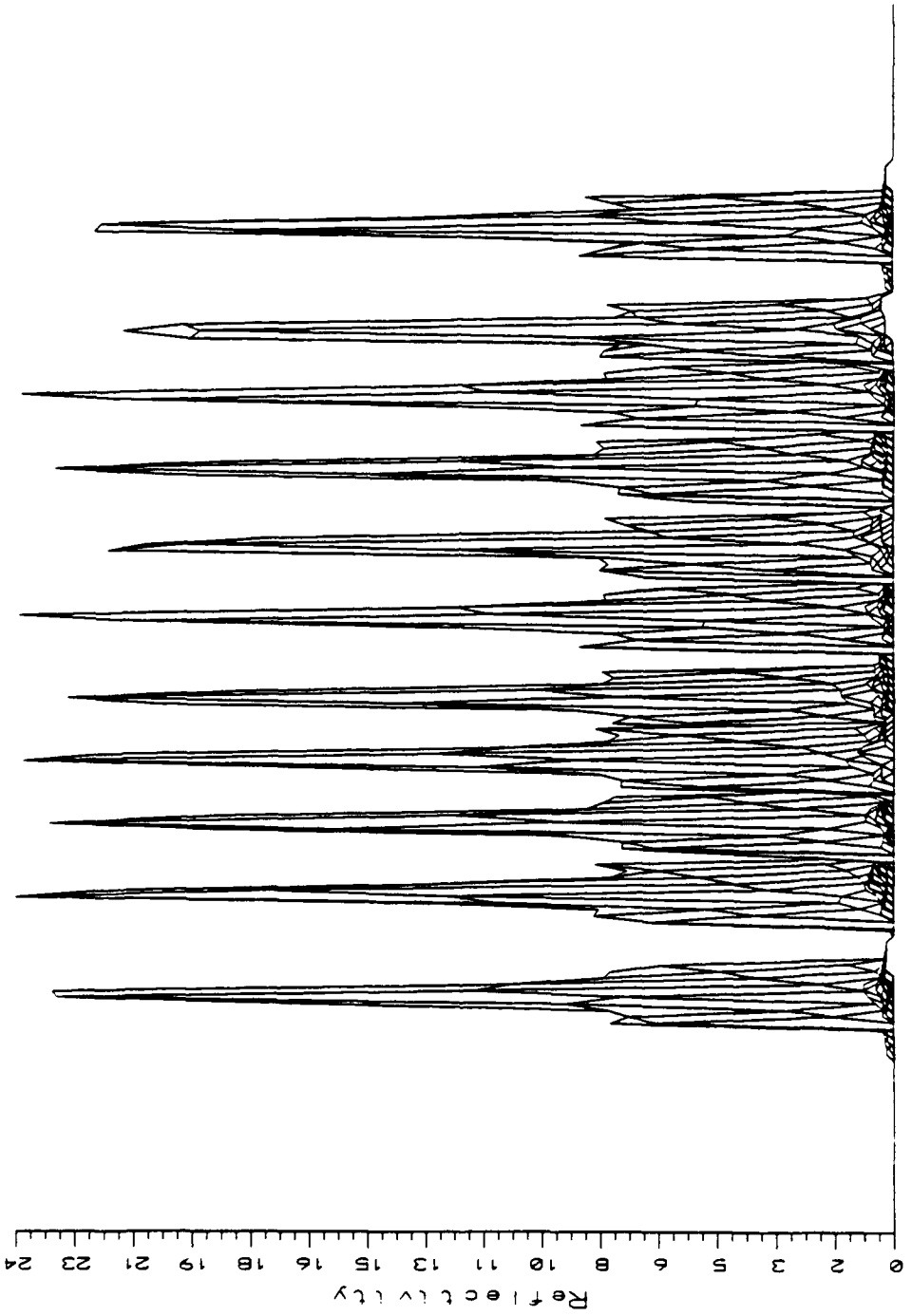

Figure 13.2. Side view of the 3-D plot in Fig. 13.1a.

Figure 13.3. Absolute reflectivity map, central scatterer (Tables 12.3 & 12.9).

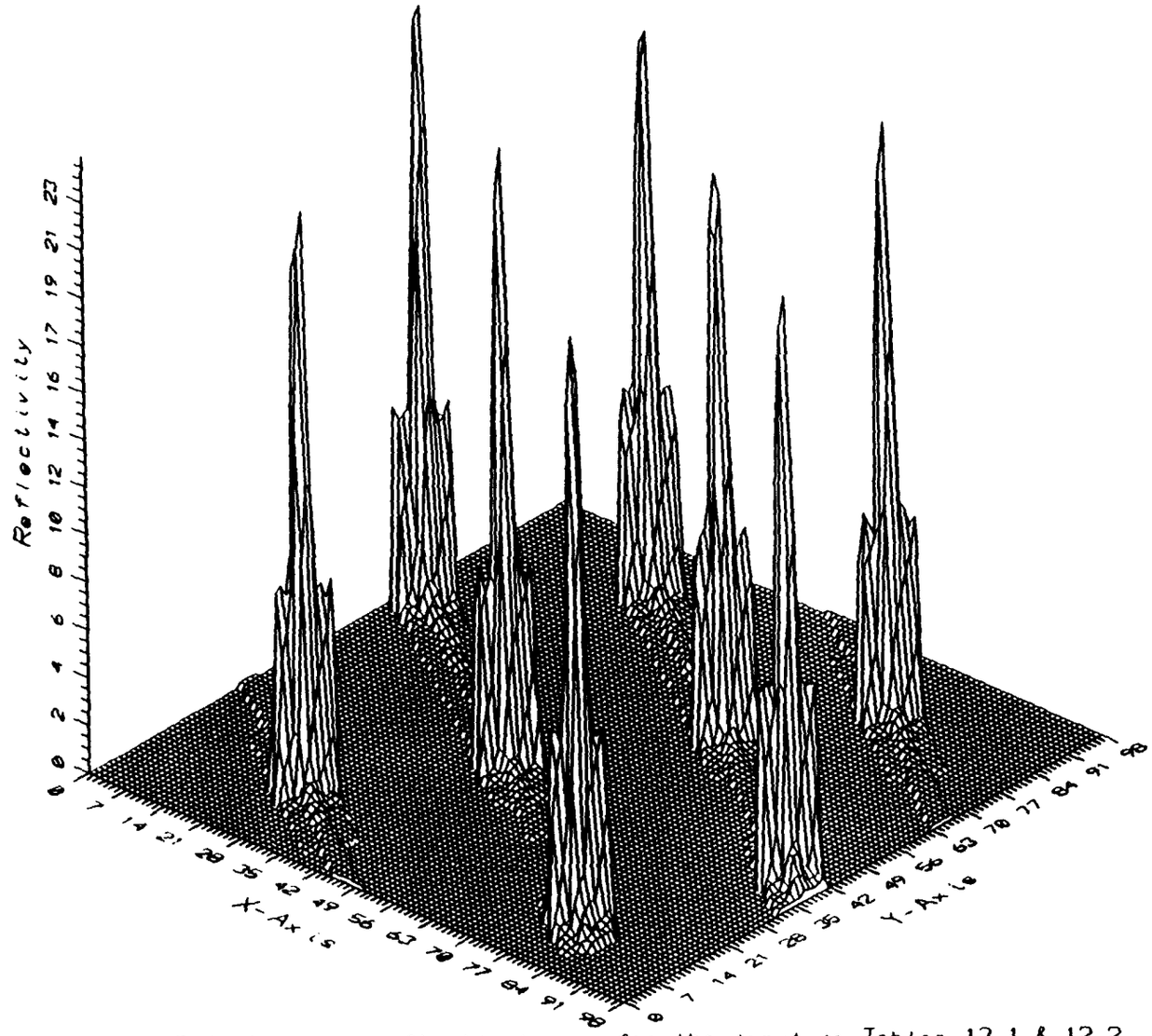

Figure 13.4. Absolute reflectivity map for the input in Tables 13.1 & 12.2.

Table 13.1. Data set #3 for file CRRDPI.INP

.0100000	TIMCNT=central time
.0200000	TIMDWL=observation interval
150.0000000	VELOM=seeker velocity
.0000000	ALFM=seeker azimuth angle
1.2217300	BETM=seeker elevation angle
.0000000	WMPOSC(1)=seeker x-coordinate
7.0000000	WMPOSC(2)=seeker y-coordinate
900.0000000	WMPOSC(3)=seeker z-coordinate
516.2463000	WQ9(1)=footprint center x-coordinate
744.2216000	WQ9(2)=footprint center y-coordinate
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
.0000000	DUMMY=future use
1.0000000	DELRAN=range bin
6.0000000	FFNBIT=# FFT bits
.7500000	WPAR(1)=rectangular grid increment
.1000000	WPAR(2)=mapping threshold
491.0000000	WPAR(3)=x-min display window
721.0000000	WPAR(4)=y-min display window
550.0000000	WPAR(5)=x-max display window
780.0000000	WPAR(6)=y-max display window
.0000000	XIFXMT=transmitter phase
4.0000000	SELWIN=define FFT window
3.1415900	BETKAI=Kaiser window parameter
2.0000000	SELDIS=define dish simulation
1.0000000	BITINT=define FFT interpolation

14. Summary, Conclusions and Recommendation for Future Work

This report extends and upgrades the computer simulation for Range Relative Doppler Processing (RRDP) and invariant mapping for MM Wave Seekers. The main program and most of the subroutines are significantly different from those in [1,2]. Major improvements include: (1) a more general geometry where the trajectory is not restricted to a vertical plane, (2) an exact simulation for the dish antenna, (3) a more exact signal synthesis that includes the effect of scatterer height, (4) the development of an interactive input subroutine which reads, displays on screen, prompts for online changes, and updates the data file, (5) a much better compensation for antenna gain and range attenuation, (5) a reduction of the peak reflectivity variation due to FFT quantization through frequency interpolation, (7) a new, better, and more efficient invariant mapping algorithm, and (8) online 3–D and contour graphic displays for both types of reflectivity maps.

Two main programs were developed, FRRDP and CRRDP. The first, FRRDP, generates the file FRRDPG.GRD which SURFER uses to display the footprint reflectivity map versus azimuth and range. This program should be useful when one needs to perform target detection within one footprint. The second program, CRRDP, generates the file CRRDPG.GRD which SURFER uses to display the reflectivity versus the absolute x–y grid. Our invariant mapping technique is used to transfer the information from the footprint to the absolute map. This program should be useful to accumulate the information of a sequence of footprints prior to final detection.

Examples show various footprint reflectivity maps obtained by using FRRDP. These examples demonstrate the following: (1) the scatterers are detected at the proper locations, (2) impulse invariance has been achieved over the entire footprint, and (3) the effect of scatterer height is significant and has been properly simulated.

Examples also show various absolute x–y reflectivity maps obtained by using CRRDP. They demonstrate the performance of Invariant Mapping: (1) the peaks remain practically the same in value and location, (2) impulse invariance has been achieved over the absolute map within a ±5.5% variation, and (3) the volume invariance of the impulses is nearly perfect. Most of the examples were run with a rectangular grid of size $\Delta x = \Delta y = 0.75$ m. Increasing the grid size to 1m will increase the peak variation over the absolute map to about ±10%. On the other hand, a reduction of the grid size to 0.6m reduces the variation to about ±4%, but it increases the computation requirements. At the present time the invariant mapping involves only a single footprint but it will be easy to extend it to a sequence of footprints. One expects that the impulse invariance over the absolute map would become even better due to the smoothing effect of footprint integration.

Recommendations for future work include the following: (1) using realistic target models as input, (2) using realistic clutter models with or without targets, (3) incorporating pulse compression in the simulation for a significant increase in the signal–to–noise ratio, (4) adding detection and pattern recognition to the programs, and (5) extending the simulation to multiple polarization.

References

[1] Polge, R. J., Mahafza, B. R., and Kim, J. G.," Computer Simulation for a MM Wave Seeker Including SAR Processing and Invariant Maping", prepared for Georgia Institute of Technology under subcontract # A–4521–S1 in support of U. S. Army Missile Command Prime Contract # DAAH01–84–DA029, D.O. 0088 UAH Technical Report, May 1987. SECRET.

[2] Polge, R. J., Mahafza, B. R., and Kim, J. G., "Increasing Azimuth and Elevation Resolution of MM Wave Seeker Systems Using Coherent or Noncoherent Range Relative Doppler Processing (RRDP) with Constant or Linear Frequency Modulation and Invariant Mapping", prepared by the University of Alabama in Huntsville for the U. S. Army Missile Command under Contract DAAH01–87–D–0021, D.O. 18, February 1988.

[3] Rulf, Benjamin and Robertshaw, Gregory A., "Understanding Antennas for Radar, Communications, and Avionics", Van Nostrand, Reinhold, N.Y., 1987.

[4] Balanis, Constantine A., "Antenna Theory Analysis and Design," Harper and Row, N.Y., 1982.

[5] Stanley, W. D., Dougherty, G. R., and Dougherty, R., "Digital Signal Processing," 2nd Ed., Reston Publishing, 1984.

[6] Hamming, R. W., "Digital Filters," 2nd Ed., Prentice Hall, 1983.

[7] SURFER Reference Manual, by Golden Software Inc, 807 14th Street, P. O. Box 281, Golden, Colorado, 80402.

[8] Polge, R. J., Mahafza, B. R., and Kim, J. G., "User Manual of the Range Relative Doppler Processing Simulation for MM Wave Seekers," prepared by the University of Alabama in Huntsville for the U. S. Army Missile Command under Contract D.O. 0064, DAAH01–87–D–0021, January 1989.

APPENDIX A

Relevant Footprint Information For $\vec{m} = \{d_x, d_y, h\}$

A.1 Footprint center $\vec{q}^* = \vec{q}(t_c)$, see Fig. A.1.

Rectangular coordinates of footprint center = $\{x_q, y_q, 0\}$

range: $r^* = \sqrt{\quad}(h^2 + (x_q - d_x)^2 + (y_q - d_y)^2)$

azimuth angle: $\alpha^* = \tan^{-1}[(x_q - d_x)/(y_q - d_y)]$ (A.1)

elevation angle = $\beta^* = \cos^{-1}\left[\frac{h}{r^*}\right]$

A.2 Range bounds and indexing, see Fig. A.1.

Minimum range: $\text{Ranmn} = \frac{h}{\cos(\beta^* - \theta_H)}$

Maximum range: $\text{Ranmx} = \frac{h}{\cos(\beta^* + \theta_H)}$ (A.2)

θ_H = half beamwidth

Range index for $\vec{q}^*$: $j^* = \text{INT}[(r^* - \text{Ranmn} + \Delta r/2)/\Delta r]$

Number of range cells: $2j^*$

A.3 Information for the jth range cell

median range: $r_j^* = r^* + (j - j^*)\ \Delta r$

elevation angle: $\beta_j^* = \cos^{-1}\left[\frac{h}{r_j^*}\right]$ (A.3)

maximum relative elevation (see Fig. A.2): $\varepsilon mx = \beta j^* - \cos^{-1}\left[\frac{h}{r_j^* - \Delta r/2}\right]$

maximum relative azimuth (see Fig. A.3): $\mu mx = \cos^{-1}\left[\frac{\cos\theta_H - \cos\beta_j^* \cos\beta^*}{\sin\beta_j^* \sin\beta^*}\right]$

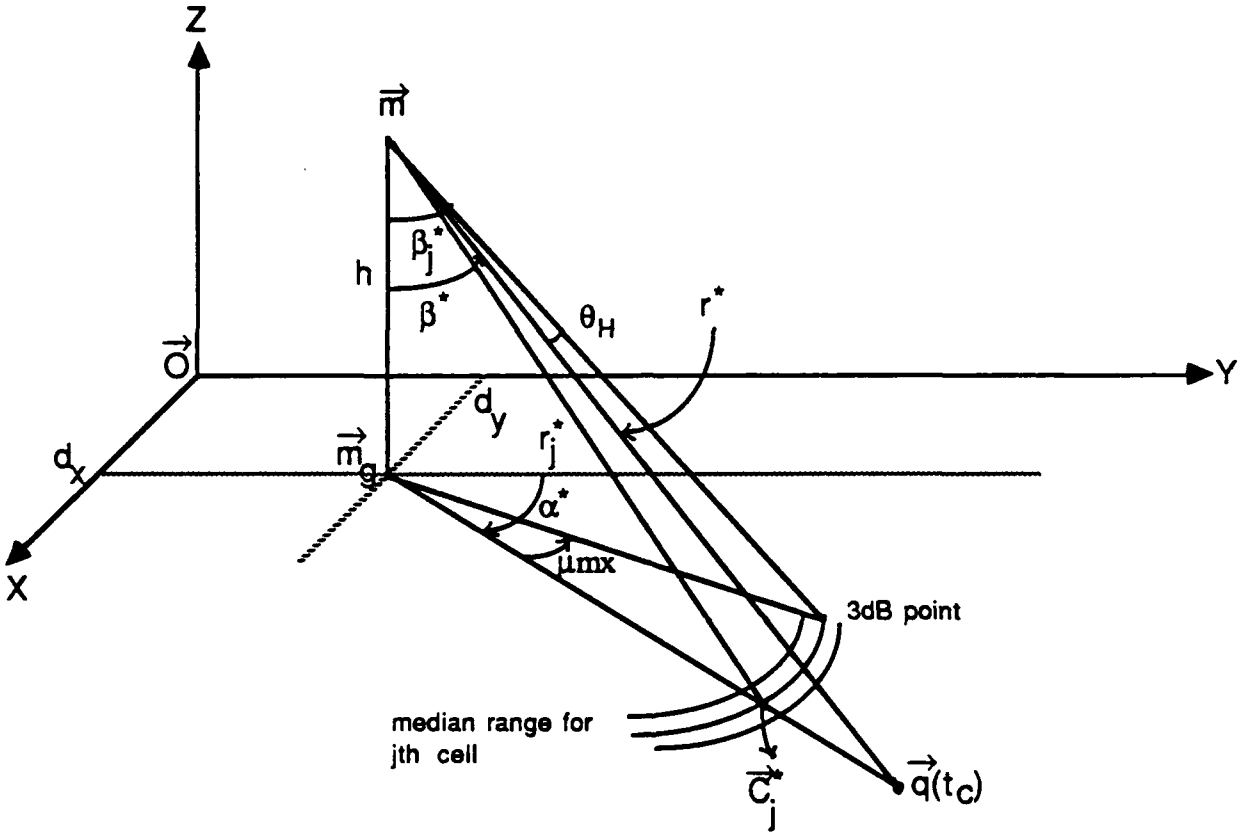

Figure A.1 Footprint center,range bounds,and jth range cell.

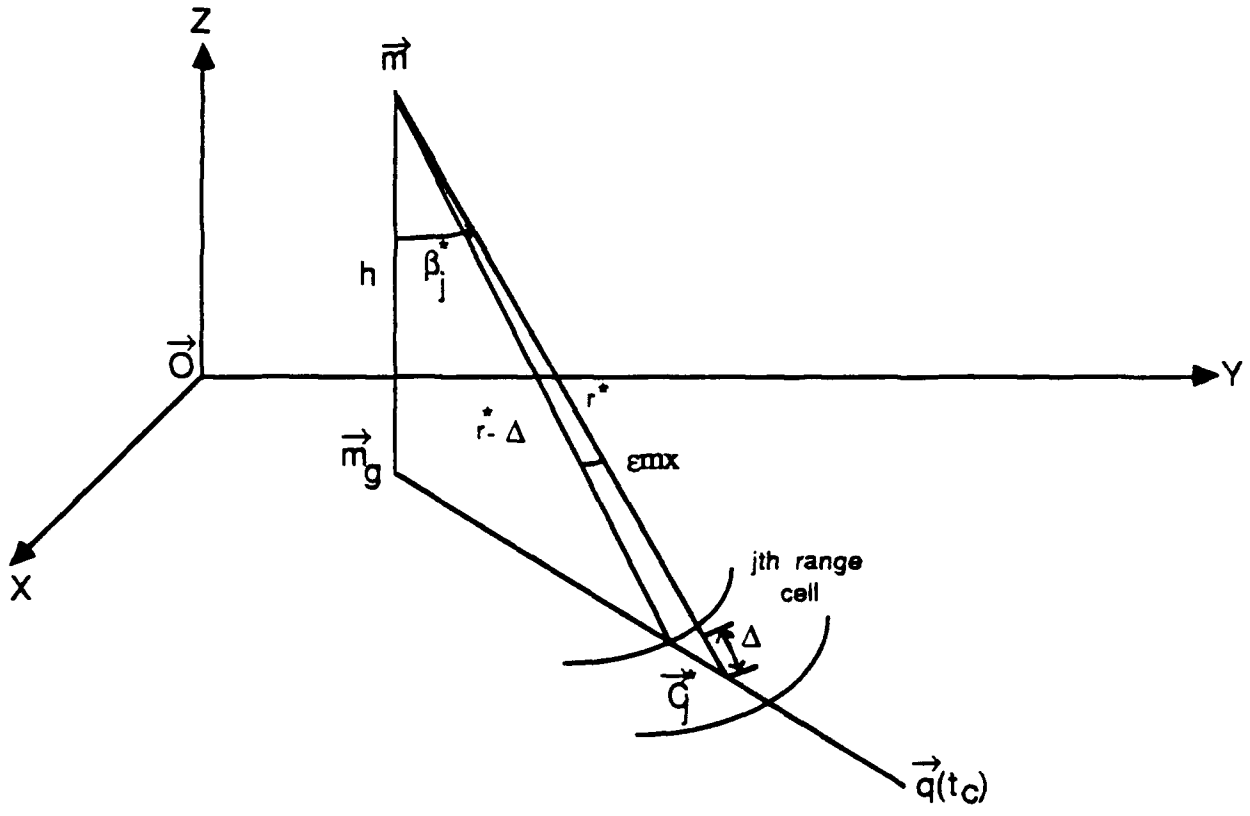

Figure A.2 Maximum relative elevation εmx ($\Delta = \Delta r/2$).

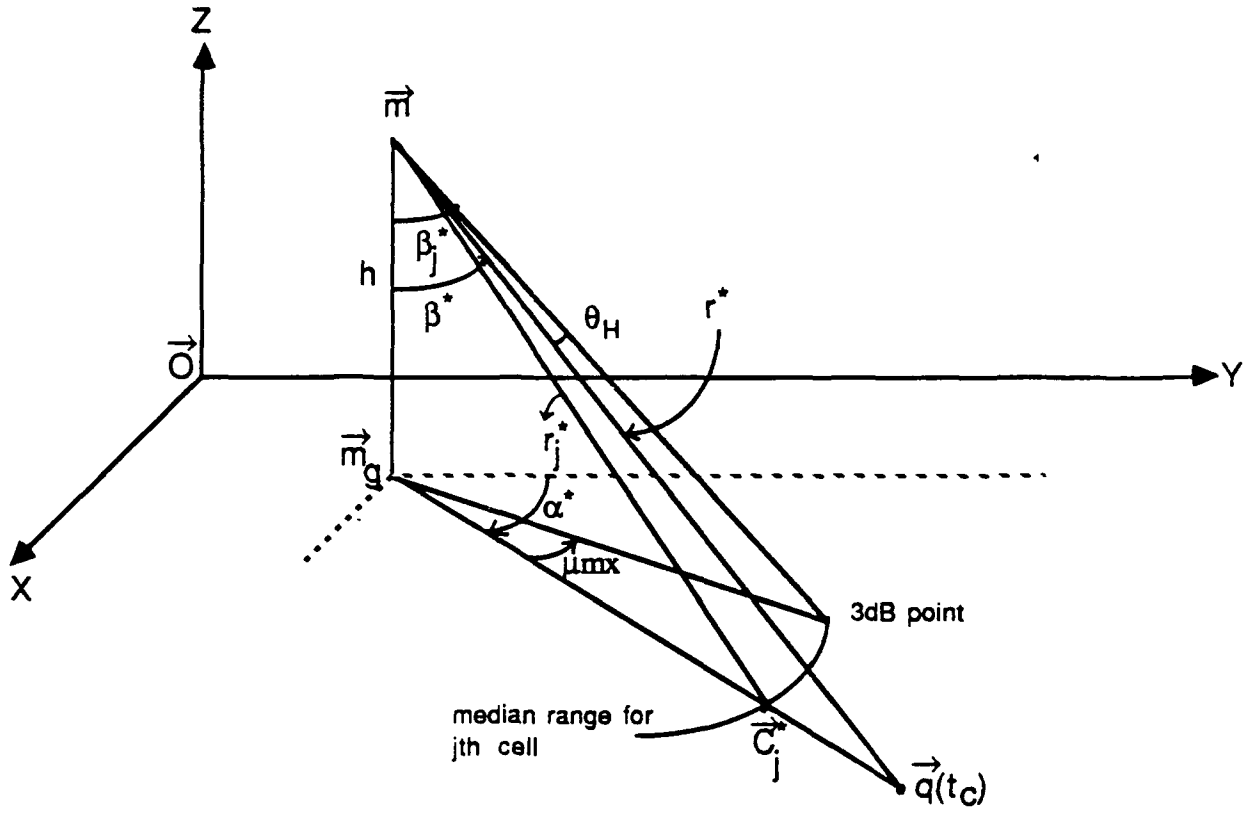

Figure A.3 Maximum relative azimuth angle μ_{mx,θ_H} =half beamwidth.

APPENDIX B

Relevant Information for a Scatterer $\vec{C}_i$ (see Fig. B.1).

B.1 Computation of $\{r_{gi}, r_i, \alpha_i, \beta_i, \mu_i\}$ from $\{\vec{m}, \vec{C}_i, \alpha^*\}$

$(\vec{m}_g, \vec{C}_{gi})$ are the vertical projections of $(\vec{m}, \vec{C}_i)$

ground range: $GR = |\vec{C}_{gi} - \vec{m}_g| = \sqrt{\quad} [(x_i - d_x)^2 + (y_i - d_y)^2]$

range to $\vec{C}_{gi} = r_{gi} = |\vec{C}_{gi} - \vec{m}| = \sqrt{\quad} [GR^2 + h^2]$

slant range to $\vec{C}_i$: $r_i = |\vec{C}_i - \vec{m}| = \sqrt{\quad} [GR^2 + (h - h_i)^2]$

range cell index: $j = INT[(r_i - r^* + \Delta r/2)/\Delta r] + j^*$ (B.1)

elevation angle: $\beta_i = \tan^{-1}\left[\frac{GR}{h}\right]$

azimuth angle: $\alpha_i = \tan^{-1}[(x_i - d_x)/(y_i - d_y)]$

relative azimuth increment: $\mu_i = \alpha_i - \alpha^*$

B.2 Computation of $\{r_i, \alpha_i, \beta_i, x_i, y_i\}$ from $\{r_{gi}, \mu_i, \varepsilon_i, h_i, \alpha^*\}$

ground range: $GR = \sqrt{\quad} (r_{gi}^2 - h^2)$

slant range to $\vec{C}_i$: $r_i = \sqrt{\quad} [GR^2 + (h - h_i)^2]$

range cell index j from r_i is as above

azimuth angle: $\alpha_i = \alpha^* + \mu_i$

elevation angle: $\beta_i = \beta_j^* + \varepsilon_i$ (B.2)

x–coordinate: $x_i = GR \sin \alpha_i + d_x$

y–coordinate: $y_i = GR \cos \alpha_i + d_y$

B.3 Antenna Gain for $\vec{C}_i$ at time t.

Antenna angle to $\vec{C}_i$: Ang= Angle$(\vec{q}^* - \vec{a}(t), \vec{C}_i - \vec{a}(t))$

let $w1 = \{a_x(t)\ ,\ a_y(t)\ ,\ a_z(t)\}$

$w2 = \{q_x^*, q_y^*, q_z^*\}$

and $w3 = \{C_{ix}, C_{iy}, C_{iz}\}$

$s = \sin(Ang) = \sqrt{\quad} (1 - DOTCOS(w1,w2,w3)**2)$ (B.3a)

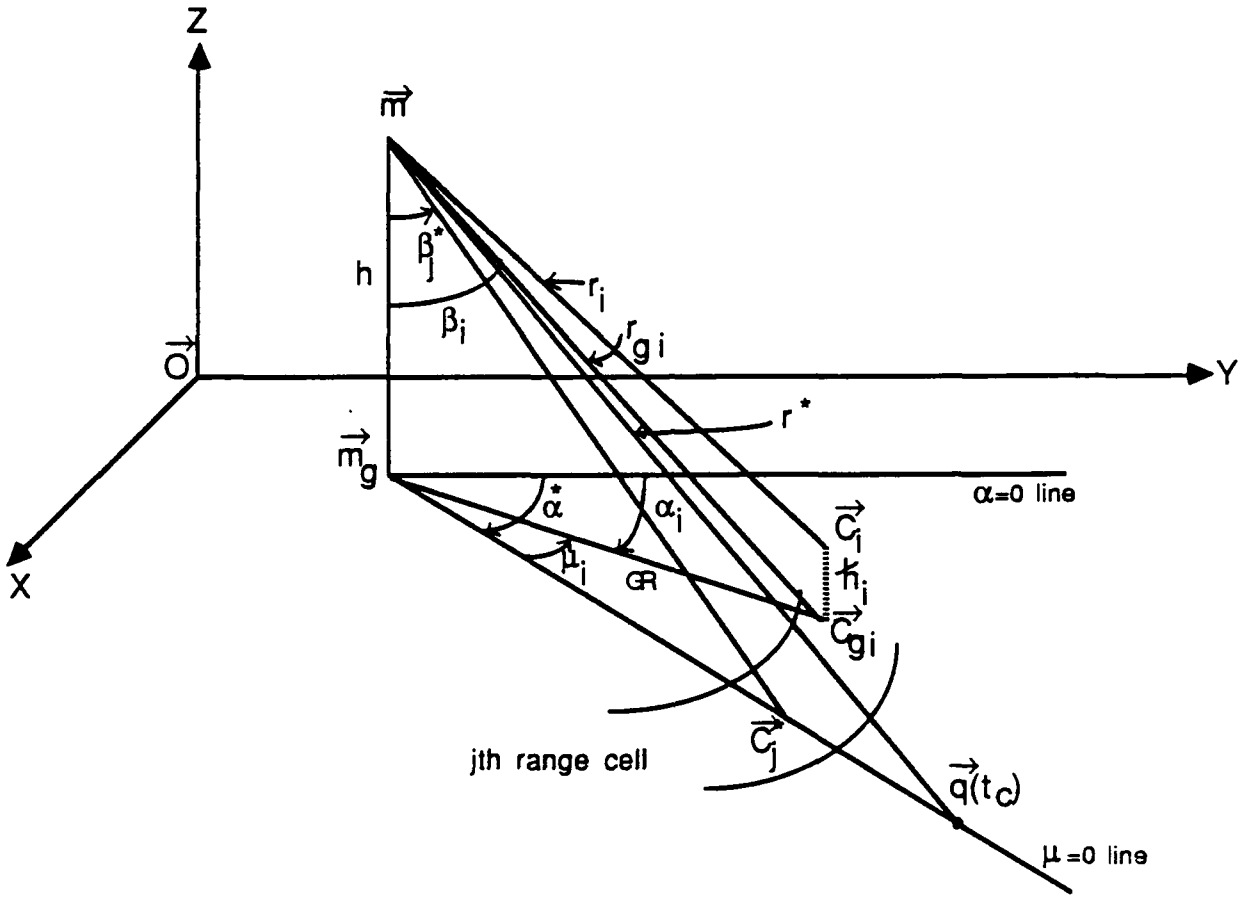

Figure B.1 Fictitious scatterer $\vec{C}_j^*$ and ith scatterer $\vec{C}_i$ in the jth range cell:$\alpha_i=\alpha^*+\mu_i$.

where the function DOTCOS is listed in Appendix E.

One–way Antenna gain = DISH(s,2,DOVL) (B.3b)

where the function DISH is listed in Appendix C, and DOVL is the dish diameter in wavelength units.

The gain is computed at three times

$$\{t = -\frac{D_{ob}}{2} \ , \ t = 0 \ , \ t = \frac{D_{ob}}{2}\} \tag{B.3c}$$

APPENDIX C

Listings for DISH and BESSJ1

These functions are used to compute the dish antenna gain:

```
      FUNCTION DISH(SINBET,IDDISH,DDISH)
C      SINBET=SINE OF OFF-BROADSIDE ANGLE
      EPS=.000001
      IF(ABS(SINBET).LT.EPS) GO TO 1
      PI=3.141592654
      GO TO (1,2,3), IDDISH
    1  DISH=1.
      RETURN
    2  X=SINBET*PI*DDISH
      DISH=ABS(2.*BESSJ1(X)/X)
      RETURN
    3  BET=ASIN(SINBET)
      X=PI*DDISH*BET
      IF(ABS(X).LT.EPS) X=EPS
      DISH=ABS(SIN(X)/X)
      RETURN
      END

      FUNCTION BESSJ1(X)
      REAL*8 Y,P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5,R1,R2,R3,R4,R5,R6,
     *   S1,S2,S3,S4,S5,S6
      DATA R1,R2,R3,R4,R5,R6/72362614232.D0,-7895059235.D0,242396853.1D0
     *,
     *   -2972611.439D0,15704.48260D0,-30.16036606D0/,
     *   S1,S2,S3,S4,S5,S6/144725228442.D0,2300535178.D0,
     *   18583304.74D0,99447.43394D0,376.9991397D0,1.D0/
      DATA P1,P2,P3,P4,P5/1.D0,.183105D-2,-.3516396496D-4,.2457520174D-5
     *,
     *   -.240337019D-6/, Q1,Q2,Q3,Q4,Q5/.04687499995D0,-.2002690873D-3
     *,
     *   .8449199096D-5,-.88228987D-6,.105787412D-6/
      IF(ABS(X).LT.8.)THEN
        Y=X**2
        BESSJ1=X*(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6)))))
     *     /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6)))))
      ELSE
        AX=ABS(X)
        Z=8./AX
        Y=Z**2
        XX=AX-2.356194491
        BESSJ1=SQRT(.636619772/AX)*(COS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y
     *     *P5))))-Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5)))))
     *     *SIGN(1.,X)
      ENDIF
      RETURN
      END
```

APPENDIX D

Details of Taylor Expansion of τ_i

This appendix shows the computation of the Taylor expansion given in (5.9). This Taylor expansion differs in three important ways from that presented in appendix A of [2] part II: (1) it accounts for scatterer's height, (2) the notation is more general, and (3) the derivation is more elegant.

The range r_i is expressed as a function of four incremental variables $\{\hbar,t,\mu,\varepsilon\}$ in (4.4). Our strategy is to expand r_i, then the round trip delay τ_i is obtained through multiplication by 2/c. The notation for the partial derivatives was explained in (4.8), where an upper bar means evaluation for zero increments. In our application the incremental variables are defined by

$$\{v_1,v_2,v_3,v_4\} = \{\hbar,t,\mu,\varepsilon\}. \tag{D.1}$$

It is convenient to introduce a new variable $y = r_i^2$,

$$y = h^2 + h^2 \tan^2(\beta+\varepsilon) + v^2t^2 - 2hvt\{\tan(\beta+\varepsilon)\sin\beta_v \cos(\alpha+\mu-\alpha_v)-\cos\beta_v\} \tag{D.2}$$

where β stands for β_j^*, α stands for α^*, and r stands for r_i in this appendix.

The expression for y is simplified by introducing two new functions, f and g:

$$y = h^2 + h^2f^2 + v^2t^2 - 2hvt\{fg - \cos\beta_v\} \tag{D.3a}$$

$$f = \tan(\beta + \varepsilon) \tag{D.3b}$$

$$g = \cos(\alpha + \mu - \alpha_v)\sin\beta_v \ . \tag{D.3c}$$

The chain rule is used to compute the partial derivatives. It follows that we need all the partial derivatives of r with respect to y.

$$r_y = 0.5y^{-0.5} \tag{D.4a}$$

$$r_{yy} = -0.25y^{-1.5} \tag{D.4b}$$

$$r_{yyy} = 0.375y^{-2.5} \ . \tag{D.4c}$$

Evaluation for zero increments yields

$$\bar{f} = \tan\beta \tag{D.5a}$$

$$\bar{g} = \cos(\alpha - \alpha_v)\sin\beta_v \tag{D.5b}$$

$$\bar{y} = h^2 + h^2\bar{f}^2 \tag{D.5c}$$

$$\bar{r} = \sqrt{\bar{y}} \tag{D.5d}$$

$$\bar{r}_y = 0.5/\bar{r} \tag{D.5e}$$

$$\bar{r}_{yy} = -0.25/\bar{r}^3 \tag{D.5f}$$

$$\bar{r}_{yyy} = 0.375/\bar{r}^5 \ . \tag{D.5g}$$

The expressions for the partial derivatives of f and the corresponding reference values are

$$f_4 = f_\varepsilon = \frac{1}{\cos^2(\beta+\varepsilon)} \qquad \bar{f}_4 = \frac{1}{\cos^2\beta} \tag{D.6a}$$

$$f_{44} = f_{\varepsilon\varepsilon} = \frac{2\sin(\beta+\varepsilon)}{\cos^3(\beta+\varepsilon)} \qquad \bar{f}_{44} = \frac{2\sin\beta}{\cos^3\beta} \tag{D.6b}$$

$$f_{444} = f_{\varepsilon\varepsilon\varepsilon} = \frac{2 + 4\sin^2(\beta+\varepsilon)}{\cos^4(\beta+\varepsilon)} \qquad \bar{f}_{444} = \frac{2 + 4\sin^2\beta}{\cos^4\beta} \ . \tag{D.6c}$$

Similarly, the expressions for the partial derivatives of g and the corresponding reference values are

$$g_3 = g_\mu = -\sin(\alpha+\mu-\alpha_v)\sin\beta_v \qquad \bar{g}_3 = -\sin(\alpha-\alpha_v)\sin\beta_v \tag{D.7a}$$

$$g_{33} = g_{\mu\mu} = -\cos(\alpha+\mu-\alpha_v)\sin\beta_v \qquad \bar{g}_{33} = -\cos(\alpha-\alpha_v)\sin\beta_v \tag{D.7b}$$

$$g_{333} = g_{\mu\mu\mu} = \sin(\alpha+\mu-\alpha_v)\sin\beta_v \qquad \bar{g}_{333} = \sin(\alpha-\alpha_v)\sin\beta_v \ . \tag{D.7c}$$

The expression for the partial derivatives of y and the corresponding reference values are

$$y_1 = -2H - 2v_3 t \qquad \bar{y}_1 = -2h$$

$$y_2 = 2v^2 t - fg + 2Hv_z \qquad \bar{y}_2 = -\bar{f}\,\bar{g} + 2hv_z$$

$$y_3 = -fg_\mu t \qquad \bar{y}_3 = 0$$

$$y_4 = 2f\,f_\varepsilon - f_\varepsilon g t \qquad \bar{y}_4 = 2\bar{f}\,\bar{f}_4$$

$$y_{11} = 2 \qquad \bar{y}_{11} = 2$$

$$y_{12} = -2v_z \qquad \bar{y}_{12} = -2v_z$$

$$y_{13} = 0 \qquad \bar{y}_{13} = 0$$

$$y_{14} = 0 \qquad \bar{y}_{14} = 0$$

$$y_{22} = 2v^2 \qquad \bar{y}_{22} = 2v^2$$

$$y_{23} = -f\,g_\mu \qquad \bar{y}_{23} = -\bar{f}\,\bar{g}_3$$

$$y_{24} = -f_\varepsilon g = -f_4 g \qquad \bar{y}_{24} = -\bar{f}_4 \bar{g}$$

$$y_{33} = -fg_{\mu\mu} t \qquad \bar{y}_{33} = 0$$

$$y_{34} = -f_\varepsilon g_\mu t \qquad \bar{y}_{34} = 0$$

$$y_{44} = 2f_\varepsilon^2 + 2f\,f_{\varepsilon\varepsilon} - f_{\varepsilon\varepsilon} g t \qquad \bar{y}_{44} = 2\bar{f}_4^2 + 2\bar{f}\,\bar{f}_{44}$$

$$y_{111} = 0 \qquad \bar{y}_{111} = 0$$

$$y_{112} = 0 \qquad \bar{y}_{112} = 0$$

$$y_{113} = 0 \qquad \bar{y}_{113} = 0 \qquad \text{(D.8)}$$

$$y_{114} = 0 \qquad \bar{y}_{114} = 0$$

$$y_{122} = 0 \qquad \bar{y}_{122} = 0$$

$$y_{123} = 0 \qquad \bar{y}_{123} = 0$$

$$y_{124} = 0 \qquad \bar{y}_{124} = 0$$

$$y_{133} = 0 \qquad \bar{y}_{133} = 0$$

$$y_{134} = 0 \qquad \bar{y}_{134} = 0$$

$$y_{144} = 0 \qquad \bar{y}_{144} = 0$$

$$y_{222} = 0 \qquad \bar{y}_{222} = 0$$

$$y_{223} = 0 \qquad \bar{y}_{223} = 0$$

$$y_{224} = 0 \qquad \bar{y}_{224} = 0$$

$$y_{233} = -f\, g_{\mu\mu} \qquad \bar{y}_{233} = -\bar{f}\, \bar{g}_{33}$$

$$y_{234} = -f_{\varepsilon}\, g_{\mu} \qquad \bar{y}_{234} = -\bar{f}_4\, \bar{g}_3$$

$$y_{244} = -f_{\varepsilon\varepsilon} g \qquad \bar{y}_{244} = -\bar{f}_{44} \bar{g}$$

$$y_{333} = -f g_{\mu\mu\mu}{}^{t} \qquad \bar{y}_{333} = 0$$

$$y_{334} = -f_{\varepsilon} g_{\mu\mu}{}^{t} \qquad \bar{y}_{334} = 0$$

$$y_{344} = -f_{\varepsilon\varepsilon} g_{\mu}{}^{t} \qquad \bar{y}_{344} = 0$$

$$y_{444} = 4f_{\varepsilon}\, f_{\varepsilon\varepsilon} + 2f_{\varepsilon}\, f_{\varepsilon\varepsilon} + 2f\, f_{\varepsilon\varepsilon\varepsilon} - f_{\varepsilon\varepsilon\varepsilon} gt \qquad \bar{y}_{444} = 6\bar{f}_4 \bar{f}_{44} + 2\bar{f}\, \bar{f}_{444}\ .$$

Finally all the reference values of the partial derivatives of range r are:

$$\bar{r}_1 = \bar{r}_y\, \bar{y}_1\ ,\ \bar{r}_2 = \bar{r}_y\, \bar{y}_2\ ,\ \bar{r}_3 = 0\ ,\ \bar{r}_4 = \bar{r}_y\, \bar{y}_4$$

$$\bar{r}_{11} = \bar{r}_y\, \bar{y}_{11} + \bar{r}_{yy}\, \bar{y}_1^2$$

$$\bar{r}_{12} = \bar{r}_y\, \bar{y}_{12} + \bar{r}_{yy}\, \bar{y}_1\, \bar{y}_2$$

$$\bar{r}_{13} = 0$$

$$\bar{r}_{14} = \bar{r}_{yy}\, \bar{y}_1\, \bar{y}_4$$

$$\bar{r}_{22} = \bar{r}_y\, \bar{y}_{22} + \bar{r}_{yy}\, \bar{y}_2^2$$

$$\bar{r}_{23} = \bar{r}_y\, \bar{y}_{23}$$

$$\bar{r}_{24} = \bar{r}_y\, \bar{y}_{24} + \bar{r}_{yy}\, \bar{y}_2\, \bar{y}_4$$

$$\bar{r}_{33} = 0$$

$$\bar{r}_{34} = 0$$

$$\bar{r}_{44} = \bar{r}_y\, \bar{y}_{44} + \bar{r}_{yy}\, \bar{y}_4^2$$

$$\bar{r}_{111} = 3\bar{r}_{yy}\, \bar{y}_1\, \bar{y}_{11} + \bar{r}_{yyy}\, \bar{y}_1^3$$

$$\bar{r}_{112} = \bar{r}_{yy}(2\bar{y}_1\, \bar{y}_{12} + \bar{y}_2\, \bar{y}_{11}) + \bar{r}_{yyy}\, \bar{y}_1^2\, \bar{y}_2$$

$$\bar{r}_{113} = 0$$

$$\bar{r}_{114} = \bar{r}_{yy}\, \bar{y}_4\, \bar{y}_{11} + \bar{r}_{yyy}\, \bar{y}_1^2\, \bar{y}_4$$

$$\bar{r}_{122} = \bar{r}_{yy}(2\bar{y}_2\, \bar{y}_{12} + \bar{y}_1 \bar{y}_{22}) + \bar{r}_{yyy}\, \bar{y}_2^2\, \bar{y}_1 \qquad \text{(D.9)}$$

$$\bar{r}_{123} = \bar{r}_{yy}\,\bar{y}_1\,\bar{y}_{23}$$

$$\bar{r}_{124} = \bar{r}_{yy}(\bar{y}_1\,\bar{y}_{24} + \bar{y}_4\,\bar{y}_{12}) + \bar{r}_{yyy}\,\bar{y}_1\,\bar{y}_2\,\bar{y}_4$$

$$\bar{r}_{133} = 0$$

$$\bar{r}_{134} = 0$$

$$\bar{r}_{144} = \bar{r}_{yy}\,\bar{y}_1\,\bar{y}_{44} + \bar{r}_{yyy}\,\bar{y}_4^2\,\bar{y}_1$$

$$\bar{r}_{222} = 3\bar{r}_{yy}\,\bar{y}_2\,\bar{y}_{22} + \bar{r}_{yyy}\,\bar{y}_2^3$$

$$\bar{r}_{223} = 2\bar{r}_{yy}\,\bar{y}_2\,\bar{y}_{23}$$

$$\bar{r}_{224} = \bar{r}_{yy}(2\bar{y}_2\,\bar{y}_{24} + \bar{y}_4\,\bar{y}_{22}) + \bar{r}_{yyy}\,\bar{y}_2^2\,\bar{y}_4$$

$$\bar{r}_{233} = \bar{r}_y\,\bar{y}_{233}$$

$$\bar{r}_{234} = \bar{r}_y\,\bar{y}_{234} + \bar{r}_{yy}\,\bar{y}_4\,\bar{y}_{23}$$

$$\bar{r}_{244} = \bar{r}_y\,\bar{y}_{244} + \bar{r}_{yy}(2\bar{y}_4\,\bar{y}_{24} + \bar{y}_2\,\bar{y}_{44}) + \bar{r}_{yyy}\,\bar{y}_4^2\,\bar{y}_2$$

$$\bar{r}_{333} = 0$$

$$\bar{r}_{334} = 0$$

$$\bar{r}_{344} = 0$$

$$\bar{r}_{444} = \bar{r}_y\,\bar{y}_{444} + 3\bar{r}_{yy}\,\bar{y}_4\,\bar{y}_{44} + \bar{r}_{yyy}\,\bar{y}_4^3$$

The Taylor expansion for range r is given by

$$r(1,2,3,4) = \bar{r} + (\bar{r}_1\hbar + \bar{r}_2 t + \bar{r}_4\varepsilon) + \{\bar{r}_{11}\frac{\hbar^2}{2} + \bar{r}_{12}\hbar t + \bar{r}_{14}\hbar\varepsilon + \bar{r}_{22}\frac{t^2}{2} + \bar{r}_{23}t\mu$$

$$+ \bar{r}_{24}t\varepsilon + \bar{r}_{44}\frac{\varepsilon^2}{2}\} + \{\bar{r}_{111}\frac{\hbar^3}{6} + \bar{r}_{112}\frac{\hbar^2 t}{2} + \bar{r}_{114}\frac{\hbar^2\varepsilon}{2} + \bar{r}_{122}\frac{\hbar t^2}{2} + \bar{r}_{123}\hbar t\mu + \bar{r}_{124}\hbar t\varepsilon$$

$$+ \bar{r}_{144}\frac{\hbar\varepsilon^2}{2} + \bar{r}_{222}\frac{t^3}{6} + \bar{r}_{223}\frac{t^2\mu}{2} + \bar{r}_{224}\frac{t^2\varepsilon}{2} + \bar{r}_{233}\frac{t\mu^2}{2} + \bar{r}_{234}t\mu\varepsilon + \bar{r}_{244}\frac{t\varepsilon^2}{2}$$

$$+ \bar{r}_{444}\frac{\varepsilon^3}{6}\} \quad . \qquad \text{(D.10)}$$

For the round trip delay, multiply (D.10) by 2/c. This is the same as replacing $\bar{r}$ by $\bar{\tau}$, $\bar{r}_i$ by $\bar{\tau}_i$, $\bar{r}_{ij}$ by $\bar{\tau}_{ij}$ and $\bar{r}_{ijk}$ by $\bar{\tau}_{ijk}$, where i, j, k take the values {1,2,3,4}. The Taylor expansion for the round trip delay is in (4.9)

APPENDIX E

Listings for FRRDP and Associated Subprograms

This Appendix contains the listings for FRRDP and for the associated subprograms in alphabetical order. Excluded are: DISH and BESSJ1 which are in Appendix C.

PROGRAM FRRDP.FOR

```
C ASSUME NBSCA IN GROUNDMAP AND A FOOTPRINT AT WQ9 ,1/16/89, 1700 hr
$LARGE
      CHARACTER*240 TITLE
      REAL W(27),TIMCNT,TIMDWL,VELOM,ALFM,BETM,WMPOSC(3),WQ9(5),DELRAN,
     &FFNBIT,WPAR(6),XIFXMT,SELWIN,BETKAI,SELDIS,BITINT
      REAL SMOMAP(450),TXD(64),TXQ(64),
     &WADRLO(165),WADRUP(165),WADSMO(90),WAGAIN(3),WANGN9(165),
     &WAGNCL(128),WBET9(165),WDUM(3),WDOVL(3),WEPSMX(165),WIND(64)
      REAL WMPOSB(3),WMPOSE(3),WMU3DB(165),WMUMDR(166),WMVELO(3),
     &WRAN9(165),WTAU(25),WTIME(64),WTSMO(5),WWAPEX(128,9),
     &XD(128),XM(128),XQ(128)
      INTEGER LNPPP(128)
      REAL DISH,DOTCOS,FRQOFM,FREQ
      REAL CLIGHT,DELFIN,DELFUT,DELTIM,DOVL,EPS,FMU,FREQO,GDISH2,
     &GDISH3,GRC,HDELRA,HTHETA,HTIMDW,RANCOF,RANMN,RANMX,OMEGO,SCAFAC,
     &SINANG,TEMP,THETA,TMPBET,TPI,V1,V2,V3,X,XSCA,YSCA,Y,Z,FZMX
      INTEGER IDDISH,IFILE,INDX,INTCOF,IW,JRAN,JRANQ9,JRANMX,JW,K,
     &KADSMO,KDQ,KI,KJ,KFF,KNPPP,KSCALO,KSCAUP,KT,KTIM,KTHSCA,
     &NBITFF,NBITFX,NFFT,NFFTH,NFFX,NFFXH,NPPPMN,NPPPMX,NSCAFP,NXMAP,
     &NYMAP
      REAL SYNMAP(128,55)
      EQUIVALENCE (W(1),TIMCNT),(W(2),TIMDWL),(W(3),VELOM),(W(4),ALFM),
     &(W(5),BETM),(W(6),WMPOSC(1)),(W(9),WQ9(1)),(W(15),DELRAN),(W(16),
     &FFNBIT),(W(17),WPAR(1)),(W(23),XIFXMT),(W(24),SELWIN),(W(25),BETKA
     &I),(W(26),SELDIS),(W(27),BITINT)
C     PARAMETERS FOR OUTPUT MAP ARE IN WPAR=(DELMAP,THRESH,XMNMAP,YMNMAP
C     ,XMXMAP,YMXMAP) ************************************************
C     WQ9=(XQ,XQ,ZQ,ALFQ9,BETQ9,RANQ9) ********************************
C     DIMENSION OF SMOMAP = NBSCA*15 (SEE SUB. CLUTINFO) ***************
C     INPUT FILES: CRRDPI.INP AND CLUTINFI.INP *************************
C     OUTPUT FILES: FRRDPG.GRD AND FRRDPP.PRT **************************
      FRQOFMU(V1,FMU,V2,V3)=-V1*(V2*FMU+(V3/2.)*FMU**2)
      OPEN(UNIT=11,FILE='CLUTINFI.INP',STATUS='OLD')
      OPEN(UNIT=9,FILE='CRRDPI.INP',STATUS='OLD')
      OPEN(UNIT=3,FILE='FRRDPP.PRT',STATUS='OLD')
      OPEN(UNIT=7,FILE='FRRDPG.GRD',STATUS='OLD')
```

```
C    HALF 3db BEAMWIDTH=.0215 gives DOVL=14.83343168=DISH DIAMETER IN
C    &LAMBDA UNITS ( 2-WAY GAIN = -3.00 DB means DISH=0.84139514)*******
     DATA THETA,FREQO,CLIGHT,TPI,EPS,IFILE,WDOVL/.043,94.E9,3.E8,
    &6.28318531,.0001,3,17.15094,17.15094,14.80884/
     SINANG=SIN(0.0215)
     GDISH2=DISH(SINANG,2,WDOVL(2))
     GDISH3=DISH(SINANG,3,WDOVL(3))
     WRITE(3,*)'GDISH2,GDISH3',GDISH2,GDISH3
     TITLE='TIMCNT=1; TIMDWL=2; VELOM=3; ALFM=4; BETM=5; WMPOSC=6-8; WQ
    &9=9-10          DUMMY=11-14;DELRAN=15;FFNBIT=16;WPAR1=17=DELM
    &AP;WPAR2=18=THRESH;WPAR3=19=XMN ;WPAR4=20=YMN/ WPAR5=21=XMX; WPAR6
    &=22=YMX; XIFXMT=23,SELWIN=24; BETKAI=25;SELDIS=26; BITINT=27'
     CALL WINPUT(27,W,9,TITLE)
     CALL PRTXY(W,1,27,28,IFILE)
C    READJUST (XMNMAP,YMNMAP) TO BE INTEGER MULTIPLE OF DELMAP ********
     DO 3 K=3,4
 3   WPAR(K)=WPAR(1)*FLOAT(INT(WPAR(K)/WPAR(1)))
     IDDISH=INT(SELDIS+.00001)
     DOVL=WDOVL(IDDISH)
     NXMAP=1.+(WPAR(5)-WPAR(3))/WPAR(1)
     NYMAP=1.+(WPAR(6)-WPAR(4))/WPAR(1)
     WRITE(*,*)'NXMAP=',NXMAP,'  NYMAP=',NYMAP
     OMEGO=TPI*FREQO
     HDELRA=DELRAN/2.
     HTHETA=THETA/2.
     NBITFF=INT(FFNBIT+EPS)
     NBITFX=NBITFF+INT(BITINT+EPS)
     NFFT=2**NBITFF
     NFFTH=NFFT/2
     NFFX=2**NBITFX
     NFFXH=NFFX/2
     DELFIN=1./TIMDWL
     INTCOF=2**INT(BITINT+EPS)
     DELFUT=DELFIN/FLOAT(INTCOF)
     DELTIM=TIMDWL/NFFT
     DO 2 KTIM=1,NFFT
 2   WTIME(KTIM)=DELTIM*(KTIM-NFFTH-0.5)
C    COMPUTE MISSILE INFORMATION ***************************************
     WMVELO(1)=VELOM*SIN(BETM)*SIN(ALFM)
     WMVELO(2)=VELOM*SIN(BETM)*COS(ALFM)
     WMVELO(3)=-VELOM*COS(BETM)
     HTIMDW=TIMDWL*.5
     DO 21 K=1,3
     WMPOSB(K)=WMPOSC(K)+WTIME(1)*WMVELO(K)
 21  WMPOSE(K)=WMPOSC(K)+WTIME(NFFT)*WMVELO(K)
C    COMPUTE FOOTPRINT INFORMATION AT WQ9 ******************************
     GRC=SQRT((WQ9(1)-WMPOSC(1))**2+(WQ9(2)-WMPOSC(2))**2)
     WQ9(4)=ATAN2((WQ9(1)-WMPOSC(1)),(WQ9(2)-WMPOSC(2)))
     WQ9(5)=ATAN2(GRC,WMPOSC(3))
     WQ9(6)=SQRT(GRC**2+WMPOSC(3)**2)
     RANMN=WMPOSC(3)/COS(WQ9(5)-HTHETA)
     RANMX=WMPOSC(3)/COS(WQ9(5)+HTHETA)
     JRANQ9=INT((WQ9(6)-RANMN+HDELRA)/DELRAN)
```

```
      JRANMX=2*JRANQ9
C     STORE RANGE,BET,EPSMX,AND MU-3DB FOR FICTITIOUS SCATTERERS *******
      DO 7 JRAN=1,JRANMX
      WRAN9(JRAN)=WQ9(6)+(JRAN-JRANQ9)*DELRAN
      WBET9(JRAN)=ACOS(WMPOSC(3)/WRAN9(JRAN))
      TMPBET=ACOS(WMPOSC(3)/(WRAN9(JRAN)-HDELRA))
      WEPSMX(JRAN)=WBET9(JRAN)-TMPBET
    7 WMU3DB(JRAN)=ACOS((COS(HTHETA)-COS(WQ9(5))*COS(WBET9(JRAN)))/
     &(SIN(WQ9(5))*SIN(WBET9(JRAN)))-0.00000016)
C     INITIALIZE ARRAY SYNMAP ******************************************
      DO 13 JRAN=1,JRANMX
      DO 13 KNPPP=1,NFFX
   13 SYNMAP(KNPPP,JRAN)=0.
C     THIS IN PREPARATION FOR MAPPING **********************************
C     ENTER CLUTTER INFORMATION FOR FOOTPRINT AT WQ9 AND MAP PARAMETERS
      WRITE(3,*)'JRAMNX=',JRANMX
      CALL CLUTINFO(DELRAN,HTHETA,JRANQ9,JRANMX,WQ9,WMPOSC,WPAR,NSCAFP,
     &SMOMAP,WADRLO,WADRUP,WADSMO)
C     INDEX KNPPP=1,NFFX FOR FREQ=[-NFFXH*DELFUT,(NFFXH-1)*DELFUT] *****
C     COMPUTE ARRAY LNPPP,THEN KFF=LNPPP(KNPPP) ************************
      DO 9 KNPPP=1,NFFX
      INDX=KNPPP+NFFXH
      IF(INDX.GT.NFFX) INDX=INDX-NFFX
    9 LNPPP(KNPPP)=INDX
C     PROCESS RANGE CELLS SEQUENTIALLY AND COMPUTE REFLECTIVITY MAP ****
      DO 35 JRAN=1,JRANMX
      KSCALO=WADRLO(JRAN)+EPS
      IF(KSCALO.EQ.0) GO TO 35
C     INITIALIZING QUADRATURE ARRAYS=(XD,XQ) ***************************
      DO 11 K=1,NFFX
      XD(K)=0.
   11 XQ(K)=0.
      WDUM(3)=0.
C     COMPUTE TAYLOR COEFFICIENTS WTAU=(TAU,TAUt,TAUe,TAUtt,TAUee,TAUtm,
C     TAUte,TAUttt,TAUeee,TAUtmm,TAUtee,TAUtme,TAUttm,TAUtte) **********
      CALL TAYLOR(VELOM,BETM,ALFM,WQ9(4),WQ9(5),WMPOSC(3),HTHETA,
     &DELRAN,JRAN,WTAU)
      WRITE(3,*)'ARRAY WTAU'
      CALL PRTXY(WTAU,1,25,119,3)
C     PREPARATION FOR MAPPING: COMPUTE (NPPPMN,NPPPMX) ON 3DB FOOTPRINT
C     TO LIMIT # CELLS BEING MAPPED TO KNPPP=NPPPMN,NPPPMX *************
      NPPPMN=FRQOFMU(FREQO,WMU3DB(JRAN),WTAU(6),WTAU(10))/DELFUT
     &+NFFXH-5.
      NPPPMX=FRQOFMU(FREQO,-WMU3DB(JRAN),WTAU(6),WTAU(10))/DELFUT
     &+NFFXH+5.
C     COMPUTE MU AT CENTER OF CELLS FOR INCREASING DOPPLER *************
      DO 17 K=NPPPMN-1,NPPPMX+1
      FREQ=DELFUT*FLOAT(K-NFFXH-1)/FREQO
   17 WMUMDR(K)=(-WTAU(6)+SQRT(WTAU(6)**2-2.*WTAU(10)*FREQ))/WTAU(10)
C     COEFFICIENT FOR RANGE NORMALIZATION: RANCOF **********************
      RANCOF=(WRAN9(JRAN)/WRAN9(JRANQ9))**4
      KSCAUP=WADRUP(JRAN)+EPS
      DO 30 KTHSCA=KSCALO,KSCAUP
```

```
      KADSMO=WADSMO(KTHSCA)+EPS
C     INFORMATION ON SCATTERER,WTSMO=(AMPSCA,FASSCA,MUSCA,JRAN-SCA,ZSCA)
      DO 31 KT=1,5
 31   WTSMO(KT)=SMOMAP(KADSMO-4+KT)
C     ANTGAIN(t)=WAGAIN(1)+WAGAIN(2)*t+WAGAIN(3)*t^2 ********************
C     WAGAIN IS OUTPUT OF SCAGAN,INPUTS ARE DOVL,HTIMDW,WMPOS=(BEG,
C     &CENT,END),WQ9,WTSMO=(AMP,FAS,MU,JRAN,ZSCA),JRANQ9,DELRAN *********
      CALL SCAGAN(IDDISH,DOVL,HTIMDW,WMPOSB,WMPOSC,WMPOSE,WQ9,WTSMO,
     &JRANQ9,DELRAN,WAGAIN)
      CALL TSIGNAL(OMEGO,NFFT,RANCOF,XIFXMT,WTSMO,WTAU,WTIME,WAGAIN,TXD,
     &TXQ)
      DO 33 KDQ=1,NFFT
      XD(KDQ)=XD(KDQ)+TXD(KDQ)
 33   XQ(KDQ)=XQ(KDQ)+TXQ(KDQ)
 30   CONTINUE
      CALL DERAMP(OMEGO,WTAU(4),NFFT,XD,XQ,WTIME)
      CALL WINDOWS(NFFT,4,BETKAI,WIND)
      DO 37 K=1,NFFT
      XD(K)=XD(K)*WIND(K)
 37   XQ(K)=XQ(K)*WIND(K)
      CALL FFT(-NBITFX,1,1,1,XD,XQ)
C     COMPUTE ARRAY OF ANTENNA GAIN AT CENTER OF CELLS ******************
      DO 39 KNPPP=NPPPMN,NPPPMX
      SINANG=SQRT(1.-(SIN(WQ9(5))*SIN(WBET9(JRAN))*COS(WMUMDR(KNPPP))+
     &COS(WQ9(5))*COS(WBET9(JRAN)))**2)
 39   WAGNCL(KNPPP)=DISH(SINANG,IDDISH,DOVL)**2
C     (XD,XQ) NEED TO BE SCALED FOR RANGE AND ANTENNA GAIN AT KNPPP ****
      DO 41 KNPPP=NPPPMN,NPPPMX
      KFF=LNPPP(KNPPP)
      SCAFAC=1./(RANCOF*WAGNCL(KNPPP))
      XD(KFF)=XD(KFF)*SCAFAC
      XQ(KFF)=XQ(KFF)*SCAFAC
      XM(KFF)=SQRT(XD(KFF)**2+XQ(KFF)**2)
C     SYNMAP STORES THE FOOTPRINT MAP, X<==>JRAN; Y<==>DOPPLER *********
      IF(XM(KFF).GT.1.0) SYNMAP(KNPPP,JRAN)=XM(KFF)
      WRITE(3,*)'JRAN=',FLOAT(JRAN),'  KNPPP=',FLOAT(KNPPP),'  SYNMAP=',
     &SYNMAP(KNPPP,JRAN)
 41   CONTINUE
 35   CONTINUE
      FZMX=0.
      DO 60 IW=1,NFFX
      DO 60 JW=1,JRANMX
      IF(SYNMAP(IW,JW).GT.FZMX) FZMX=SYNMAP(IW,JW)
 60   CONTINUE
      WRITE(7,'("DSAA")')
      WRITE(7,'(I5,1X,I5)')NFFX,JRANMX
      WRITE(7,'(E12.5,1X,E12.5)')1.,FLOAT(NFFX)
      WRITE(7,'(E12.5,1X,E12.5)')1.,FLOAT(JRANMX)
      WRITE(7,'(E12.5,1X,E12.5)')0.,FZMX
      DO 69 JW=1,JRANMX
      WRITE(7,62) (SYNMAP(IW,JW),IW=1,NFFX)
      WRITE(7,'( )')
 69   CONTINUE
```

```
62  FORMAT(10(1XF8.3))
    CLOSE(3)
    CLOSE(9)
    CLOSE(7)
    CLOSE(11)
    STOP
    END
```

SUBROUTINE CLUTINFO.FOR

```
    SUBROUTINE CLUTINFO(DELRAN,HTHETA,JRANQ9,JRANMX,WQ9,WMPOSC,WPAR,
   &NSCAFP,SMOMAP,WADRLO,WADRUP,WADSMO)
    CHARACTER*240 TITLE
    REAL WMPOSC(1),WSCA(7),WQ9(1),WPAR(1),WMSCA(3),SMOMAP(1),
   &GRDMAP(150),WADRLO(1),WADRUP(1),WJRAN(200),WADSMO(1),W(34)
    REAL W1,COSANT,XSCA,YSCA,HDELRA,DOTCOS,EPS,RGI,XI,YI,FMUI
    INTEGER NW,KK,KW,NBSCA,KSMO,KSCA,KX,JRANON,KSCAT,KINC,K,J
C    WQ9=(XQ,YQ,ZQ,ALFQ9,BETQ9,RANQ9) **********************************
C    DIMENSION OF GRDMAP IS NBSCA*4 ************************************
C    FIRST CALL ENTER THE GROUND MAP INTO GRDMAP ***********************.
C    INPUTS FOR GRDMAP:NBSCA,SCAAMP,SCAFAS,SCAX,SCAY ******************
    TITLE='
   &                ENTERING SCAT-DATA, W(1)=3*NUMBER OF SCAT+1,
   &{W(3i-1),W(3i),W(3i+1)}={Xi,Yi,Hbari} or {MUi,RGi,Hbari} for ith
   &SCATTERER'
    READ(11,2) W1
2   FORMAT(F10.5)
    NW=INT(ABS(W1)+.0001)
    REWIND 11
    CALL WINPUT(NW,W,11,TITLE)
    COSANT=COS(HTHETA*1.13)
    KK=0
    DO 3 KW=2,NW,3
    GRDMAP(KK+1)=1.
    GRDMAP(KK+2)=0.
    XI=W(KW)
    YI=W(KW+1)
    FMUI=ATAN2((XI-WMPOSC(1)),(YI-WMPOSC(2)))-WQ9(4)
    RGI=SQRT((XI-WMPOSC(1))**2+(YI-WMPOSC(2))**2+WMPOSC(3)**2)
    IF(W1.GT.0.) GO TO 1
    FMUI=W(KW)
    RGI=W(KW+1)
    XI=SQRT(RGI**2-WMPOSC(3)**2)*SIN(FMUI+WQ9(4))+WMPOSC(1)
    YI=SQRT(RGI**2-WMPOSC(3)**2)*COS(FMUI+WQ9(4))+WMPOSC(2)
    W(KW)=XI
    W(KW+1)=YI
1   WRITE(*,99)XI, YI, FMUI, RGI, W(KW+2)
99  FORMAT(1X,'XI=',E10.5,1X,'YI=',E10.5,1X,'MUI=',E12.5,1X,'RGI=',
   &E10.5,1X,'HI=',E10.5)
    GRDMAP(KK+3)=W(KW)
```

```
      GRDMAP(KK+4)=W(KW+1)
      GRDMAP(KK+5)=W(KW+2)
    3 KK=KK+5
      NBSCA=(NW-1)/3
C     SMOOTHED OUTPUT FOR CURRENT FOOTPRINT IS SMOMAP, ONE REAL
C     &SCATTERER GIVES 15-COMPONENTS ************************************
C     SMOMAP=(EARLY,SCAFAS,MU,JRANON-1,ZSCA,ON,SCAFAS,MU,JRANON,ZSCA,LATE
C     &,SCAFAS,MU,JRANON+1,ZSCA) ****************************************
      KSMO=0
      HDELRA=DELRAN/2.
C     WSCA=(X-SCA,Y-SCA,Z-SCA,AMP-SCA,FAS-SCA) **************************
      DO 5 KSCA=1,NBSCA
      KX=(KSCA-1)*5+3
      WSCA(1)=GRDMAP(KX)
      WSCA(2)=GRDMAP(KX+1)
      WSCA(3)=GRDMAP(KX+2)
      WSCA(4)=GRDMAP(KX-2)
      WSCA(5)=GRDMAP(KX-1)
      IF(DOTCOS(WMPOSC,WQ9,WSCA).LT.COSANT) GO TO 5
      IF(WSCA(1).LT.(WPAR(3)-WPAR(1)).OR.WSCA(1).GT.(WPAR(5)+WPAR(1)))
     &GO TO 5
      IF(WSCA(2).LT.(WPAR(4)-WPAR(1)).OR.WSCA(2).GT.(WPAR(6)+WPAR(1)))
     &GO TO 5
C     REPLACE (X,Y) BY (MU,RANSCA) **************************************
      WSCA(6)=WSCA(1)
      WSCA(7)=WSCA(2)
      WSCA(1)=ATAN2(WSCA(6)-WMPOSC(1),WSCA(7)-WMPOSC(2))-WQ9(4)
      WSCA(2)=SQRT((WSCA(6)-WMPOSC(1))**2+(WSCA(7)-WMPOSC(2))**2+
     &(WSCA(3)-WMPOSC(3))**2)
C     RANGE SMOOTHING:1-SCATTERER GIVES WMSCA=(EARLY,ON,LATE) **********
      JRANON=((WSCA(2)-WQ9(6)+HDELRA)/DELRAN)+JRANQ9
      WRITE(3,*)'JRANON=',JRANON,'  JRANQ9=',JRANQ9
      CALL RANGSMOO(DELRAN,JRANON,JRANQ9,WQ9(6),WMPOSC,WSCA,WMSCA)
      KSCAT=0
      DO 7 KINC=2,12,5
      SMOMAP(KSMO+KINC)=WSCA(5)
      SMOMAP(KSMO+KINC+1)=WSCA(1)
      KSCAT=KSCAT+1
      SMOMAP(KSMO+KINC-1)=WMSCA(KSCAT)
      SMOMAP(KSMO+KINC+2)=JRANON-2+KSCAT
    7 SMOMAP(KSMO+KINC+3)=WSCA(3)
      KSMO=KSMO+15
    5 CONTINUE
      EPS=0.00001
      NSCAFP=0
      DO 10 K=4,KSMO,5
      NSCAFP=NSCAFP+1
      WADSMO(NSCAFP)=K
   10 WJRAN(NSCAFP)=SMOMAP(K)
      CALL SORTRI(WJRAN,WADSMO,-NSCAFP)
      DO 12 J=1,JRANMX
      WADRLO(J)=0.
      WADRUP(J)=0.
```

```
      DO 14 K=NSCAFP,1,-1
   14  IF(ABS(WJRAN(K)-FLOAT(J)).LE.EPS) WADRLO(J)=FLOAT(K)
      DO 16 K=1,NSCAFP
   16  IF(ABS(WJRAN(K)-FLOAT(J)).LE.EPS) WADRUP(J)=FLOAT(K)
   12  CONTINUE
      RETURN
      END
```

SUBROUTINE DERAMP.FOR

```
      SUBROUTINE DERAMP(OMEGO,TAU4,NFFT,XD,XQ,WTIME)
      REAL WTIME(1),XD(1),XQ(1)
      DOUBLE PRECISION HOMTTT,TIM,ANGROT,TXD,TXQ
      HOMTTT=0.5*OMEGO*TAU4
      DO 5 K=1,NFFT
      TIM=WTIME(K)
      ANGROT=HOMTTT*TIM**2
      TXD=XD(K)
      TXQ=XQ(K)
      XD(K)=TXD*DCOS(ANGROT)-TXQ*DSIN(ANGROT)
      XQ(K)=TXD*DSIN(ANGROT)+TXQ*DCOS(ANGROT)
    5 CONTINUE
      RETURN
      END
```

FUNCTION DOTCOS.FOR

```
      FUNCTION DOTCOS(W1,W2,W3)
C     DOTCOS=COS{ANG(W2-W1,W3-1)} ***************************************
      REAL W1(1),W2(1),W3(1),V1(3),V2(3),V1M,V2M
      V1(1)=W2(1)-W1(1)
      V1(2)=W2(2)-W1(2)
      V1(3)=W2(3)-W1(3)
      V1M=SQRT(V1(1)**2+V1(2)**2+V1(3)**2)
      V2(1)=W3(1)-W1(1)
      V2(2)=W3(2)-W1(2)
      V2(3)=W3(3)-W1(3)
      V2M=SQRT(V2(1)**2+V2(2)**2+V2(3)**2)
      DOTCOS=(V1(1)*V2(1)+V1(2)*V2(2)+V1(3)*V2(3))/(V1M*V2M)
      IF(ABS(DOTCOS).GE.1.) DOTCOS=SIGN(0.99999999,DOTCOS)
      RETURN
      END
```

SUBROUTINE FFT.FOR

```
      SUBROUTINE FFT(L,KRZIN,KRZOUT,KUNSCR,XR,XI)
      REAL XR(1),XI(1),TPI,EPSFI,EPSMO,HOLDR,ANG,TR,TI,DELANG,HOLDI
      INTEGER M,N,J,I,K,J1,JG,JUMP,JUMPPO,JUMPMO,J1MAX,JUMPEX,KMAX,IMAX,
     &NDISTM,KANDS,NDIST,NSPAN,NHALF,NHALFP
      TPI=6.28318530717958
      M=IABS(L)
      EPSFI=1.E-30
      EPSMO=1.E-15
      IF(M.EQ.0) RETURN
      N=2**M
      GO TO (9,10),KRZIN
10    DO 25 J=1,N
      HOLDR=XR(J)
      XR(J)=HOLDR*COS(XI(J))
25    XI(J)=HOLDR*SIN(XI(J))
9     NHALF=N/2
      NHALFP=NHALF+1
      DELANG=TPI/FLOAT(N)
      IF(L.LT.0) DELANG=-DELANG
      NSPAN=N
      NDIST=NSPAN/2
      DO 2 I=1,M
      ANG=0.0
      DO 3 J=1,NDIST
      TR=COS(ANG)
      TI=SIN(ANG)
      DO 5 K=J,N,NSPAN
      KANDS=K+NDIST
      HOLDR=XR(K)-XR(KANDS)
      HOLDI=XI(K)-XI(KANDS)
      XR(K)=XR(K)+XR(KANDS)
      XI(K)=XI(K)+XI(KANDS)
      XR(KANDS)=TR*HOLDR-TI*HOLDI
      XI(KANDS)=TI*HOLDR+TR*HOLDI
5     CONTINUE
3     ANG=ANG+DELANG
      NSPAN=NDIST
      NDIST=NSPAN/2
2     DELANG=DELANG+DELANG
      IF(L.LT.0) GO TO 23
      GO TO (18,19),KRZOUT
18    DO 1 J=1,N
      XR(J)=XR(J)/FLOAT(N)
1     XI(J)=XI(J)/FLOAT(N)
      GO TO 21
19    DO 20 J=1,N
      HOLDR=XR(J)/FLOAT(N)
      HOLDI=XI(J)/FLOAT(N)
      XR(J)=SQRT(HOLDR**2+HOLDI**2)
      IF(XR(J).GT.EPSMO) GO TO 26
      XI(J)=0.0
      GO TO 20
```

```
26  IF(ABS(HOLDR).LT.EPSFI) HOLDR=SIGN(EPSFI,HOLDR)
    XI(J)=ATAN2(HOLDI,HOLDR)
20  CONTINUE
21  CONTINUE
    GO TO 4
23  GO TO (4,11),KRZOUT
11  DO 22 J=1,N
    HOLDR=XR(J)
    XR(J)=SQRT(HOLDR**2+XI(J)**2)
    IF(XR(J).GT.EPSMO) GO TO 27
    XI(J)=0.0
    GO TO 22
27  IF(ABS(HOLDR).LT.EPSFI) HOLDR=SIGN(EPSFI,HOLDR)
    XI(J)=ATAN2(XI(J),HOLDR)
22  CONTINUE
4   IF(M.EQ.1) RETURN
    IF(KUNSCR.EQ.0) RETURN
    JUMPEX=1
    NDIST=N
    IMAX=M/2
    DO 6 I=1,IMAX
    JUMP=JUMPEX
    JUMPPO=JUMP+1
    JUMPMO=JUMP-1
    JUMPEX=JUMPEX+JUMPEX
    NSPAN=NDIST
    NDIST=NDIST/2
    NDISTM=NDIST-JUMP
    DO 8 JG=JUMPPO,N,NSPAN
    J1MAX=JG+NDISTM
    DO 8 J1=JG,J1MAX,JUMPEX
    KMAX=J1+JUMPMO
    DO 8 K=J1,KMAX
    KANDS=K+NDISTM
    HOLDR=XR(K)
    HOLDI=XI(K)
    XR(K)=XR(KANDS)
    XI(K)=XI(KANDS)
    XR(KANDS)=HOLDR
8   XI(KANDS)=HOLDI
6   CONTINUE
    RETURN
    END
```

SUBROUTINE KINPUT.FOR

```
    SUBROUTINE KINPUT(K)
    INTEGER K
    WRITE(*,'(1X,"K=?",\)')
    READ(*,'(I10)')K
```

```
      RETURN
      END
```

SUBROUTINE PAYOFX.FOR

```
      SUBROUTINE PAYOFX(X1,X2,X3,Y1,Y2,Y3,A1,A2,A3)
      REAL X12M,X12P,X13M,X13P
      X12M=X1-X2
      X12P=X1+X2
      X13M=X1-X3
      X13P=X1+X3
      A3=((Y1-Y2)*X13M-(Y1-Y3)*X12M)/(X12M*X12P*X13M-X13M*X13P*X12M)
      A2=((Y1-Y2)-A3*X12M*X12P)/X12M
      A1=Y1-A2*X1-A3*X1*X1
      RETURN
      END
```

SUBROUTINE PRTXY.FOR

```
      SUBROUTINE PRTXY(XY,KSTART,KEND,ISEQ,IFILE)
      REAL XY(1)
      INTEGER L
    1 FORMAT(6(1X,E12.6))
    2 FORMAT(1X,13HPRTXY KSTART=,I8,18X,5HKEND=,I8,8X,5HISEQ=,I4)
    3 FORMAT(1X,35H************************************)
      WRITE(IFILE,3)
      WRITE(IFILE,2)KSTART,KEND,ISEQ
      WRITE(IFILE,1)(XY(L),L=KSTART,KEND)
      RETURN
      END
```

SUBROUTINE RANGSMOO.FOR

```
      SUBROUTINE RANGSMOO(DELRAN,JRANON,JRANQ9,RANGE9,WMPOSC,WSCA,WMSCA)
      REAL WMPOSC(1),WSCA(1),WMSCA(1),F,X,RANON,RANER,RANLT,GRANON,
     &GRANSC,DELON,DELER,DELLT,SUM
      INTEGER K
      F(X)=1.-ABS(X)/1.5
      RANON=RANGE9+(JRANON-JRANQ9)*DELRAN
      RANER=RANON-DELRAN
      RANLT=RANON+DELRAN
      GRANON=SQRT(RANON**2-WMPOSC(3)**2)
      GRANSC=SQRT(WSCA(2)**2-WMPOSC(3)**2)
      DELON=(GRANON-GRANSC)/DELRAN
```

```
      DELER=(DELON-DELRAN)/DELRAN
      DELLT=(DELON+DELRAN)/DELRAN
      WMSCA(1)=F(DELER)
      WMSCA(2)=F(DELON)
      WMSCA(3)=F(DELLT)
      SUM=WMSCA(1)+WMSCA(2)+WMSCA(3)
      DO 5 K=1,3
    5  WMSCA(K)=WSCA(4)*WMSCA(K)/SUM
      RETURN
      END
```

SUBROUTINE SCAGAN.FOR

```
      SUBROUTINE SCAGAN(IDDISH,DOVL,HTIMDW,WMPOSB,WMPOSC,WMPOSE,WQ9,
     &WTSMO,JRANQ9,DELRAN,WAGAIN)
C     ANTGAIN(t)=WAGAIN(1)+WAGAIN(2)*t+WAGAIN(3)*t^2 ********************
C     WAGAIN IS OUTPUT OF SCANTGAN,INPUTS ARE IDDISH,DOVL,HTIMDW,WMPOS
C     &=(BEG,CENT,END),WQ9,WTSMO=(AMP,FAS,MU,JRAN,ZSCA),JRANQ9,DELRAN ***
C     WGAIN=SCAT-ANTENNA GAIN AT (BEG,CENT,END) OF DWELL ***************
      REAL DOVL,HTIMDW,WMPOSB(1),WMPOSC(1),WMPOSE(1),WQ9(1),WTSMO(1),
     &DELRAN,WAGAIN(1),RANSCA,GRANSC,WSCA(3),WGAIN(3),SINANG,DOTCOS,DISH
      INTEGER IDDISH,JRANQ9
      RANSCA=WQ9(6)+(WTSMO(4)-FLOAT(JRANQ9))*DELRAN
      GRANSC=SQRT(RANSCA**2-(WMPOSC(3)-WTSMO(5))**2)
      WSCA(1)=WMPOSC(1)+GRANSC*SIN(WQ9(4)+WTSMO(3))
      WSCA(2)=WMPOSC(2)+GRANSC*COS(WQ9(4)+WTSMO(3))
      WSCA(3)=WTSMO(5)
      SINANG=SQRT(1.-DOTCOS(WMPOSB,WQ9,WSCA)**2)
      WGAIN(1)=DISH(SINANG,IDDISH,DOVL)**2
      SINANG=SQRT(1.-DOTCOS(WMPOSC,WQ9,WSCA)**2)
      WGAIN(2)=DISH(SINANG,IDDISH,DOVL)**2
      SINANG=SQRT(1.-DOTCOS(WMPOSE,WQ9,WSCA)**2)
      WGAIN(3)=DISH(SINANG,IDDISH,DOVL)**2
      CALL PAYOFX(-HTIMDW,0.,HTIMDW,WGAIN(1),WGAIN(2),WGAIN(3),WAGAIN(1)
     &,WAGAIN(2),WAGAIN(3))
      RETURN
      END
```

SUBROUTINE SORTRI.FOR

```
      SUBROUTINE SORTRI (VAL,LOC,NS)
      REAL VAL(1),F,XS
      INTEGER LOC(1),N,K,M,J,I,II
      N=IABS(NS)
      IF(NS.LT.0) GO TO 7
      DO 6 K=1,N
    6  LOC(K)=K
```

```
7    M=N
1    CONTINUE
     M=M/2
     IF(M.EQ.0) RETURN
     K=N-M
     J=1
2    CONTINUE
     I=J
3    CONTINUE
     II=I+M
     IF(VAL(I).LT.VAL(II)) GO TO 4
     F=VAL(I)
     XS=LOC(I)
     VAL(I)=VAL(II)
     LOC(I)=LOC(II)
     VAL(II)=F
     LOC(II)=XS
     I=I-M
     IF(I.GE.1) GO TO 3
4    CONTINUE
     J=J+1
     IF(J.GT.K) GO TO 1
     GO TO 2
     RETURN
     END
```

SUBROUTINE TAYLOR.FOR

```
     SUBROUTINE TAYLOR(VELOM,BETM,ALFM,ALFQ,BETQ,HIGHT,HTHETA,DR,JCELL,
    &WTAU)
C    HBAR<==>1, T<==>2, MU<==>3, EPS<==>4, TAU<==>X ********************
C    WTAU=X,X2,X4,X22,X44,X23,X24,X222,X444,X233 / X244,X234,X223,X224,
C    &X1,X11,X12,X14,X111,X112 / X114,X122,X123,X124,X144 **************
     REAL WTAU(1),COF,RQ,RMN,RO,VELOM,BETM,ALFM,ALFQ,BETQ,HIGHT,
    &HTHETA,DR,BETAO,F4,F44,F444,G,G3,Y,Y1,Y2,Y4,Y11,Y12,Y22,Y23,Y24,
    &Y44,Y233,Y234,Y244,Y444,R,RY,RYY,RYYY,VZ,F
     INTEGER JQ,JCELL
     VZ=-VELOM*COS(BETM)
     COF=2./3.0E8
     RQ=HIGHT/COS(BETQ)
     RMN=HIGHT/COS(BETQ-HTHETA)
     JQ=INT((RQ-RMN+DR/2.)/DR)
     RO=RQ+(JCELL-JQ)*DR
     BETAO=ACOS(HIGHT/RO)
     F=HIGHT*TAN(BETAO)
     F4=HIGHT/COS(BETAO)**2
     F44=2.*HIGHT*SIN(BETAO)/COS(BETAO)**3
     F444=2.*HIGHT*(1.+2.*SIN(BETAO)**2)/COS(BETAO)**4
     G=2.*VELOM*SIN(BETM)*COS(ALFQ-ALFM)
     G3=-2.*VELOM*SIN(BETM)*SIN(ALFQ-ALFM)
```

```
Y=F**2+HIGHT**2
R=Y**.5
RY=.5/R
RYY=-.25/R**3
RYYY=0.375/R**5
Y1=-2.*HIGHT
Y2=-F*G+2.*HIGHT*VZ
Y4=2.*F*F4
Y11=2.
Y12=-2.*VZ
Y22=2.*VELOM**2
Y23=-F*G3
Y24=-F4*G
Y44=2.*F4**2+2.*F*F44
Y233=F*G
Y234=-F4*G3
Y244=-F44*G
Y444=6.*F4*F44+2.*F*F444
WTAU(1)=COF*R
WTAU(2)=COF*RY*Y2
WTAU(3)=COF*RY*Y4
WTAU(4)=COF*(RY*Y22+RYY*Y2**2)
WTAU(5)=COF*(RY*Y44+RYY*Y4**2)
WTAU(6)=COF*RY*Y23
WTAU(7)=COF*(RY*Y24+RYY*Y2*Y4)
WTAU(8)=COF*(3.*RYY*Y2*Y22+RYYY*Y2**3)
WTAU(9)=COF*(RY*Y444+3.*RYY*Y4*Y44+RYYY*Y4**3)
WTAU(10)=COF*RY*Y233
WTAU(11)=COF*(RY*Y244+RYY*(2.*Y4*Y24+Y2*Y44)+RYYY*Y4**2*Y2)
WTAU(12)=COF*(RY*Y234+RYY*Y4*Y23)
WTAU(13)=COF*2.*RYY*Y2*Y23
WTAU(14)=COF*(RYY*(2.*Y2*Y24+Y4*Y22)+RYYY*Y2**2*Y4)
WTAU(15)=COF*RY*Y1
WTAU(16)=COF*(RY*Y11+RYY*Y1**2)
WTAU(17)=COF*(RY*Y12+RYY*Y1*Y2)
WTAU(18)=COF*RYY*Y1*Y4
WTAU(19)=COF*(3.*RYY*Y1*Y11+RYYY*Y1**3)
WTAU(20)=COF*(RYY*(2.*Y1*Y12+Y2*Y11)+RYYY*Y1**2*Y2)
WTAU(21)=COF*(RYY*Y4*Y11+RYYY*Y1**2*Y4)
WTAU(22)=COF*(RYY*(2.*Y2*Y12+Y1*Y22)+RYYY*Y2**2*Y1)
WTAU(23)=COF*RYY*Y1*Y23
WTAU(24)=COF*(RYY*(Y1*Y24+Y4*Y12)+RYYY*Y1*Y2*Y4)
WTAU(25)=COF*(RYY*Y1*Y44+RYYY*Y4**2*Y1)
RETURN
END
```

SUBROUTINE TSIGNAL.FOR

```
SUBROUTINE TSIGNAL(OMEGO,NFFT,RANCOF,XIFXMT,WTSMO,WTAU,WTIME,
&WAGAIN,TXD,TXQ)
```

```
      REAL WTSMO(1),WTAU(1),WTIME(1),WAGAIN(1),TXD(1),TXQ(1),TAMP,AMP
      DOUBLE PRECISION PSI,TPI,TPSI,COF0,COF1,COF2,TIM
      INTEGER KF
      TPI=6.283185307
      COF0=-OMEGO*(WTAU(1)+WTAU(15)*WTSMO(5))+XIFXMT
      COF1=-OMEGO*(WTAU(6)*WTSMO(3)+.5*WTAU(10)*WTSMO(3)**2+WTAU(17)*
     &WTSMO(5))
      COF2=-0.5*OMEGO*(WTAU(4)+WTAU(13)*WTSMO(3))
      TAMP=WTSMO(1)*RANCOF
      DO 5 KF=1,NFFT
      TIM=WTIME(KF)
      AMP=TAMP*(WAGAIN(1)+WAGAIN(2)*TIM+WAGAIN(3)*TIM*TIM)
      TPSI=COF0+COF1*TIM+COF2*TIM**2
      PSI=DMOD(TPSI,TPI)
      TXD(KF)=AMP*DCOS(PSI)
      TXQ(KF)=AMP*DSIN(PSI)
5     CONTINUE
      RETURN
      END
```

SUBROUTINE WCHANGE.FOR

```
      SUBROUTINE WCHANGE(W)
      REAL W(1)
      INTEGER K
1     PRINT*,'ENTER INDEX OF W TO CHANGE; 0 MEANS NO CHANGE '
      CALL KINPUT(K)
      IF(K.EQ.0) RETURN
      CALL XINPUT(W(K))
      GO TO 1
      END
```

SUBROUTINE WINPUT.FOR

```
      SUBROUTINE WINPUT(NW,W,IDFILE,TITLE)
      CHARACTER*240 TITLE
      CHARACTER*55 WCHAR(51)
      REAL W(1)
      INTEGER KW,K
      DO 7 KW=1,NW
7     READ(IDFILE,11) W(KW),WCHAR(KW)
11    FORMAT(F16.7,A55)
      PRINT*,'ARRAY W BEFORE UPDATE'
      WRITE(*,'(1X,A240)') TITLE
      WRITE(*,2) (W(K),K=1,NW)
2     FORMAT(4(2XF16.7))
      CALL WCHANGE (W)
```

```
      REWIND IDFILE
      DO 8 KW=1,NW
8     WRITE(IDFILE,'(F16.7,A55)') W(KW),WCHAR(KW)
      CLOSE(IDFILE)
      RETURN
      END
```

SUBROUTINE XINPUT.FOR

```
      SUBROUTINE XINPUT(X)
      REAL X
      WRITE(*,'(1X,"X=?",\)')
      READ(*,'(F16.7)')X
      RETURN
      END
```

APPENDIX F

Listings for POLONFRA and Associated Subprograms

This Appendix contains the listings for POLONFRA and its associated subprograms, except for SORTRI which is in Appendix E.

SUBROUTINE POLONFRA.FOR

```
      SUBROUTINE POLONFRA(NBA,WAX,WAY,WPAR,IIO,JJO,IMX,JMX,TAREA,WWAREA,
     &IOKRO,IFAIL)
C     (WAX,WAY)=ABSOLUTE IN METERS, (XMN,XMX,IIO,JJO,ZDIST)=ABSOLUTE IN
C     DELMAP UNITS,(A,WX,WY,WG)=RELATIVE IN DELMAP UNITS **************
      REAL WAX(1),WAY(1),WAXT(5),WAYT(5),WPAR(1),A(5,2),ZDIST(54),
     &WWAREA(0:15,0:15),DHORI(5),DVERT(5),DSIDE(5),WG(2),WXRD(54),WYRD
     &(54),EPS,EPSH,OLDDST
      INTEGER LOC2L1(54),LOCCL1(54),LOCRLC(54),IRECW(54),IDROTW(16),
     &NBCINR(0:15,0:15),IHVAW(54),INTCOR(0:16,0:16),KONTZ,IDREC
      COMMON /FIRST/KONTZ,INTCOR,WXRD,WYRD,IHVAW,EPS,EPSH /SECOND/OLDDST
     &,WG,DSIDE,DHORI,DVERT
      REAL AREA,AREAPO,FIG,FJG,SUMX,SUMY,TAREA,TXMN,TYMN,XMED,XMN,XMX,
     &YMED,YMN,YMX
      INTEGER I,IC,IMXM,INEXT,IOKRO,J,JC,JMXM,LKONT,K,KONTC,K1,K2,LC,
     &LCP,LR,L1,L1A,L1B,L2,NR,N4
      EXTERNAL AREPOL
      DO 1 K=1,NBA
      WAXT(K)=(WAX(K)-WPAR(3))/WPAR(1)
1     WAYT(K)=(WAY(K)-WPAR(4))/WPAR(1)
C     IF CONVEX & COUNTERCLOCKWISE ROTATION ,IOKRO=0, THEN SKIP CONROT *
      IF(IOKRO.NE.0) CALL CONROT(NBA,WAXT,WAYT,IFAIL)
      IF(IFAIL.EQ.1) GO TO 55
      EPS=.0001
      EPSH=EPS/2.
      XMN=WAXT(1)
      YMN=WAYT(1)
      DO 5 K2=2,NBA
      TXMN=WAXT(K2)
      TYMN=WAYT(K2)
      IF(TXMN.LT.XMN) XMN=TXMN
      IF(TYMN.LT.YMN) YMN=TYMN
5     CONTINUE
      IIO=INT(XMN-0.0001)
      JJO=INT(YMN-0.0001)
C     TRANSFER THE NBA-APEXES INTO A(3,2) IN RELATIVE DELMAP UNITS *****
      DO 3 K1=1,NBA
      A(K1,1)=WAXT(K1)-FLOAT(IIO)
3     A(K1,2)=WAYT(K1)-FLOAT(JJO)
```

```
      TAREA=AREPOL(NBA,WAXT,WAYT)
      IF(TAREA.LT.EPS) RETURN
C     COMPUTE UPPER MAP INDEXES FOR FRAME : (IMX,JMX) ******************
      XMX=A(1,1)
      YMX=A(1,2)
      DO 7 K2=2,NBA
      IF(A(K2,1).GT.XMX) XMX=A(K2,1)
      IF(A(K2,2).GT.YMX) YMX=A(K2,2)
    7 CONTINUE
      IMX=INT(XMX+.99999)
      JMX=INT(YMX+.99999)
      IMXM=IMX-1
      JMXM=JMX-1
C     COMPUTE THE CENTER OF GRAVITY WG **********************************
      SUMX=0.
      SUMY=0.
      DO 8 K2=1,NBA
      SUMX=SUMX+A(K2,1)
    8 SUMY=SUMY+A(K2,2)
      WG(1)=SUMX/FLOAT(NBA)
      WG(2)=SUMY/FLOAT(NBA)
C     INITIALIZATION OF INTCOR ******************************************
      DO 9 K1=0,IMX
      DO 9 K2=0,JMX
    9 INTCOR(K1,K2)=0
C     INITIALIZE WWAREA *************************************************
      DO 33 I=0,IMXM
      DO 33 J=0,JMXM
   33 WWAREA(I,J)=0.
C     FIND INTERSECTS AND INTERIOR CORNERS FOR EACH SIDE OF THE POLYGON
C     SEQUENTIALLY (A1 TO A2, A2 TO A3, A3 TO A1,... A(NBA-1) TO A(NBA))
      KONTZ=0
      OLDDST=0.0
      DO 10 K1=1,NBA
      K2=K1+1
      IF(K2.EQ.(NBA+1)) K2=1
      DSIDE(K1)=SQRT((A(K1,1)-A(K2,1))**2+(A(K1,2)-A(K2,2))**2)
      DHORI(K1)=ABS(A(K1,1)-A(K2,1))
      DVERT(K1)=ABS(A(K1,2)-A(K2,2))
C     COMPUTING WXRD(L1),WYRD(L1),IHVAW(L1),ZDIST(L1) FOR L1=1,KONTZ *****
      CALL LINEAB(IMX,JMX,K1,A(K1,1),A(K1,2),A(K2,1),A(K2,2),ZDIST)
   10 CONTINUE
      IF(KONTZ.GT.NBA) GO TO 4
C     KONTZ=NBA MEANS THE POLYGON IS WITHIN ONE RECTANGLE ***************
      I=INT(WG(1))
      J=INT(WG(2))
      WWAREA(I,J)=TAREA
      RETURN
    4 LKONT=KONTZ
      CALL SORTRI(ZDIST,LOC2L1,LKONT)
C     AFTER SORTRI ZDIST(L2) INCREASES WITH L2 WHERE L2=1,KONTZ
C     AND L1=LOC2L1(L2) *************************************************
C     FOR AN INTERIOR CORNER INTCOR(I,J)=NBA, MAKE IT 1 *****************
```

```
      DO 25 IC=0,IMX
      DO 25 JC=0,JMX
 25   INTCOR(IC,JC)=INTCOR(IC,JC)/NBA
C     ROTATE CC ON  CONTOUR AND MARK CORNER POINTS **********************
      DO 13 L2=1,KONTZ
      L1=LOC2L1(L2)
      IC=INT(WXRD(L1)+.49999)
      JC=INT(WYRD(L1)+.49999)
      IF(ABS(WXRD(L1)-IC)+ABS(WYRD(L1)-JC).GT.EPS) GO TO 13
      WXRD(L1)=FLOAT(IC)
      WYRD(L1)=FLOAT(JC)
      INTCOR(IC,JC)=1
 13   CONTINUE
C     IF 4 INTERIOR CORNERS NBCINR(I,J)=4,THEN AREA = 1. ***********
      DO 23 I=0,IMX
      DO 23 J=0,JMX
      NBCINR(I,J)=INTCOR(I,J)+INTCOR(I+1,J)+INTCOR(I,J+1)+
     &INTCOR(I+1,J+1)
      IF(NBCINR(I,J).EQ.4) WWAREA(I,J)=1.
 23   CONTINUE
C     CONTOUR IS COMPRESSED FROM KONTZ TO KONTC BY COPYING ONLY ID OF **
C     POINT IFF CC DIST TO PRIOR POINT.LT.EPSH. A COMPRESSED APEX FALLS
C     ON THE RECTANGLE GRID. OUTPUT IS LOCCL1;L1=LOCCL1(LC) ************
      CALL COMPRESZ(LOC2L1,KONTC,LOCCL1)
C     ROTATE CC ON COMPRESSED CONTOUR POINT ID IS LC=1,KONTC************
C     SEGMENT(A(LC),B(LC+1)) IS BOUND BY RECTANGLE IDREC:IRECW(LC)=IDREC
C     SKIP IDREC IF(A,B) IS ON THE RECTANGULAR GRID **********************
      NR=0
      DO 15 LC=1,KONTC
      LCP=LC+1
      IF (LCP.GT.KONTC) LCP=LCP-KONTC
      L1A=LOCCL1(LC)
      L1B=LOCCL1(LCP)
      XMED=0.5*(WXRD(L1A)+WXRD(L1B))
      YMED=0.5*(WYRD(L1A)+WYRD(L1B))
      FIG=FLOAT(INT(XMED+0.49999))
      FJG=FLOAT(INT(YMED+0.49999))
      IF(ABS(XMED-FIG).LT.EPSH.OR.ABS(YMED-FJG).LT.EPSH) GO TO 15
      NR=NR+1
      I=INT(XMED)
      J=INT(YMED)
      IDREC=J*IMX+I
      IF(NBCINR(I,J).EQ.4) IDREC=IDREC+1000
      IRECW(NR)=IDREC
      LOCRLC(NR)=LC
 15   CONTINUE
C     SORT IRECW FOR INCREASING IDREC ,BACKTRACK ARRAY IS LOC3LC *******
      CALL SORTII(IRECW,LOCRLC,-NR)
C     AFTER SORTII IRECW(LR) WHERE LR=1,NR AND LC=LOCRLC(LC) ********
      N4=0
      IRECW(NR+1)=99999
      DO 50 LR=1,NR
C     IF RECTANGLE IDREC BOUNDS TWO SEGMENTS(A1B1,A2B2),IDREC=IDRECW(LR)
```

```
C     =IRECW(LR+1),LC(A1)=LOCRLC(LR),LC(A2)=LOCRLC(LR+1). LCs FOR SAME *
C     IDREC ARE STORED IN IDROTW={LC(A1),LC(B1)=LC(A1)+1,LC(A2),LC(B2)=
C     LC(A2)+1}. SKIP IF IDREC.GE.1000 (RECTANGLE ALREADY MAPPED) ******
      IDREC=IRECW(LR)
      IF(IDREC.GE.1000) GO TO 50
      N4=N4+1
      LC=LOCRLC(LR)
      IDROTW(N4)=LC
      N4=N4+1
      LCP=LC+1
      IF(LCP.GT.KONTC) LCP=LCP-KONTC
      IDROTW(N4)=LCP
      INEXT=IRECW(LR+1)
      IF(INEXT.EQ.IDREC) GO TO 50
      I=MOD(IDREC,IMX)
      J=INT(IDREC/IMX)
C     AFTER SORTI IDROTW(L5) INCREASES WITH LR (INCREASING LCs )********
      CALL SORTI(IDROTW,N4)
      CALL RECTAREA(I,J,N4,KONTC,LOCCL1,IDROTW,AREA)
      WWAREA(I,J)=AREA
      N4=0
 50   CONTINUE
      RETURN
 55   WRITE(6,*)'POLYGON NOT CONVEX, OR 3 POINTS ON ONE LINE !!!!!!!!!!'
      RETURN
      END
```

FUNCTION AREPOL.FOR

```
      FUNCTION AREPOL(NB,X,Y)
      REAL AREA,ATRI,XF,XR,XS,X1,X2,X3,X(1),YF,YR,YS,Y1,Y2,Y3,Y(1)
      INTEGER K,NBM
      ATRI(X1,Y1,X2,Y2,X3,Y3)=0.5*ABS(X1*Y2+X2*Y3+X3*Y1-Y1*X2-Y2*X3-
     &Y3*X1)
      IF(NB.LE.2) GO TO 3
      AREA=0.
      XR=X(1)
      YR=Y(1)
      NBM=NB-1
      DO 5 K=2,NBM
      XF=X(K)
      YF=Y(K)
      XS=X(K+1)
      YS=Y(K+1)
      AREA=AREA+ATRI(XR,YR,XF,YF,XS,YS)
 5    CONTINUE
      AREPOL=AREA
      RETURN
 3    PRINT*,'ERROR IN AREPOL: NBA<3'
      RETURN
```

```
      END
```

SUBROUTINE COMPRESZ.FOR

```
      SUBROUTINE COMPRESZ(LOC2L1,KONTC,LOCCL1)
      INTEGER LOC2L1(1),LOCCL1(1),IHVAW(54),INTCOR(0:16,0:16),KONTZ,L1,
     &L1M,L2,L2M
      REAL WXRD(54),WYRD(54),EPS,EPSH,DIFDIS
      COMMON /FIRST/KONTZ,INTCOR,WXRD,WYRD,IHVAW,EPS,EPSH
      KONTC=0
      L2M=KONTZ
      L1M=LOC2L1(L2M)
      DO 3 L2=1,KONTZ
      L1=LOC2L1(L2)
      DIFDIS=ABS(WXRD(L1)-WXRD(L1M))+ABS(WYRD(L1)-WYRD(L1M))
      IF(DIFDIS.GT.EPSH) GO TO 5
      IF(IHVAW(L1).LE.2) LOCCL1(KONTC)=LOC2L1(L2)
      GO TO 7
5     KONTC=KONTC+1
      LOCCL1(KONTC)=LOC2L1(L2)
7     L2M=L2
      L1M=LOC2L1(L2M)
3     CONTINUE
      RETURN
      END
```

SUBROUTINE CONROT.FOR

```
      SUBROUTINE CONROT(NBA,WAX,WAY,IFAIL)
      REAL WAX(1),WAY(1),WAXT(5),WAYT(5),T
      INTEGER K,KONCC,KONCW,MA,MB,MC
      IFAIL=0
      KONCW=0
      KONCC=0
      DO 1 K=1,NBA
      MA=MOD((K-1),NBA)+1
      MB=MOD(K,NBA)+1
      MC=MOD((K+1),NBA)+1
      T=(WAX(MC)-WAX(MB))*(WAY(MA)-WAY(MB))-(WAX(MA)-WAX(MB))*(WAY(MC)
     &-WAY(MB))
      IF(T.EQ.0.) GO TO 3
      IF(T.LT.0.) KONCW=KONCW+1
      IF(T.GT.0.) KONCC=KONCC+1
1     CONTINUE
      IF(KONCC.NE.0.AND.KONCW.NE.0) GO TO 3
      IF(KONCC.GT.0) RETURN
      DO 4 K=1,NBA
```

```
      WAXT(K)=WAX(K)
4     WAYT(K)=WAY(K)
      DO 5 K=2,NBA
      WAX(K)=WAXT(NBA+2-K)
5     WAY(K)=WAYT(NBA+2-K)
      RETURN
3     IFAIL=1
      RETURN
      END
```

SUBROUTINE INSERT.FOR

```
      SUBROUTINE INSERT(K,KP,XR,YR,NBINS,AX,AY)
      REAL AX(1),AY(1),XR(1),YR(1),DISTA,DIST,DISTB
      INTEGER IDISTA,IDISTB,KD,KDT
C     COMPUTE IP1=LDROTW(K),IP2=LDROTW(KP) ON TRIANGLE CONTOUR COUNT
C     INTERIOR CORNERs  COUNTERCLOCKWISE ,#= NBINS,STORE RELATIVE X,Y
C     INTO AX,AY. COORDINATES OF IP1,IP2 ARE IN XR,YR **********
      NBINS=0
      CALL XYDIS(XR(K),YR(K),DISTA,1)
      CALL XYDIS(XR(KP),YR(KP),DISTB,1)
      IF(DISTB.LT.DISTA) DISTB=DISTB+4.
      IDISTA=INT(DISTA+1.00001)
      IDISTB=INT(DISTB-0.00001)
      IF(IDISTB.LT.IDISTA) RETURN
      DO 5 KD=IDISTA,IDISTB
      KDT=MOD(KD,4)
      DIST=FLOAT(KDT)
      NBINS=NBINS+1
      CALL XYDIS(AX(NBINS),AY(NBINS),DIST,2)
5     CONTINUE
      RETURN
      END
```

SUBROUTINE LINEAB.FOR

```
      SUBROUTINE LINEAB(IMX,JMX,K1,AX,AY,BX,BY,ZDIST)
      INTEGER IHVAW(54),INTCOR(0:16,0:16),KONTZ,I,IA,IB,ICOR,INCI,INCJ,
     &J,JA,JB,JCOR
      REAL DHORI(5),DVERT(5),DSIDE(5),ZDIST(1),WG(2),WXRD(54),WYRD(54),
     &EPS,EPSH,OLDDST,ONLIN,PTOA,PXTOAX,PYTOAY,REFSIG,TSIGN,TX,TY,UI,UJ,
     &VADI,VADJ,X,Y
      COMMON /FIRST/KONTZ,INTCOR,WXRD,WYRD,IHVAW,EPS,EPSH /SECOND/OLDDST
     &,WG,DSIDE,DHORI,DVERT
      ONLIN(X,Y,AX,AY,BX,BY)=(AX-X)*(BY-Y)-(AY-Y)*(BX-X)
C     TESTING THE CENTER OF GRAVITY WRT THE LINE A-B **********
      REFSIG=ONLIN(WG(1),WG(2),AX,AY,BX,BY)
```

```
C    ENTERING INFO ON APEX A ********************************************
     KONTZ=KONTZ+1
     ZDIST(KONTZ)=OLDDST
     WXRD(KONTZ)=AX
     WYRD(KONTZ)=AY
     IHVAW(KONTZ)=7
C    COMPUTE UNIT VECTORS UJ ON Y,AND UI ON X ***************************
     UJ=(BY-AY)/DSIDE(K1)
     UI=(BX-AX)/DSIDE(K1)
     VADI=0.5+SIGN(0.5,UI)
     VADJ=0.5+SIGN(0.5,UJ)
     IA=INT(AX+VADI)
     IB=INT(BX+1.-VADI)
     JA=INT(AY+VADJ)
     JB=INT(BY+1.-VADJ)
     IF(ABS(UI).LT.1.E-6) GO TO 6
C    ENTERING INFO ON VERTICAL INTERSECTS *******************************
     INCI=1
     IF(UI.LT.0.) INCI=-1
     DO 3 I=IA,IB,INCI
     PXTOAX=ABS(I-AX)
     PTOA=DSIDE(K1)*PXTOAX/DHORI(K1)
     KONTZ=KONTZ+1
     ZDIST(KONTZ)=PTOA+OLDDST
     WXRD(KONTZ)=FLOAT(I)
     WYRD(KONTZ)= AY+UJ*PTOA
     IHVAW(KONTZ)=2
 3   CONTINUE
 6   IF(ABS(UJ).LT.1.E-6) GO TO 10
C    ENTERING INFO ON HORIZONTAL INTERSECTS *****************************
     INCJ=1
     IF(UJ.LT.0) INCJ=-1
     DO 5 J=JA,JB,INCJ
     PYTOAY=ABS(J-AY)
     PTOA=DSIDE(K1)*PYTOAY/DVERT(K1)
     KONTZ=KONTZ+1
     ZDIST(KONTZ)=PTOA+OLDDST
     WXRD(KONTZ)=AX+UI*PTOA
     WYRD(KONTZ)= FLOAT(J)
 5   IHVAW(KONTZ)=1
 10  CONTINUE
C    INCREMENT INTCOR BY 1 IF CORNER (ICOR,JCOR) INTERIOR TO LINE AB
     DO 15 ICOR=0,IMX
     DO 15 JCOR=0,JMX
     TX=FLOAT(ICOR)
     TY=FLOAT(JCOR)
     TSIGN=ONLIN(TX,TY,AX,AY,BX,BY)
     IF((TSIGN*REFSIG).GT.0.) INTCOR(ICOR,JCOR)=INTCOR(ICOR,JCOR)+1
 15  CONTINUE
     OLDDST=OLDDST+DSIDE(K1)
     RETURN
     END
```

SUBROUTINE RECTAREA.FOR

```
      SUBROUTINE RECTAREA(I,J,N4,KONTC,LOCCL1,IDROTW,AREA)
C     COMPUTE OVERLAP AREA OF TRIANGLE & RECTANGLE AT (I,J) ***********
      INTEGER IDROTW(1),LOCCL1(1),LDROTW(16),IHVAW(54),INTCOR(0:16,0:16)
     &,LHVAW(16),LDROTX(14),K,KIN,KON,KONX,KONTZ,KP,LC,L1,NBINS,NBINSC
      REAL XR(16),YR(16),XRX(14),YRX(14),AX(4),AY(4),WXRD(54),WYRD(54),
     &AREAPO,EPS,EPSH
      EXTERNAL AREPOL
      COMMON /FIRST/KONTZ,INTCOR,WXRD,WYRD,IHVAW,EPS,EPSH
C     COMPRESS IDROTW INTO LDROTW FOR REPEATED APEX ( FROM N4 TO KON )**
      KON=0
      DO 3 K=1,N4
      KP=K+1
      IF(KP.GT.N4) KP=KP-N4
      IF (IDROTW(KP).EQ.IDROTW(K)) GO TO 3
      KON=KON+1
      LDROTW(KON)=IDROTW(K)
    3 CONTINUE
C     COMPUTE LOCAL XY-COORDINATES IN (XR,YR),IHVAW IDENTITY IN LHVAW.**
      DO 5 K=1,KON
      LC=LDROTW(K)
      L1=LOCCL1(LC)
      XR(K)=WXRD(L1)-FLOAT(I)
      YR(K)=WYRD(L1)-FLOAT(J)
    5 LHVAW(K)=IHVAW(L1)
C     EXPAND ARRAYS (LDROTW,XR,YR) INTO (LDROTX,XRX,YRX) FROM KON TO
C     KONX BY INSERTING CORNERS ON THE CONTOUR ***************************
      KONX=0
      DO 10 K=1,KON
      KP=K+1
      IF(KP.GT.KON) KP=KP-KON
      KONX=KONX+1
      LDROTX(KONX)=LDROTW(K)
      XRX(KONX)=XR(K)
      YRX(KONX)=YR(K)
C     IF NO GAP IN IDROT THEN SKIP INSERT ********************************
      IF((LDROTW(KP)-LDROTW(K)).EQ.1.OR.(LDROTW(KP)-LDROTW(K)).EQ.
     &1-KONTC) GO TO 10
      CALL INSERT(K,KP,XR,YR,NBINS,AX,AY)
      IF(NBINS.EQ.0) GO TO 10
      DO 15 KIN=1,NBINS
      KONX=KONX+1
      LDROTX(KONX)=KIN+100
      XRX(KONX)=AX(KIN)
   15 YRX(KONX)=AY(KIN)
   10 CONTINUE
      AREA=AREPOL(KONX,XRX,YRX)
      RETURN
      END
```

SUBROUTINE SORTI.FOR

```
      SUBROUTINE SORTI (IVAL,N)
      INTEGER IVAL(1),M,K,J,I,II,F
      M=N
1     CONTINUE
      M=M/2
      IF(M.EQ.0) RETURN
      K=N-M
      J=1
2     CONTINUE
      I=J
3     CONTINUE
      II=I+M
      IF(IVAL(I).LT.IVAL(II)) GO TO 4
      F=IVAL(I)
      IVAL(I)=IVAL(II)
      IVAL(II)=F
      I=I-M
      IF(I.GE.1) GO TO 3
4     CONTINUE
      J=J+1
      IF(J.GT.K) GO TO 1
      GO TO 2
      RETURN
      END
```

SUBROUTINE SORTII.FOR

```
      SUBROUTINE SORTII (VAL,XLOC,NS)
      INTEGER VAL(1),XLOC(1),N,K,M,J,I,II,F,XS
      N=IABS(NS)
      IF(NS.LT.0) GO TO 7
      DO 6 K=1,N
6     XLOC(K)=K
7     M=N
1     CONTINUE
      M=M/2
      IF(M.EQ.0) RETURN
      K=N-M
      J=1
2     CONTINUE
      I=J
3     CONTINUE
      II=I+M
      IF(VAL(I).LT.VAL(II)) GO TO 4
```

```
      F=VAL(I)
      XS=XLOC(I)
      VAL(I)=VAL(II)
      XLOC(I)=XLOC(II)
      VAL(II)=F
      XLOC(II)=XS
      I=I-M
      IF(I.GE.1) GO TO 3
    4 CONTINUE
      J=J+1
      IF(J.GT.K) GO TO 1
      GO TO 2
      RETURN
      END
```

SUBROUTINE XYDIS.FOR

```
      SUBROUTINE XYDIS(X,Y,DIS,KXYDIS)
      INTEGER IDIS
C     KXYDIS=1 FOR X,Y TO DIS, KXYDIS=2 FOR DIS TO X,Y **********
      GO TO (1,2),KXYDIS
    1 IF(X.GE.Y) DIS=X+Y
      IF(Y.GT.X) DIS=4.-X-Y
      RETURN
    2 IDIS=INT(DIS+1.00001)
      IF(DIS.LT.0.) DIS=DIS+4.
      IF(DIS.GT.4.) DIS=DIS-4.
      GO TO (4,5,6,7),IDIS
    4 X=DIS
      Y=0.
      RETURN
    5 X=1.
      Y=DIS-1.
      RETURN
    6 X=3.-DIS
      Y=1.
      RETURN
    7 X=0.
      Y=4.-DIS
      RETURN
      END
```

APPENDIX G

Listings for CRRDP and Associated Subprograms

This Appendix contains the listings for CRRDP and for the associated subprograms in alphabetical order. Excluded are: the subprograms which are already listed in Appendices C, D, E, and F.

PROGRAM CRRDP.FOR

```
C ASSUME NBSCA IN GROUNDMAP AND A FOOTPRINT AT WQ9 ,1/16/89, 1200 hr
$LARGE
      CHARACTER*240 TITLE
      REAL W(27),TIMCNT,TIMDWL,VELOM,ALFM,BETM,WMPOSC(3),WQ9(6),DELRAN,
     &FFNBIT,WPAR(6),XIFXMT,SELWIN,BETKAI,SELDIS,BITINT
      REAL SMOMAP(450),TXD(64),TXQ(64),
     &WADRLO(165),WADRUP(165),WADSMO(90),WAGAIN(3),WANGN9(165),
     &WAGNCL(128),WBET9(165),WDUM(3),WDOVL(3),WEPSMX(165),WIND(64)
      REAL WMPOSB(3),WMPOSE(3),WMU3DB(165),WMUMDR(166),WMVELO(3),
     &WRAN9(165),WTAU(25),WTIME(64),WTSMO(5),WWAPEX(128,9),
     &XD(128),XM(128),XQ(128)
      INTEGER LNPPP(128)
      REAL DISH,DOTCOS,FRQOFM,FREQ
      REAL CLIGHT,DELFIN,DELFUT,DELTIM,DOVL,EPS,FMU,FREQO,GDISH2,
     &GDISH3,GRC,HDELRA,HTHETA,HTIMDW,RANCOF,RANMN,RANMX,OMEGO,SCAFAC,
     &SINANG,TEMP,THETA,TMPBET,TPI,V1,V2,V3,X.XSCA,YSCA,Y,Z,FZMX
      INTEGER IDDISH,IFILE,INDX,INTCOF,IW,JRAN,JRANQ9,JRANMX,JW,K,
     &KADSMO,KDQ,KI,KJ,KFF,KNPPP,KSCALO,KSCAUP,KT,KTIM,KTHSCA,
     &NBITFF,NBITFX,NFFT,NFFTH,NFFX,NFFXH,NPPPMN,NPPPMX,NSCAFP,NXMAP,
     &NYMAP
      REAL SYNMAP(0:99,0:99),SUMARA(0:99,0:99)
      COMMON /MAP/SYNMAP,SUMARA
      EQUIVALENCE (W(1),TIMCNT),(W(2),TIMDWL),(W(3),VELOM),(W(4),ALFM),
     &(W(5),BETM),(W(6),WMPOSC(1)),(W(9),WQ9(1)),(W(15),DELRAN),(W(16),
     &FFNBIT),(W(17),WPAR(1)),(W(23),XIFXMT),(W(24),SELWIN),(W(25),BETKA
     &I),(W(26),SELDIS),(W(27),BITINT)
C     PARAMETERS FOR OUTPUT MAP ARE IN WPAR=(DELMAP,THRESH,XMNMAP,YMNMAP
C     ,XMXMAP,YMXMAP) ****************************************************
C     WQ9=(XQ,XQ,ZQ,ALFQ9,BETQ9,RANQ9) ***********************************
C     DIMENSION OF SMOMAP = NBSCA*15 (SEE SUB. CLUTINFO) *****************
C     INPUT FILES: CRRDPI.INP AND CLUTINFI.INP ***************************
C     OUTPUT FILES: CRRDPG.GRD AND CRRDPP.PRT ****************************
      FRQOFMU(V1,FMU,V2,V3)=-V1*(V2*FMU+(V3/2.)*FMU**2)
      OPEN(UNIT=11,FILE='CLUTINFI.INP',STATUS='OLD')
      OPEN(UNIT=9,FILE='CRRDPI.INP',STATUS='OLD')
      OPEN(UNIT=3,FILE='CRRDPP.PRT',STATUS='OLD')
```

```
      OPEN(UNIT=7,FILE='CRRDPG.GRD',STATUS='OLD')
C     HALF 3db BEAMWIDTH=.0215 gives DOVL=14.83343168=DISH DIAMETER IN
C     &LAMBDA UNITS ( 2-WAY GAIN = -3.00 DB means DISH=0.84139514)*******
      DATA THETA,FREQO,CLIGHT,TPI,EPS,IFILE,WDOVL/.043,94.E9,3.E8,
     &6.28318531,.0001,3,17.15094,17.15094,14.80884/
      SINANG=SIN(0.0215)
      GDISH2=DISH(SINANG,2,WDOVL(2))
      GDISH3=DISH(SINANG,3,WDOVL(3))
      WRITE(3,*)'GDISH2,GDISH3',GDISH2,GDISH3
      TITLE='TIMCNT=1; TIMDWL=2; VELOM=3; ALFM=4; BETM=5; WMPOSC=6-8; WQ
     &9=9-10          DUMMY=11-14;DELRAN=15;FFNBIT=16;WPAR1=17=DELM
     &AP;WPAR2=18=THRESH;WPAR3=19=XMN ;WPAR4=20=YMN/ WPAR5=21=XMX; WPAR6
     &=22=YMX; XIFXMT=23,SELWIN=24; BETKAI=25;SELDIS=26; BITINT=27'
      CALL WINPUT(27,W,9,TITLE)
      CALL PRTXY(W,1,27,28,IFILE)
C     READJUST (XMNMAP,YMNMAP) TO BE INTEGER MULTIPLE OF DELMAP ********
      DO 3 K=3,4
3     WPAR(K)=WPAR(1)*FLOAT(INT(WPAR(K)/WPAR(1)))
      IDDISH=INT(SELDIS+.00001)
      DOVL=WDOVL(IDDISH)
      NXMAP=1.+(WPAR(5)-WPAR(3))/WPAR(1)
      NYMAP=1.+(WPAR(6)-WPAR(4))/WPAR(1)
      WRITE(*,*)'NXMAP=',NXMAP,'  NYMAP=',NYMAP
      OMEGO=TPI*FREQO
      HDELRA=DELRAN/2.
      HTHETA=THETA/2.
      NBITFF=INT(FFNBIT+EPS)
      NBITFX=NBITFF+INT(BITINT+EPS)
      NFFT=2**NBITFF
      NFFTH=NFFT/2
      NFFX=2**NBITFX
      NFFXH=NFFX/2
      DELFIN=1./TIMDWL
      INTCOF=2**INT(BITINT+EPS)
      DELFUT=DELFIN/FLOAT(INTCOF)
      DELTIM=TIMDWL/NFFT
      DO 2 KTIM=1,NFFT
2     WTIME(KTIM)=DELTIM*(KTIM-NFFTH-0.5)
C     COMPUTE MISSILE INFORMATION ***************************************
      WMVELO(1)=VELOM*SIN(BETM)*SIN(ALFM)
      WMVELO(2)=VELOM*SIN(BETM)*COS(ALFM)
      WMVELO(3)=-VELOM*COS(BETM)
      HTIMDW=TIMDWL*.5
      DO 21 K=1,3
      WMPOSB(K)=WMPOSC(K)+WTIME(1)*WMVELO(K)
21    WMPOSE(K)=WMPOSC(K)+WTIME(NFFT)*WMVELO(K)
C     COMPUTE FOOTPRINT INFORMATION AT WQ9 ******************************
      GRC=SQRT((WQ9(1)-WMPOSC(1))**2+(WQ9(2)-WMPOSC(2))**2)
      WQ9(4)=ATAN2((WQ9(1)-WMPOSC(1)),(WQ9(2)-WMPOSC(2)))
      WQ9(5)=ATAN2(GRC,WMPOSC(3))
      WQ9(6)=SQRT(GRC**2+WMPOSC(3)**2)
      RANMN=WMPOSC(3)/COS(WQ9(5)-HTHETA)
```

```
      RANMX=WMPOSC(3)/COS(WQ9(5)+HTHETA)
      JRANQ9=INT((WQ9(6)-RANMN+HDELRA)/DELRAN)
      JRANMX=2*JRANQ9
C     STORE RANGE,BET,EPSMX,AND MU-3DB FOR FICTITIOUS SCATTERERS *******
      DO 7 JRAN=1,JRANMX
      WRAN9(JRAN)=WQ9(6)+(JRAN-JRANQ9)*DELRAN
      WBET9(JRAN)=ACOS(WMPOSC(3)/WRAN9(JRAN))
      TMPBET=ACOS(WMPOSC(3)/(WRAN9(JRAN)-HDELRA))
      WEPSMX(JRAN)=WBET9(JRAN)-TMPBET
    7 WMU3DB(JRAN)=ACOS((COS(HTHETA)-COS(WQ9(5))*COS(WBET9(JRAN)))/
     &(SIN(WQ9(5))*SIN(WBET9(JRAN)))-0.00000016)
C     INITIALIZE ARRAY SYNMAP ******************************************
      DO 13 IW=0,NXMAP-1
      DO 13 JW=0,NYMAP-1
      SYNMAP(IW,JW)=0.
   13 SUMARA(IW,JW)=0.
C     THIS IN PREPARATION FOR MAPPING **********************************
C     ENTER CLUTTER INFORMATION FOR FOOTPRINT AT WQ9 AND MAP PARAMETERS
      WRITE(3,*)'JRAMNX=',JRANMX
      CALL CLUTINFO(DELRAN,HTHETA,JRANQ9,JRANMX,WQ9,WMPOSC,WPAR,NSCAFP,
     &SMOMAP,WADRLO,WADRUP,WADSMO)
C     INDEX KNPPP=1,NFFX FOR FREQ=[-NFFXH*DELFUT,(NFFXH-1)*DELFUT] *****
C     COMPUTE ARRAY LNPPP,THEN KFF=LNPPP(KNPPP) ************************
      DO 9 KNPPP=1,NFFX
      INDX=KNPPP+NFFXH
      IF(INDX.GT.NFFX) INDX=INDX-NFFX
    9 LNPPP(KNPPP)=INDX
C     PROCESS AND MAP RANGE CELLS SEQUENTIALLY *************************
      DO 35 JRAN=1,JRANMX
      KSCALO=WADRLO(JRAN)+EPS
      IF(KSCALO.EQ.0) GO TO 35
C     INITIALIZING QUADRATURE ARRAYS=(XD,XQ) ***************************
      DO 27 K=1,NFFX
      XD(K)=0.
   27 XQ(K)=0.
      WDUM(3)=0.
C     COMPUTE TAYLOR COEFFICIENTS WTAU=(TAU,TAUt,TAUe,TAUtt,TAUee,TAUtm,
C     TAUte,TAUttt,TAUeee,TAUtmm,TAUtee,TAUtme,TAUttm,TAUtte) **********
      CALL TAYLOR(VELOM,BETM,ALFM,WQ9(4),WQ9(5),WMPOSC(3),HTHETA,
     &DELRAN,JRAN,WTAU)
      WRITE(3,*)'ARRAY WTAU'
      CALL PRTXY(WTAU,1,25,119,3)
C     PREPARATION FOR MAPPING: COMPUTE (NPPPMN,NPPPMX) ON 3DB FOOTPRINT
C     TO LIMIT # CELLS BEING MAPPED TO KNPPP=NPPPMN,NPPPMX *************
      NPPPMN=FRQOFMU(FREQO,WMU3DB(JRAN),WTAU(6),WTAU(10))/DELFUT
     &+NFFXH-5.
      NPPPMX=FRQOFMU(FREQO,-WMU3DB(JRAN),WTAU(6),WTAU(10))/DELFUT
     &+NFFXH+5.
C     COMPUTE MU AT CENTER OF CELLS FOR INCREASING DOPPLER *************
      DO 17 K=NPPPMN-1,NPPPMX+1
      FREQ=DELFUT*FLOAT(K-NFFXH-1)/FREQO
   17 WMUMDR(K)=(-WTAU(6)+SQRT(WTAU(6)**2-2.*WTAU(10)*FREQ))/WTAU(10)
C     COEFFICIENT FOR RANGE NORMALIZATION: RANCOF **********************
```

```
      RANCOF=(WRAN9(JRAN)/WRAN9(JRANQ9))**4
      KSCAUP=WADRUP(JRAN)+EPS
      DO 30 KTHSCA=KSCALO,KSCAUP
      KADSMO=WADSMO(KTHSCA)+EPS
C     INFORMATION ON SCATTERER,WTSMO=(AMPSCA,FASSCA,MUSCA,JRAN-SCA,ZSCA)
      DO 31 KT=1,5
 31   WTSMO(KT)=SMOMAP(KADSMO-4+KT)
C     ANTGAIN(t)=WAGAIN(1)+WAGAIN(2)*t+WAGAIN(3)*t^2 ********************
C     WAGAIN IS OUTPUT OF SCAGAN,INPUTS ARE DOVL,HTIMDW,WMPOS=(BEG,
C    &CENT,END),WQ9,WTSMO=(AMP,FAS,MU,JRAN,ZSCA),JRANQ9,DELRAN *********
      CALL SCAGAN(IDDISH,DOVL,HTIMDW,WMPOSB,WMPOSC,WMPOSE,WQ9,WTSMO,
     &JRANQ9,DELRAN,WAGAIN)
      CALL TSIGNAL(OMEGO,NFFT,RANCOF,XIFXMT,WTSMO,WTAU,WTIME,WAGAIN,TXD,
     &TXQ)
      DO 33 KDQ=1,NFFT
      XD(KDQ)=XD(KDQ)+TXD(KDQ)
 33   XQ(KDQ)=XQ(KDQ)+TXQ(KDQ)
 30   CONTINUE
      CALL DERAMP(OMEGO,WTAU(4),NFFT,XD,XQ,WTIME)
      CALL WINDOWS(64,4,BETKAI,WIND)
      DO 37 K=1,64
      XD(K)=XD(K)*WIND(K)
 37   XQ(K)=XQ(K)*WIND(K)
      CALL FFT(-NBITFX,1,1,1,XD,XQ)
C     COMPUTE ARRAY OF ANTENNA GAIN AT CENTER OF CELLS ******************
      DO 39 KNPPP=NPPPMN,NPPPMX
      SINANG=SQRT(1.-(SIN(WQ9(5))*SIN(WBET9(JRAN))*COS(WMUMDR(KNPPP))+
     &COS(WQ9(5))*COS(WBET9(JRAN)))**2)
 39   WAGNCL(KNPPP)=DISH(SINANG,IDDISH,DOVL)**2
C     (XD,XQ) NEED TO BE SCALED FOR RANGE AND ANTENNA GAIN AT KNPPP ****
      DO 41 KNPPP=NPPPMN,NPPPMX
      KFF=LNPPP(KNPPP)
      SCAFAC=1./(RANCOF*WAGNCL(KNPPP))
      XD(KFF)=XD(KFF)*SCAFAC
      XQ(KFF)=XQ(KFF)*SCAFAC
 41   XM(KFF)=SQRT(XD(KFF)**2+XQ(KFF)**2)
C     APEXTRAP COMPUTES WWAPEX AT CENTERS OF CELLS,**********************
C     WWAPEX=(KNPPP;X1,Y1,X2,Y2,X3,Y3,X4,Y4,AREA) ***********************
      CALL APEXTRAP(NFFX,WBET9(JRAN),DELRAN,WQ9(4),WEPSMX(JRAN),NPPPMN,
     &NPPPMX,WTAU,WMPOSC,WMUMDR,WWAPEX)
C     FFT CELLS ARE MAPPED EXCEPT (BELOW THRESH.or.OUT FOOTPRT) KNPPP **
C     GOES:(NPPPMN,NPPPMX) ie:(MUMX,MUMN);FOR FFT USE KFF=LNPPP(KNPPP) *
      CALL MAPRGM(NXMAP,NYMAP,NPPPMN,NPPPMX,XM,WPAR,LNPPP,WWAPEX)
 35   CONTINUE
      FZMX=0.
      DO 60 IW=0,NXMAP-1
      DO 60 JW=0,NYMAP-1
      IF(SUMARA(IW,JW).LT.EPS) GO TO 60
      SYNMAP(IW,JW)=SYNMAP(IW,JW)/SUMARA(IW,JW)
      IF(SYNMAP(IW,JW).GT.FZMX) FZMX=SYNMAP(IW,JW)
 60   CONTINUE
      WRITE(7,'("DSAA")')
      WRITE(7,'(I2,1X,I2)')NXMAP,NYMAP
```

```
      WRITE(7,'(I2,1X,I2)')0,NXMAP-1
      WRITE(7,'(I2,1X,I2)')0,NYMAP-1
      WRITE(7,'(I2,1X,I2)')0,INT(FZMX+EPS)
      DO 69 JW=0,NYMAP-1
      WRITE(7,62) (SYNMAP(IW,JW),IW=0,NXMAP-1)
      WRITE(7,'( )')
   69  CONTINUE
   62  FORMAT(10(1XF8.3))
      CLOSE(3)
      CLOSE(9)
      CLOSE(7)
      CLOSE(11)
      STOP
      END
```

SUBROUTINE APEXTRAP.FOR

```
      SUBROUTINE APEXTRAP(NFFTUT,BET9,DELRAN,ALFQ,EPSMX,NPPPMN,NPPPMX,
     &WTAU,WMPOSC,WMUMDR,WWAPEX)
C     PROGRAM GENERATED ON 1/16/1989, 1200 hr **************************
      REAL WTAU(1),WMPOSC(1),WMUMDR(1),WWAPEX(NFFTUT,9),W(2,2,2),VARMU,
     &RANMDR,RANMNR,RANMXR,TEMP0,TEMP1
      INTEGER KTRAP,K1,K2
C     TRAPEZOIDS TO MAP:KTRAP=(NPPPMN,NPPPMX) FOR EACH KTRAP COMPUT MU
C     AT(EPSMN,EPSMX) FOR UPPER BOUNDARY (UPPER=NEW LINE,LOWER=OLD LINE)
C     FOR INITIALIZATION COMPUTE NEW LINE AT KTRAP=NPPPMN-1 ************
      VARMU=(WTAU(7)*EPSMX)/WTAU(6)
      RANMDR=WMPOSC(3)/COS(BET9)
      RANMNR=RANMDR-DELRAN/2.
      RANMXR=RANMNR+DELRAN
      TEMP0=-VARMU+(WMUMDR(NPPPMN)+WMUMDR(NPPPMN-1))*0.5
      TEMP1=TEMP0+2.*VARMU
      W(1,1,2)=SQRT(RANMNR**2-WMPOSC(3)**2)*SIN(TEMP0+ALFQ)+WMPOSC(1)
      W(2,1,2)=SQRT(RANMNR**2-WMPOSC(3)**2)*COS(TEMP0+ALFQ)+WMPOSC(2)
      W(1,2,2)=SQRT(RANMXR**2-WMPOSC(3)**2)*SIN(TEMP1+ALFQ)+WMPOSC(1)
      W(2,2,2)=SQRT(RANMXR**2-WMPOSC(3)**2)*COS(TEMP1+ALFQ)+WMPOSC(2)
      DO 15 KTRAP=NPPPMN,NPPPMX
      DO 10 K1=1,2
      DO 10 K2=1,2
   10  W(K1,K2,1)=W(K1,K2,2)
      TEMP0=-VARMU+(WMUMDR(KTRAP+1)+WMUMDR(KTRAP))*0.5
      TEMP1=TEMP0+2.*VARMU
      W(1,1,2)=SQRT(RANMNR**2-WMPOSC(3)**2)*SIN(TEMP0+ALFQ)+WMPOSC(1)
      W(2,1,2)=SQRT(RANMNR**2-WMPOSC(3)**2)*COS(TEMP0+ALFQ)+WMPOSC(2)
      W(1,2,2)=SQRT(RANMXR**2-WMPOSC(3)**2)*SIN(TEMP1+ALFQ)+WMPOSC(1)
      W(2,2,2)=SQRT(RANMXR**2-WMPOSC(3)**2)*COS(TEMP1+ALFQ)+WMPOSC(2)
      WWAPEX(KTRAP,1)=W(1,1,1)
      WWAPEX(KTRAP,2)=W(2,1,1)
      WWAPEX(KTRAP,3)=W(1,2,1)
      WWAPEX(KTRAP,4)=W(2,2,1)
```

```
      WWAPEX(KTRAP,5)=W(1,2,2)
      WWAPEX(KTRAP,6)=W(2,2,2)
      WWAPEX(KTRAP,7)=W(1,1,2)
      WWAPEX(KTRAP,8)=W(2,1,2)
   15 CONTINUE
      RETURN
      END
```

SUBROUTINE MAPAZM.FOR

```
      SUBROUTINE MAPAZM(NXMAP,NYMAP,XMVAL,WAX,WAY,WPAR)
      REAL WAX(1),WAY(1),WPAR(6),WWAREA(0:15,0:15)
      REAL SYNMAP(0:99,0:99),SUMARA(0:99,0:99)
      REAL AREPOL
      INTEGER I,IIO,IFAIL,IMX,IW,J,JJO,JMX,JW
      COMMON /MAP/SYNMAP,SUMARA
      CALL POLONFRA(4,WAX,WAY,WPAR,IIO,JJO,IMX,JMX,AREPOL,WWAREA,0,
     &IFAIL)
C     3--SYSTEMS OF COORDINATES: WPAR define ABS--location of window ;****
C     (IW,JW)=WINDOW--coordinates ; (I,J)=FRAME--coordinates **************
      DO 43 I=0,IMX-1
      DO 43 J=0,JMX-1
      IF(WWAREA(I,J).LT.0.002) GO TO 43
      IW=IIO+I
      JW=JJO+J
      IF(IW.LT.0.OR.JW.LT.0) GO TO 43
      IF(IW.GT.NXMAP.OR.JW.GT.NYMAP) GO TO 43
      SYNMAP(IW,JW)=SYNMAP(IW,JW)+WWAREA(I,J)*XMVAL
      SUMARA(IW,JW)=SUMARA(IW,JW)+WWAREA(I,J)
   43 CONTINUE
      RETURN
      END
```

SUBROUTINE MAPRGM.FOR

```
      SUBROUTINE MAPRGM(NXMAP,NYMAP,NPPPMN,NPPPMX,XM,WPAR,LNPPP,WWAPEX)
      REAL WPAR(6),WWAREA(0:15,0:15),WAX(4),WAY(4),WWAPEX(128,9),
     &XM(1)
      INTEGER KA,KAPX,KFF,KNPPP,LNPPP(1)
      DO 45 KNPPP=NPPPMN,NPPPMX
      KFF=LNPPP(KNPPP)
      IF(XM(KFF).LT.WPAR(2)) GO TO 45
      KA=0
      DO 40 KAPX=1,7,2
      KA=KA+1
      WAX(KA)=WWAPEX(KNPPP,KAPX)
```

```
40  WAY(KA)=WWAPEX(KNPPP,KAPX+1)
    CALL MAPAZM(NXMAP,NYMAP,XM(KFF),WAX,WAY,WPAR)
45  CONTINUE
    RETURN
    END
```

INITIAL DISTRIBUTION

	No. of Copies
U.S. Army Materiel System Analysis Activity ATTN: AMXSY-MP (erbert Cohen) Aberdeen Proving Ground, MD 21005	1
IIT Research Institute ATTN: GACIAC 10 W. 35th Street Chicago, IL 60616	1
AMSMI-RD	1
AMSMI-RD-CS-R	15
AMSMI-RD-CS-T	1
AMSMI-RD-AS	1
AMSMI-RD-AS, A.H. GREEN	4
AMSMI-GC-IP, Mr. Fred Bush	1